# Design of Discrete-Time Filters for Efficient Implementation

by

Dennis Wei

S.B. EECS, Massachusetts Institute of Technology (2006)
S.B. Physics, Massachusetts Institute of Technology (2006)
M.Eng. EECS, Massachusetts Institute of Technology (2007)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2011

Author . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
Department of Electrical Engineering and Computer Science
May 18, 2011

Certified by . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
Alan V. Oppenheim
Ford Professor of Engineering
Thesis Supervisor

Accepted by . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
Leslie A. Kolodziejski
Chair, Department Committee on Graduate Students

# Design of Discrete-Time Filters for Efficient Implementation

by

Dennis Wei

Submitted to the Department of Electrical Engineering and Computer Science
on May 18, 2011, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering

## Abstract

The cost of implementation of discrete-time filters is often strongly dependent on the number of non-zero filter coefficients or the precision with which the coefficients are represented. This thesis addresses the design of sparse and bit-efficient filters under different constraints on filter performance in the context of frequency response approximation, signal estimation, and signal detection. The results have applications in several areas, including the equalization of communication channels, frequency-selective and frequency-shaping filtering, and minimum-variance distortionless-response beamforming.

The design problems considered admit efficient and exact solutions in special cases. For the more difficult general case, two approaches are pursued. The first develops low-complexity algorithms that are shown to yield optimal or near-optimal designs in many instances, but without guarantees. The second focuses on optimal algorithms based on the branch-and-bound procedure. The complexity of branch-and-bound is reduced through the use of bounds that are good approximations to the true optimal cost. Several bounding methods are developed, many involving relaxations of the original problem. The approximation quality of the bounds is characterized and efficient computational methods are discussed. Numerical experiments show that the bounds can result in substantial reductions in computational complexity.

Thesis Supervisor: Alan V. Oppenheim
Title: Ford Professor of Engineering

# Acknowledgments

A thesis is a single-author document but it is never just the work of a single author. First I would like to thank my research advisor, Prof. Alan Oppenheim (whose name at least appears on the title page). Al's contributions to this thesis are many, but perhaps the most important is one of perspective, helping me to step back from the work, whether at the level of a section, a chapter, or the whole thesis, and distill the implications in an honest and positive way. Al has also engaged in a meticulous reading of drafts of this thesis, his latest effort in my development as a technical writer. More generally, I am greatly indebted to Al for his masterfully subtle and highly successful mentorship, and for his true concern for the growth and well-being of his students. There's a reason why one speaks of an "academic genealogy".

Next I would like to thank the members of my thesis committee, Prof. Vivek Goyal and Prof. Pablo Parrilo. Their comments in committee meetings and one-on-one discussions were most valuable and contributed to a stronger thesis. The ideas I couldn't get to are among the directions for future work that I am most excited about. Pablo in particular was instrumental in validating the contributions to optimization in this thesis and encouraging me to pursue them further. Vivek spurred me to think more about signal processing applications and biased me toward a more rigorous approach. I have also benefited from Vivek's mentorship at several points in my graduate school career.

Continuing with the genealogical analogy, there is my academic "family" for the past six years, the Digital Signal Processing Group (DSPG). I had the privilege of interacting with the following members, past and present: Tom Baran, Ballard Blair, Ross Bland, Petros Boufounos, Sefa Demirtas, Sourav Dey, Dan Dudgeon, Xue Feng, Kathryn Fischer, Zahi Karam, Al Kharbouch, Jon Paul Kitchens, Joonsung Lee, Jeremy Leow, Shay Maymon, Martin McCormick, Joe McMichael, Milutin Pajovic, Charlie Rohrs, Melanie Rudoy, Charles Sestok, Joe Sikora, Eric Strattman, Archana Venkataraman, and Matt Willsey. DSPG strives to be a welcoming, supportive, and wildly creative research environment, and it is the people who make it so. Eric Strattman and Kathryn Fischer deserve special mention as the quiet performers of many feats that keep DSPG running smoothly; they have always been happy to help with anything. With regard to this thesis specifically, additional thanks go to the following individuals (again in alphabetical order): To Tom, for starting

down this path and for our early collaboration on sparse filter design, and for his continued support of the research, including his maintenance of the group computer to facilitate numerical experiments. To Ballard, for educational discussions about channel equalization. To Petros, for his suggestion during a group meeting that diversified the thesis away from frequency response approximation and for stimulating discussions on quantization, for hiring me as an intern at Mitsubishi Electric Research Laboratories where I learned about synthetic aperture radar, and for help and advice with the next steps in my career. To Xue, for her assistance in developing the idealized channel equalization example, running the design experiments, and interpreting the results. To Jon Paul, for enthusiastic discussions about filter and array design and enclosing shapes, and for his guidance in formulating the MVDR beamforming example. To Charles Sestok, for the contributions in his thesis to special cases of the sparse filter design problem and for discussions that were essential to the organization of this thesis.

Many other people have shepherded me through my graduate school experience. I will just mention a few of them here: my undergraduate academic advisor, Prof. George Verghese, the faculty and fellow students who I have taught with, and numerous colleagues and friends (especially Dan and John from STIR, our closest "neighbours"), including some from my undergrad days (Gabe, Jason, Nat and Nori) who help me to escape the grind.

I consider myself very fortunate to have had the love and companionship of my girlfriend Joyce for the greater part of my time in graduate school. She has given me happiness and comfort, especially in these last few months when I had doubts that I would ever finish. Being a fellow electrical engineering Ph.D. and digital circuit designer, her expertise in hardware implementation of digital filters helped me to frame and motivate this work. I look forward to a life together for many years to come.

Lastly, to my parents, for unwavering love and support that I cannot hope to repay. They have brought me so far, through elementary school, high school, university, and now a Ph.D. At each step, they have always gently encouraged me to aim for the next level. During these last ten years at MIT, they have helped me with or reminded me about many of my non-academic obligations so that I could concentrate on my studies. I will always value their wisdom and counsel, and I hope that the next steps in this journey will be equally rewarding.

# Contents

# List of Figures

# List of Tables

# List of Algorithms

# Chapter 1

# Introduction

Discrete-time filters are widely used in signal processing applications spanning digital communications, radar and sonar, consumer electronics, biomedical diagnostics, and many additional domains. The prevalence of discrete-time filtering is expected to increase as more devices are endowed with sensing, processing, and communication capabilities. Accordingly, the design of discrete-time filters has remained an area of ongoing research.

As with many problems in engineering, filter design can be viewed as a trade-off between achieving a desired level of performance and maintaining low complexity. Naturally, it is desirable to make the most efficient use of resources to accomplish a filtering task. The measure of efficiency that is most relevant depends on the medium in which the filter is implemented and the availability of different resources. For example, the amount of computation may still be a limiting factor when processing data at high rates or in high dimensions, notwithstanding advances in digital electronics. If the filter is to be implemented on a wireless device as is increasingly common, power consumption may be the primary concern. In integrated circuit implementations, circuit area is often used as a measure of complexity. Similar considerations carry over to the design of sensor arrays, which is a close analogue to discrete-time filter design in many cases. Sensors can be expensive and can therefore dominate the cost in applications such as underwater acoustic arrays, or they can be inexpensive and consequently constrained in terms of computation, power, and communication.

The efficiency of a filter implementation also depends on many factors at different stages of the design and implementation process. For example, the ultimate speed and fabrication cost of integrated circuit implementations are strongly influenced by the underlying technology. Numerous techniques at the circuit and architectural levels can be employed to increase speed or decrease area and power consumption. With sensor arrays, the physical design of the individual sensors determines to a large extent the cost and capabilities of the array.

Given the variety of relevant complexity measures and contributors to efficiency, from a signal processing standpoint it is useful to define more abstract measures of complexity that are appropriate for design at the algorithmic level. With these abstractions, the design of the filter, or more precisely the specification of the algorithm embodied by the filter, can be done without the need for detailed knowledge of the implementation. Indeed, as discussed above, the filter designer often does not have full control over the physical realization of the filter. The choice of more abstract measures of complexity (and also of performance) is somewhat of an art as it should balance faithfulness to the eventual implementation against the tractability of the design problem. On the one hand, it is desirable that the chosen measure of complexity be an accurate reflection of the true implementation cost. On the other hand, the ease of generating and optimizing designs depends on the measures of complexity and performance selected, and also on the relationship between them.

A conventional abstraction in this regard is to measure the complexity of a filter by the total number of coefficients, which is often referred to as the order of the filter. The order can be a reasonable metric since it controls both the amount of computation and the amount of memory at the same time. Many classical methods in filter design optimize different performance measures given a fixed filter order. For example, the Parks-McClellan algorithm [5] minimizes the Chebyshev error with respect to an ideal frequency response for a linear-phase FIR filter of a given length. If the error is instead measured in a least-squares sense, an optimal design can be specified in closed form. In a similar vein, when the filter length is restricted, the linear minimum mean-square error (MMSE) estimate of a random process is given by the FIR Wiener filter. These conventional methods make it straightforward to determine an optimal trade-off between performance and filter order.

This thesis addresses the design of discrete-time filters according to less conventional measures of complexity that can be more representative of the actual cost of implementation in several common forms. A variety of complexity and performance measures are considered, giving rise to different trade-offs and problem formulations. Once a particular pair of measures has been selected, a natural objective is to obtain the best possible trade-off between them. Accordingly, the tools of optimization, both theoretical and practical, will play a prominent role throughout the thesis. The problem can be posed as either one of minimizing complexity given performance specifications, or of maximizing performance given complexity constraints, depending on which formulation is more convenient. It will be seen however that for the complexity measures considered in this thesis, an optimal trade-off is often computationally difficult to determine. The difficulty of optimal design necessitates a second trade-off involving the design algorithm, specifically a trade-off between the optimality of the design and the computational complexity of the algorithm.

At one end of the spectrum, we will be interested in algorithms that are constructed to be low in complexity. Such algorithms are desirable in situations where the amount of computation is limited, for example when a filter is redesigned adaptively. In some special cases, low-complexity algorithms are sufficient to ensure optimal designs, but in the general case they do not. It will be seen however in numerical experiments and design examples that low-complexity algorithms can yield designs that are optimal or near optimal in many instances. Unfortunately, these algorithms by themselves do not provide any indication of the proximity to the true optimum.

An equally important objective is to develop optimal algorithms. Aside from the obvious benefit of guaranteeing optimal solutions, the main value of optimal algorithms lies in determining fundamental limits and serving as a benchmark against which lower-complexity algorithms may be compared. This last point can be especially relevant when it is the lower-complexity algorithms that will be used in practice. In this thesis, we will focus on a general-purpose combinatorial optimization procedure known as branch-and-bound. The emphasis will be on improving the efficiency of branch-and-bound to mitigate the high intrinsic complexity of the design problem. The principal approach to achieving greater efficiency is to develop approximations that lead to bounds on the true optimal value (true minimum complexity or maximum performance) that can be leveraged by the branch-and-bound algorithm.

A third alternative intermediate between the first two can be obtained by limiting the complexity of the branch-and-bound algorithm, for example by terminating it after a specified period of time. In this case, the design returned is at least as good as one given by a low-complexity algorithm, and we have in addition a bound on the deviation from optimality. This intermediate option will be used in some of the design examples in the thesis.

In the next section, the measures of complexity emphasized in this thesis are discussed and motivated in the context of common filter implementations. We also delineate the scope of the thesis by briefly discussing areas of filter design that are not addressed further. Section 1.2 summarizes previous work in designing filters according to the complexity measures introduced in Section 1.1. In Section 1.3, the organization of the body of the thesis is described and the content of each chapter is highlighted.

## 1.1 Complexity measures and scope of thesis

As noted in the introduction, the total number of coefficients is traditionally used as an indication of filter complexity. It is often the case however that the cost of implementation is dominated by arithmetic operations. In these situations, the number of non-zero coefficients may be a more appropriate metric given that operations associated with zero-valued coefficients may be omitted. This leads to a desire for designs with fewer non-zero coefficients, i.e., sparse designs, which can be exploited to reduce computation, hardware, or power consumption, depending on the form of implementation. For instance, in integrated circuit implementations, multipliers and adders can be deactivated or even eliminated to save power and area, or the supply voltage may be lowered to take advantage of the slower rate at which computations can be performed. Sparsity is perhaps of even greater interest in the case of sensor arrays since a sparse design allows for the elimination of array elements, yielding savings in data acquisition and communication costs. With these motivations in mind, a major part of the thesis focuses on the design of sparse filters. We note here that it is straightforward to generalize the coefficient value associated with lower cost, which is zero in the case of sparse filters, to an arbitrary non-zero value.

The metric of coefficient sparsity assumes implicitly that all operations involving non-zero coefficients are equally costly. This assumption is largely true when the operations are performed by standard units, e.g. generic multiplier circuits or calls to multiplication functions. However, there are other situations where the specific values of non-zero coefficients can significantly affect the complexity. In digital implementations in particular, coefficients must be quantized and the complexity depends on the number of bits used to represent each coefficient. Usually the number of bits is the same for all coefficients and is referred to as the coefficient wordlength. In this thesis, we consider a measure of complexity that refines the concept of wordlength by permitting different numbers of bits for each coefficient. We assume a conventional sign-magnitude binary representation and measure the cost of each coefficient by the position of the most significant non-zero bit, or equivalently, by the number of bits excluding leading zeros. Hence we refer to this cost measure as the number of non-leading-zero (NLZ) bits. The NLZ criterion is motivated by the tendency of arithmetic operations involving larger coefficients to be more expensive, requiring for instance larger multipliers or more active bits when all multipliers are constrained to be of the same size. Electrical switching activity also increases with larger coefficients [6,7]. Thus a decrease in the total number of NLZ bits can result in corresponding decreases in power and hardware.

The NLZ measure assumes that it is relatively simple to avoid operations associated with leading zero bits. The idea can be naturally extended to include trailing zeros and zeros between non-zero bits, leading to a so-called multiplierless implementation in which multiplications are realized using a combination of additions and bit shifts, omitting additions corresponding to zero-valued bits. In the context of multiplierless implementations, it has been observed that the number of non-zero bits, and hence the number of additions, can be significantly reduced by using a number representation based on signed powers-of-two (SPTs). In an SPT representation, each filter coefficient is represented in the form

$$\sum_{p=0}^{P-1} s_p 2^p, \quad s_p \in \{-1, 0, +1\},$$

where $P$ is the maximum allowable wordlength. In terms of implementation, a value of $+1$ for the digit $s_p$ indicates an addition following a bit shift by $p$ positions, whereas $s_p = -1$ indicates subtraction. The use of signed digits enables greater efficiency in terms of the number of additions and subtractions; for example, multiplication by the constant $15 = 2^4 - 1$ can be implemented through one subtraction instead of three additions in an unsigned representation. It can be seen that the SPT representation of a number is not unique, and even the representation with the minimal number of non-zero digits may not be unique. Among minimal representations, one uniquely defined choice is the canonic signed digit (CSD) representation, which has the additional property that no two non-zero digits are adjacent [8]. A CSD representation reduces the number of additions and subtractions by about 1/3 on average relative to a conventional two's complement representation [9]. Accordingly, many researchers have chosen to focus on the CSD representation, and we will do the same in this thesis. Specifically, the complexity of a design will be measured by the total number of non-zero digits in all coefficients.

We note that the NLZ measure can be viewed as an intermediate step between coefficient sparsity and bit sparsity in a multiplierless setting. The NLZ measure is emphasized in this thesis because of its mathematical properties, notably monotonicity with respect to coefficient magnitude, and because it is often possible to omit leading zeros without resorting to a full multiplierless implementation.

Thus far, we have discussed only time-domain implementations of linear time-invariant filters. FIR filters can also be implemented in the frequency domain through multiplication with the discrete Fourier transform of the input sequence. Frequency-domain implementations can be particularly efficient in terms of computation thanks to FFT algorithms; for a filter with impulse response of length $N$, the computation of $N$ output samples requires on the order of $N \log N$ arithmetic operations compared to $N^2$ for direct convolution [10]. However, this comparison does not take into account other measures of efficiency. Depending on the implementation, FFT-based filtering may require more complicated data access, control mechanisms, and/or additional memory, leading to higher hardware cost and power consumption. The latency introduced by filtering the input in blocks may also be unacceptable. Moreover, the computational complexity of FFT-based filtering is usually dominated by the FFT itself, and further optimization of the complexity is mainly an issue of FFT implementation and not of filter design. Hence in keeping with the emphasis on filter design, we will restrict our attention to time-domain implementations.

This thesis focuses mainly on FIR filters, and more specifically on direct form structures which are commonly used in the FIR case. Since the design of direct-form FIR filters is analogous to the design of linear sensor arrays, the methods in this thesis are directly applicable to the latter as well and some of the design examples are motivated by array processing problems. The algorithms that we develop for direct-form structures can be seen as complementary to the many approaches for reducing computational complexity that rely on cascade structures [11–22]. Most of the cascade-form techniques are restricted to frequency-selective filters and narrowband filters in particular, for which specific cascade configurations have been found to be efficient. Moreover, they do not apply to sensor arrays.

In contrast, our algorithms are equally applicable to all types of filters. Furthermore, most of the constituent filters in the cascade structures cited above are assumed to be implemented in direct form and designed using conventional algorithms. Thus the problem of optimizing the computational complexity of direct-form filters is still unresolved. Direct-form and cascade techniques can also be combined to yield more efficient implementations than with either alone as demonstrated in [23, 24].

IIR filters can be more attractive than FIR filters in terms of efficiency, assuming that linear phase is not a requirement and that issues with stability and roundoff error feedback can be managed. In particular, given similar frequency responses, the total number of coefficients for an IIR filter is usually lower than for an FIR filter. Methods developed for direct-form FIR filters can be applied to design the numerator polynomial in the transfer function of an IIR filter. Similarly, the design of the denominator polynomial can be transformed into an FIR design problem provided that there is some means to ensure the stability of the filter. However, even for the FIR case, the problems of optimizing the complexity measures we have chosen, namely the number of non-zero coefficients, the number of NLZ bits, and the number of non-zero digits in a CSD representation, generally have non-polynomial complexity and are computationally difficult. As we will see, the difficulty is mitigated somewhat in the FIR case by the linear or convex quadratic relationship between the filter coefficients and the performance measures considered in the thesis. In contrast, in the general IIR problem of simultaneously designing the transfer function numerator and denominator, there is a nonlinear, non-convex relationship between the chosen performance measures and the most common choices for implementation parameters, i.e., the coefficients in a direct form or cascade form structure or the locations of polynomial roots. While designs can be obtained heuristically or in some special cases with the aid of design formulas, optimal design is difficult even in the conventional setting of fixed filter order. For these reasons, we do not treat the IIR case in this thesis. Similar statements apply to lattice structures and other structures for which there is a complicated relationship between the coefficients in the implementation domain and the measure of performance.

## 1.2 Previous work

In this section, we summarize previous work in filter design that relates to the measures of complexity identified in Section 1.1. The design of sparse filters is treated in Section 1.2.1 and the design of filters with a reduced number of bits or SPTs is treated in Section 1.2.2.

### 1.2.1 Sparse filter design

A classical problem in filter design is the approximation of an ideal frequency response. Previous work on the design of sparse filters to approximate frequency responses can be broadly categorized into two approaches. In the first approach, which is applicable mostly to frequency-selective filters, the locations of zero-valued coefficients are pre-determined in accordance with the characteristics of the desired frequency response. In examples such as interpolated FIR filters [17, 18] and the frequency-response masking technique [20, 21], a sparse filter is cascaded with one or more equalizing filters. Sparse direct-form designs for approximately $n$th-band filters were developed in [25] [1]. The second approach is more general and does not pre-specify the locations of zero-valued coefficients, instead attempting to choose them optimally to minimize the number of non-zero coefficients subject to frequency response constraints. The resulting combinatorial optimization problem can be solved exactly using integer programming [3, 26]. The complexity of optimal design has also motivated the development of low-complexity heuristics, based for example on forcing small coefficients to zero [27], orthogonal matching pursuit [1], or $\ell^1$ relaxation [28]. A non-convex approximate measure of sparsity based on the $p$-norm has also been proposed [29].

All of the references above address the approximation of frequency responses according to a Chebyshev error criterion. In comparison, weighted least-squares criteria have received less attention. As discussed in [30], a weighted least-squares metric is commonly employed as an alternative to the Chebyshev metric because of greater tractability and an association with signal energy or power. Approaches based on zeroing small coefficients [31] and subset selection [22] have been developed for the weighted least-squares case.

A second application of discrete-time filters is in estimating the values of a signal from those of another. In the context of sparsity, a particularly important example is the equalization of communication channels, which involves the estimation of transmitted symbols

[1] An $n$-th band filter has the property that every $n$th impulse response coefficient is equal to zero except for the central coefficient.

from received samples corrupted by noise and inter-symbol interference. It has been observed that the sparsity of the power-delay profiles of many communication channels can be exploited in the design of the equalizer. While there are numerous specialized methods for estimating sparse channel responses that can be combined with conventional equalization techniques (for example in [32]), here we focus on methods for designing equalizers that are also intended to be sparse. Optimal algorithms for minimizing the mean squared estimation error given a fixed number of equalizer taps are developed in [33] and [34], the former based on branch-and-bound for discrete-time equalizers and the latter based on nonlinear optimization for continuous-time tapped-delay-line equalizers. In [35], the locations of non-zero equalizer coefficients are chosen to coincide with the locations of large channel coefficients. This approach is refined in [36] and [37], which derive analytical expressions for the coefficients of non-sparse equalizers in terms of the channel coefficients and use these expressions to allocate non-zero taps in a sparse equalizer. A modified decision-feedback equalizer (DFE) structure is proposed in [38] to better exploit the sparsity of the channel response. An alternative class of approximate methods allocates taps according to simplified mean squared error (MSE) or output signal-to-noise ratio (SNR) metrics. The allocation can be done in a single pass [39], two alternating passes [40], or one tap at a time in a recursive manner [41,42]. The sparsity of the channel response is used in [42] to reduce the tap allocation search space. A method based on the theory of statistical sampling design is also presented in [41].

Another variant of the estimation problem is signal prediction in which past values of a signal are used to predict future values. Sparse linear prediction is discussed in [43], which uses iteratively reweighted $\ell^1$ minimization to promote sparsity in the predictor coefficients and in the residuals, thereby improving the efficiency of speech coding.

A third context in which filters are used is in the detection of signals in noisy environments, where the objective of filtering is to increase the probability of detection. A widely used measure of performance in detection is the SNR of the filter output. It is well-known that the SNR is monotonically related to the probability of detection in the case of Gaussian noise [44]. Motivated by the desire to reduce communication costs in distributed systems, Sestok [2,45] considered the design of linear detectors that use only a subset of the available measurements.

### 1.2.2 Bit-efficient filter design

The effect of coefficient quantization on the frequency response has long been a concern in filter design. One method for combating coefficient quantization is to carry out some or all of the design in the discrete-valued domain. Early heuristic efforts [46, 47] demonstrated that discrete optimization can reduce the coefficient wordlength necessary to satisfy given frequency response specifications when compared to straightforward rounding of a continuous-valued design. The wordlength can also be reduced by minimizing a certain statistical measure before rounding [48]. More recently, there has been research into representing coefficients by their differences or applying other transformations to decrease the dynamic range and wordlength [6, 7]. Perhaps the first application of integer programming to the problem is by Kodek in [49] for the case of Chebyshev-optimal linear-phase FIR filters. A simplified branch-and-bound algorithm was suggested in [50]. For the specific case of uniformly quantized coefficients, a comparison between integer programming and local search methods suggested that integer programming offers a small performance benefit relative to its much greater complexity [51].

In contrast, integer programming was found to significantly improve performance in the case of non-uniform coefficient quantization, and specifically for coefficients that are sums of a limited number of SPTs [52]. As discussed in Section 1.1, SPT representations are particularly attractive for multiplierless filter design, and hence there is a significant body of research directed at these representations that is summarized in the remainder of the section.

When the number of SPTs per coefficient is restricted, the coefficient values that can be realized are non-uniformly distributed. Many authors have therefore considered the option of scaling the filter coefficients by an overall factor to improve the approximation to an ideal frequency response. Formal algorithms for choosing an appropriate scale factor are presented in [53].

Several techniques have been developed to alleviate the complexity of integer programming by exploiting properties of the filter design problem. In [54], the convexity of the minimax criterion was exploited to reduce the complexity of a branch-and-bound procedure. In a similar spirit, it was observed in [55] that nearby subproblems in a branch-and-bound tree often exhibit similar sets of active constraints, suggesting a way to reduce the number of constraints in the solution of subsequent subproblems. Theoretical limits on the performance achievable by discrete-coefficient filters were used to improve lower bounds on the optimal cost of subproblems in [56].

As a complement to integer programming, various local search strategies have been devised in which the search complexity is controlled by limiting the number of coefficients that are varied and the number of values that are searched at any one time [8, 57, 58]. Other techniques include using heuristic sensitivity criteria to guide the quantization of coefficients [59, 60].

Later it was observed by many researchers that the quality of the frequency response approximation tends to saturate when the number of SPTs is constrained on a per-coefficient basis. Cascade structures [61], filter sharpening [58,62] and differential coefficients [63] were proposed to circumvent this fundamental limit. The saturation effect can also be avoided to some extent and without increasing the implementation complexity by constraining only the total number of SPTs and allowing them to be distributed as needed among the coefficients. An integer programming formulation of the minimization of the total number of SPTs subject to frequency response constraints was given in [64]. In both [64] and [65], knowledge of the feasible range for each coefficient was used to reduce the number of free variables. The idea of using information provided by feasible ranges was further developed and combined with an empirically effective branching strategy in [66].

Alternatives to integer programming are also available for the case in which the total number of SPTs is fixed rather than the number for each coefficient. A low-complexity strategy for allocating SPTs is given in [67]. Li et al. have developed an efficient algorithm for minimizing the Chebyshev error in the impulse response domain rather than in the frequency domain under a constraint on the total number of SPTs [68]. While a close approximation of the impulse response tends to result in a close approximation of the frequency response, this is not guaranteed and there is no direct control over the quality of the frequency response approximation. More sophisticated extensions of the algorithm in [68] attempt to incorporate a measure of the frequency response performance as well [9,69]. In a related approach, Llorens et al. consider the minimization of the impulse response error in a $p$-norm sense for finite values of $p$ [70].

While the majority of the literature has focused on Chebyshev approximation of frequency responses, algorithms have also been developed to design discrete-coefficient FIR filters that minimize the weighted squared error. The weighted least-squares case is computationally easier than the Chebyshev case because the continuous-valued version of the problem has a closed-form solution. Branch-and-bound algorithms are presented in [71, 72], while an approach based on moving time horizons is proposed in [73]. The approach in [73] can be seen as a generalization of sigma-delta quantization methods (e.g. [74]) in which the frequency response distortion is directed away from a band of interest.

Given that the optimal design of discrete-coefficient filters often involves difficult discrete optimization problems, a number of authors have been motivated to apply general-purpose stochastic algorithms. Examples in this category include simulated annealing [75], mean field annealing [76], and genetic algorithms [77, 78].

A different approach altogether to reducing the computational complexity of multiplierless filters is to identify and eliminate redundant patterns of bits shared by many coefficients, a technique known as common subexpression elimination. Computational savings are realized by performing the computation specified by the bit pattern once and then distributing the result to all coefficients that require it. Algorithms for common subexpression elimination can be found in [79–83].

## 1.3 Outline of thesis

The remainder of the thesis is organized into three parts. Each part examines a different class of performance-complexity trade-offs. A conclusion and suggestions for future work follow in Chapter 9.

The first part consists of Chapters 2–4 and addresses the design of sparse filters under a quadratic constraint on performance. In Chapter 2, it is shown that three different filter design problems: weighted least-squares approximation of frequency responses, signal estimation, and signal detection, can be accommodated within this general framework. In the remainder of Chapter 2, we present design algorithms that are restricted to be low in computational complexity. Some of these algorithms are directed at special cases for which they result in optimal designs. For the general case, we discuss a heuristic algorithm that frequently yields optimal or near-optimal designs as seen in the examples in Chapter 4.

At the other end of the optimality-complexity trade-off, Chapter 3 discusses optimal algorithms for quadratically constrained sparse filter design. An algorithm is developed based on the branch-and-bound procedure, which is reviewed in Section 3.1. As will be made clear in that section, the complexity of branch-and-bound is strongly affected by the quality of available lower bounds on the optimal cost. Therefore the bulk of Chapter 3 is focused on the development of lower bounds. Emphasis is placed on understanding the quality of the bounds from both analytical and numerical perspectives. Equally important is the complexity involved in computing the bounds, and hence we also discuss efficient computational methods in Chapter 3.

In Chapter 4, the algorithms developed in Chapters 2 and 3 are applied to a range of examples. We verify that the heuristic algorithm from Chapter 2 produces near-optimal designs, and that the lower bounds derived in Chapter 3 can significantly reduce the complexity of the branch-and-bound algorithm. Potential applications of the algorithms are illustrated through design examples, specifically the design of sparse equalizers for wireless communication channels and the design of sparse beamformers for detection and interference rejection.

Chapters 5–7 form the second part of the thesis and extend the development of Chapters 2–4 to measures of complexity for quantized representations, specifically the number of NLZ bits in a sign-magnitude binary representation and the number of SPTs in a CSD representation. The same quadratic performance constraint is considered and our framework applies again to the three problems of frequency response approximation, signal estimation, and signal detection as discussed in Chapter 5. Chapter 5 also presents low-complexity design algorithms, specifically exact algorithms for special cases and a heuristic algorithm for the general case as in Chapter 2.

Chapter 6 follows the structure of Chapter 3 in developing optimal branch-and-bound algorithms for bit-efficient filter design under a quadratic constraint. Much of the chapter is again focused on lower bounds. The techniques of Chapter 3 are extended to the bit-based complexity measures and the quality of the resulting bounds is evaluated. Efficient computational methods are also addressed.

In Chapter 7, we discuss the application of the design algorithms in Chapters 5–6 to a variety of examples. Similar to Chapter 4, it is shown that the lower bounds developed in Chapter 6 are capable of making the branch-and-bound algorithm more efficient. We also revisit some of the design examples from Chapter 4 and observe similar trade-offs and dependences for the bit-based cost metrics.

The third part of the thesis consists of Chapter 8 and is centered on the design of sparse filters under a Chebyshev constraint in the frequency domain, i.e., a bound on the maximum weighted frequency response error. We extend some of the methods of Chapters 2–3 to the Chebyshev error criterion. Two low-complexity heuristic algorithms are discussed and some of the bounding techniques of Chapter 3 are generalized. However, because of the significant increase in computational complexity, optimal algorithms are not developed fully. We illustrate the performance of the heuristic algorithms and lower bounds through several examples involving the design of frequency-selective filters and beamformers and a frequency response equalizer.

# Chapter 2

# Sparse filter design under a quadratic constraint: Problem formulations and low-complexity algorithms

In Chapters 2–4, we consider three related problems in sparse filter design, the first involving a weighted least-squares constraint on the frequency response, the second a constraint on MSE in estimation, and the third a constraint on SNR in detection. It is shown in Section 2.1 that all three problems can be formulated under a common framework corresponding to

$$\min_{\mathbf{b}} \quad \|\mathbf{b}\|_0 \qquad \text{s.t.} \qquad (\mathbf{b}-\mathbf{c})^T\mathbf{Q}(\mathbf{b}-\mathbf{c}) \leq \gamma, \tag{2.0.1}$$

where $\mathbf{b}$ is a vector of coefficients, $\mathbf{Q}$ is a symmetric positive definite matrix, $\mathbf{c}$ is a vector of the same length as $\mathbf{b}$, and $\gamma > 0$. We use for convenience the zero-norm notation $\|\mathbf{b}\|_0$ to refer to the number of non-zero components in $\mathbf{b}$. The abstract formulation in (2.0.1) allows for a unified approach to solving not only the three stated problems but also others with quadratic performance criteria as in (2.0.1).

The design of sparse filters according to (2.0.1) is in general a computationally difficult problem. As is explained in Section 2.1, sparse filter design differs from the problem of obtaining sparse approximate solutions to underdetermined systems of linear equations, i.e., the sparse linear inverse problem, which has received considerable attention recently in compressive sensing. Therefore a different set of approaches is required. In this chapter, we focus on design algorithms that are low in complexity. In some special cases, low-complexity algorithms are sufficient to ensure optimal solutions to problem (2.0.1). Section 2.2 presents three such cases in which the matrix $\mathbf{Q}$ is diagonal, block-diagonal, or banded. For the more difficult general case, a low-complexity heuristic algorithm is developed in Section 2.3. This algorithm occupies one end of the optimality-computational complexity trade-off discussed in Chapter 1. Numerical experiments and design examples in Chapter 4 demonstrate that the algorithm often yields optimal or near-optimal solutions, albeit without guarantees. Optimal algorithms are considered later in Chapter 3.

## 2.1 Problem formulations and reductions

We begin this section with a discussion of the abstract problem (2.0.1), interpreting it from a geometric perspective and contrasting it with the sparse linear inverse problem. We then formulate in Sections 2.1.1–2.1.3 the problems of sparse filter design for weighted least-squares approximation of frequency responses, for estimation or prediction under an MSE constraint, and for signal detection under an SNR constraint. It is shown that all three problems can be reduced to (2.0.1), making it sufficient to focus on (2.0.1) alone.

Problem (2.0.1) is characterized by a single quadratic constraint,

$$(\mathbf{b}-\mathbf{c})^T\mathbf{Q}(\mathbf{b}-\mathbf{c}) \leq \gamma. \tag{2.1.1}$$

This constraint may be interpreted geometrically as specifying an ellipsoid centered at $\mathbf{c}$. As will be noted in Sections 2.1.1–2.1.3, the center $\mathbf{c}$ corresponds to the solution that maximizes performance when all coefficients are permitted to be non-zero. The matrix $\mathbf{Q}$ and parameter $\gamma$ determine the size and shape of the set of feasible solutions surrounding $\mathbf{c}$. Specifically, as illustrated in Fig. 2-1, the eigenvectors and eigenvalues of $\mathbf{Q}$ determine the orientation and relative lengths of the axes of the ellipsoid while $\gamma$ determines its absolute size. We will make reference to the ellipsoidal interpretation of (2.1.1) at several points in Chapters 2–7.

The problem of sparse filter design as stated in (2.0.1) differs in at least two important respects from the sparse linear inverse problem and more specifically its manifestations in

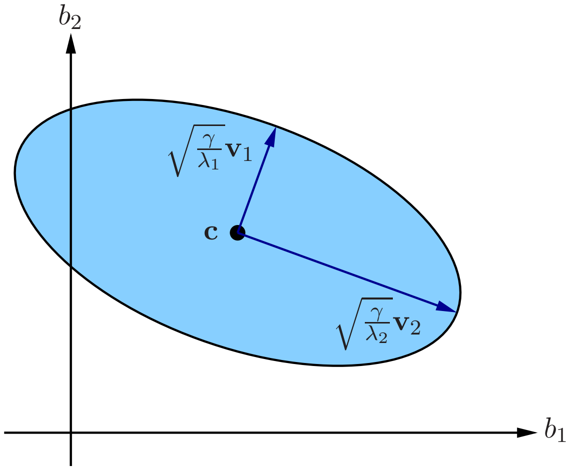

$b_2$
$\sqrt{\frac{\gamma}{\lambda_1}}\mathbf{v}_1$
$\mathbf{c}$
$\sqrt{\frac{\gamma}{\lambda_2}}\mathbf{v}_2$
$b_1$

Figure 2-1: Ellipsoid consisting of solutions satisfying the quadratic constraint (2.1.1). $\lambda_1$ and $\lambda_2$ are eigenvalues of $\mathbf{Q}$ and $\mathbf{v}_1$ and $\mathbf{v}_2$ are the associated eigenvectors.

compressive sensing with noisy measurements [84,85], atomic decomposition in overcomplete dictionaries [86], sparsity-regularized image restoration (e.g. [87] and references therein), and sparse channel estimation [32,88,89]. The sparse linear inverse problem can be formulated as

$$\min_{\mathbf{x}} \quad \|\mathbf{x}\|_0 \qquad \text{s.t.} \qquad \|\mathbf{\Phi}\mathbf{x} - \mathbf{y}\|_2 \leq \varepsilon, \tag{2.1.2}$$

where $\varepsilon$ is a parameter that limits the size of the residual $\mathbf{\Phi}\mathbf{x} - \mathbf{y}$. The first distinction between sparse filter design (2.0.1) and the sparse linear inverse problem (2.1.2) is in the structure of the feasible solution sets. In many applications of (2.1.2), the dimension of $\mathbf{y}$ is significantly lower than that of $\mathbf{x}$ and the system of equations is underdetermined. This is deliberately the case in compressive sensing and in overcomplete decomposition, while in channel estimation it is desirable to use relatively few measurements to estimate a long but sparse response, especially if done adaptively. As a consequence, the matrix $\mathbf{\Phi}^T\mathbf{\Phi}$, which corresponds to $\mathbf{Q}$ in (2.0.1), is rank-deficient and the set of feasible solutions is not a bounded ellipsoid but instead has infinite extent along certain directions. The second difference between sparse filter design and sparse linear inverse problems is one of perspective. In compressive sensing, image restoration, and sparse channel estimation, a certain level of sparsity or near-sparsity is assumed to enable reconstruction or estimation from fewer measurements. This assumption leads to a formulation such as in (2.1.2). However, the actual sparsity of a solution to (2.1.2) is of secondary importance as long as the larger goal of accurate reconstruction or estimation is achieved. In contrast, in sparse filter design, maximizing sparsity is the main objective, while no prior assumption is made regarding the expected level of sparsity. An algorithm that produces designs that are near-sparse in the sense of having many small but non-zero coefficients is not sufficient by itself.

In the remainder of this section, the three problems considered in Chapters 2–4 are reduced to problem (2.0.1). More specifically, it is shown that the performance constraint in each problem can be reduced to

$$\mathbf{b}^T\mathbf{Q}\mathbf{b} - 2\mathbf{f}^T\mathbf{b} \leq \beta, \tag{2.1.3}$$

which is equivalent to (2.1.1) with $\mathbf{f} = \mathbf{Q}\mathbf{c}$ and $\beta = \gamma - \mathbf{c}^T\mathbf{Q}\mathbf{c}$.

As mentioned in Section 1.1, this thesis is focused on FIR filter design, and we will use $N$ to denote the total number of coefficients, i.e., the dimension of the vector $\mathbf{b}$. The choice of $N$ is governed by two considerations. First, $N$ should be large enough to ensure the existence of designs meeting the performance specifications. Equivalently, the parameter $\gamma$ must be positive. In Sections 2.1.1–2.1.3, it will be made clear how $\gamma$ depends on the specifications in each problem. As $N$ is increased beyond the minimum required for feasibility, the optimal cost in problem (2.0.1), i.e., the minimum number of non-zero coefficients, decreases or at least stays the same. This is because all solutions that are feasible for a smaller value of $N$ are also feasible for a larger $N$. Thus to maximize sparsity, $N$ should be chosen based on the maximum allowable number of delay elements in a given application. For this reason, we will often refer to $N$ as the length of the filter, with the understanding that the final design may require fewer delays if coefficients at the ends of the vector $\mathbf{b}$ are zero.

### 2.1.1 Weighted least-squares filter design

In this problem, we wish to design a causal FIR filter with coefficients $b_0, b_1, \ldots, b_{N-1}$ and frequency response

$$H(e^{j\omega}) = \sum_{n=0}^{N-1} b_n e^{-j\omega n} \tag{2.1.4}$$

chosen to approximate a desired frequency response $D(e^{j\omega})$ (assumed to be conjugate symmetric). Specifically, the weighted integral of the squared error is constrained to not exceed a tolerance $\delta$, i.e.,

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} W(\omega)\left|H(e^{j\omega}) - D(e^{j\omega})\right|^2 d\omega \leq \delta, \tag{2.1.5}$$

where $W(\omega)$ is a non-negative and even-symmetric weighting function. The number of non-zero coefficients is to be minimized. Substituting (2.1.4) into (2.1.5), expanding, and comparing the result with (2.1.3), we can identify

$$Q_{mn} = \frac{1}{2\pi}\int_{-\pi}^{\pi} W(\omega)\cos\bigl((m-n)\omega\bigr)d\omega, \quad m = 0,\dots,N-1, \quad n = 0,\dots,N-1, \tag{2.1.6a}$$

$$f_n = \frac{1}{2\pi}\int_{-\pi}^{\pi} W(\omega)D(e^{j\omega})e^{j\omega n}\,d\omega, \quad n = 0,\dots,N-1, \tag{2.1.6b}$$

$$\beta = \delta - \frac{1}{2\pi}\int_{-\pi}^{\pi} W(\omega)\left|D(e^{j\omega})\right|^2\,d\omega. \tag{2.1.6c}$$

The matrix $\mathbf{Q}$ defined by (2.1.6a) is symmetric, Toeplitz, and positive definite, the last property holding as long as $W(\omega)$ is non-zero over some interval. Thus the frequency response constraint (2.1.5) can be rewritten in the form of (2.1.3) or (2.1.1). The fact that $\mathbf{Q}$ is Toeplitz is relatively unimportant as we will often work with submatrices extracted from $\mathbf{Q}$, which in general are no longer Toeplitz.

In the present case, the parameter $\gamma$ is given by

$$\gamma = \delta - \left(\frac{1}{2\pi}\int_{-\pi}^{\pi} W(\omega)\left|D(e^{j\omega})\right|^2\,d\omega - \mathbf{c}^T\mathbf{Q}\mathbf{c}\right).$$

It can be seen from (2.1.3) and (2.1.5) that $\mathbf{c} = \mathbf{Q}^{-1}\mathbf{f}$ is the minimum-error design of length $N$ and the quantity in parentheses above is the corresponding error. Hence $\gamma$ is the amount by which $\delta$ exceeds the minimum error.

The weighted least-squares criterion in (2.1.5) can arise as the result of modelling the input to the filter as a wide-sense stationary (WSS) random process $x[n]$ and using the mean-squared deviation of the filter output from the ideal output as the error metric. The error signal of interest $e[n]$ is depicted in Fig. 2-2. The mean-squared error $E\{e[n]^2\}$ is given by the left side of (2.1.5) with $W(\omega) = \Phi_{xx}(e^{j\omega})$, the power spectral density of the input. In this case, (2.1.6) can be rewritten in terms of the autocorrelation function $\phi_{xx}[m]$ as

$$\begin{aligned} Q_{mn} &= \phi_{xx}\left[|m-n|\right], \\ f_n &= \sum_{k=-\infty}^{\infty} d[k]\phi_{xx}[n-k], \\ \beta &= \delta - \sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} d[k]d[l]\phi_{xx}[k-l], \end{aligned}$$

where $d[n]$ is the inverse Fourier transform of $D(e^{j\omega})$.

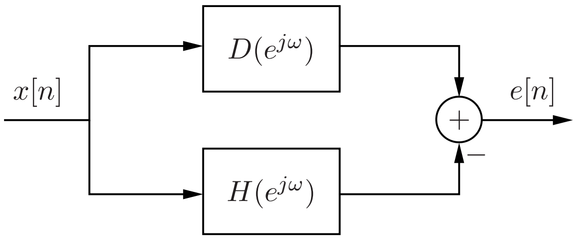


Figure 2-2: Definition of error signal $e[n]$.

A slightly different criterion that also leads to a problem of the same form is

$$\frac{1}{2\pi}\int_{-\pi}^{\pi}\left|G(e^{j\omega})H(e^{j\omega})-D(e^{j\omega})\right|^2 d\omega \le \delta, \tag{2.1.7}$$

where $G(e^{j\omega})$ is the frequency response of a given system and $H(e^{j\omega})$ is chosen such that the response of the cascade does not deviate from $D(e^{j\omega})$ by more than $\delta$ in the mean-square sense. The filter to be designed can be regarded as a compensator for $G(e^{j\omega})$. The entries of $\mathbf{Q}$, $\mathbf{f}$, and $\beta$ are now given by

$$\begin{aligned} Q_{mn} &= \frac{1}{2\pi}\int_{-\pi}^{\pi}\left|G(e^{j\omega})\right|^2\cos\big((m-n)\omega\big)d\omega = \sum_{k=-\infty}^{\infty} g[k]g[(m-n)+k],\\ f_n &= \frac{1}{2\pi}\int_{-\pi}^{\pi} G(e^{-j\omega})D(e^{j\omega})e^{j\omega n}\,d\omega = \sum_{k=-\infty}^{\infty} g[k]d[n+k],\\ \beta &= \delta - \frac{1}{2\pi}\int_{-\pi}^{\pi}\left|D(e^{j\omega})\right|^2 d\omega, \end{aligned}$$

assuming that $g[n]$ and $d[n]$ are real-valued. The addition of a weighting function to the constraint in (2.1.7) can be readily accommodated.

### 2.1.2 Estimation, prediction, and equalization

Another problem that can be reduced to the formulation in (2.0.1) is the estimation of a random process $x[n]$ from observations of a second random process $y[n]$ under the assumption that $x[n]$ and $y[n]$ are jointly WSS. The estimate $\hat{x}[n]$ is produced by processing $y[n]$ with a causal FIR filter of length $N$,

$$\hat{x}[n] = \sum_{m=0}^{N-1} b_m y[n-m]. \tag{2.1.8}$$

The goal is to minimize the number of non-zero coefficients $b_m$ while keeping the mean-squared estimation error below a threshold $\delta$, i.e.,

$$E\left\{(\hat{x}[n]-x[n])^2\right\} \le \delta. \tag{2.1.9}$$

Substituting (2.1.8) into (2.1.9), expanding, and comparing with (2.1.3), we find

$$Q_{mn} = \phi_{yy}\left[|m-n|\right], \tag{2.1.10a}$$

$$f_n = \phi_{xy}[n], \tag{2.1.10b}$$

$$\beta = \delta - \phi_{xx}[0], \tag{2.1.10c}$$

where we have defined the cross-correlation as $\phi_{xy}[m] = E\{x[n+m]y[n]\}$. The matrix $\mathbf{Q}$ is again symmetric, Toeplitz, and positive definite. In the estimation context, the vector $\mathbf{c} = \mathbf{Q}^{-1}\mathbf{f}$ corresponds to the causal Wiener filter of length $N$, $\phi_{xx}[0] - \mathbf{c}^T\mathbf{Q}\mathbf{c}$ is the corresponding error, and $\gamma$ is again equal to the difference between $\delta$ and the minimum error.

The problem of $p$-step linear prediction is a special case of the estimation problem with $x[n] = y[n+p]$ and $p$ a positive integer. Equation (2.1.10a) remains unchanged while $\phi_{xy}[n]$ is replaced with $\phi_{yy}[n+p]$ in (2.1.10b) and $\phi_{xx}[0]$ with $\phi_{yy}[0]$ in (2.1.10c).

An important application of the basic estimation problem formulated above is to the equalization of communication channels, in which case $x[n]$ represents a transmitted signal. For the case of linear equalization, $y[n]$ corresponds to the received sequence and can be modelled according to

$$y[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k] + \eta[n], \tag{2.1.11}$$

where $h[k]$ represents samples of the overall impulse response due to the combination of the transmit pulse, channel, and receive filter, and $\eta[n]$ is additive noise, assumed to be zero-mean, stationary with autocorrelation $\phi_{\eta\eta}[m]$, and uncorrelated with $x[n]$. Under this channel model, the auto-correlation and cross-correlation in (2.1.10) can be expressed as

$$\phi_{yy}[m] = \sum_{k=-\infty}^{\infty} \phi_{hh}[k]\phi_{xx}[m-k] + \phi_{\eta\eta}[m], \tag{2.1.12a}$$

$$\phi_{xy}[m] = \sum_{k=-\infty}^{\infty} h[k]\phi_{xx}[m+k], \tag{2.1.12b}$$

where $\phi_{hh}[k]$ is the deterministic autocorrelation of the channel response $h[n]$. Equation (2.1.12a) in particular implies that if $h[n]$ is nearly sparse or decays at a certain rate and if the correlation times of $x[n]$ and $\eta[n]$ are short, then the autocorrelation $\phi_{yy}[m]$ and the matrix $\mathbf{Q}$ also tend to be nearly sparse or decay at the same rate. The formulation in this subsection can be extended in a straightforward manner to more elaborate equalization techniques such as decision-feedback equalization, channel shortening, and MIMO systems; see [90] for more details on these extensions.

Under the complex-baseband equivalent channel model for quadrature-amplitude modulation (QAM), all of the quantities above become complex-valued, including the equalizer coefficients $b_n$, and $\mathbf{Q}$ becomes Hermitian positive definite. We can accommodate complex-valued coefficients within our real-valued framework by separating the real and imaginary parts of $\mathbf{b}$ to create a $2N$-dimensional vector $\widetilde{\mathbf{b}}$ of real coefficients. The vector $\mathbf{c}$, which is still equal to $\mathbf{Q}^{-1}\mathbf{f}$, is transformed similarly. If the real and imaginary components of $\mathbf{b}$ are interleaved in $\widetilde{\mathbf{b}}$, i.e., $\widetilde{\mathbf{b}} = \begin{bmatrix}\mathrm{Re}(b_1) & \mathrm{Im}(b_1) & \mathrm{Re}(b_2) & \mathrm{Im}(b_2) & \dots\end{bmatrix}^T$, then the corresponding transformation for $\mathbf{Q}$ is to replace each complex-valued entry $\mathbf{Q}_{mn}$ with the $2 \times 2$ submatrix

$$\begin{bmatrix}\mathrm{Re}(Q_{mn}) & -\mathrm{Im}(Q_{mn}) \\ \mathrm{Im}(Q_{mn}) & \mathrm{Re}(Q_{mn})\end{bmatrix}.$$

The resulting matrix $\widetilde{\mathbf{Q}}$ will generally have the same sparsity and decay properties as $\mathbf{Q}$. The zero-norm $\left\|\widetilde{\mathbf{b}}\right\|_0$ now measures the number of non-zero real and imaginary components of $\mathbf{b}$ counted separately as opposed to the number of non-zero components of $\mathbf{b}$ as a complex vector. Counting the number of non-zero real and imaginary components separately is a reasonable metric because the cost of implementation is usually determined by the number of operations on real numbers, even for complex-valued filters. As an example, if the real part of a coefficient is zero, multiplication by that coefficient requires only two real multiplications instead of the usual four.

### 2.1.3 Signal detection

The design of sparse filters for use in signal detection can also be formulated as in (2.0.1). We assume that a signal $s[n]$ is to be detected in the presence of stationary zero-mean additive noise $\eta[n]$ with autocorrelation $\phi_{\eta\eta}[m]$. The received signal is processed with an

FIR filter of length $N$ and sampled at $n = N - 1$, yielding

$$y[N-1] = \sum_{n=0}^{N-1} b_n \left( s[N-1-n] + \eta[N-1-n] \right)$$

when the signal is present. The filter coefficients $b_n$ are chosen such that the SNR is greater than a pre-specified threshold $\rho$, where the SNR is defined as the ratio of the mean of $y[N-1]$ given that the signal is present to the standard deviation of $y[N-1]$, the latter being the same under both hypotheses. By defining $\mathbf{s} \in \mathbb{R}^N$ and $\mathbf{R} \in \mathbb{R}^{N \times N}$ according to $s_n = s[N-1-n]$ and $R_{mn} = \phi_{\eta\eta}\left[|m-n|\right]$, the problem of sparse design can be expressed as

$$\min_{\mathbf{b}} \quad \|\mathbf{b}\|_0 \qquad \text{s.t.} \qquad \frac{\mathbf{s}^T\mathbf{b}}{\sqrt{\mathbf{b}^T\mathbf{R}\mathbf{b}}} \geq \rho. \tag{2.1.13}$$

While the SNR constraint in (2.1.13) cannot be rewritten directly in the form of (2.1.3), we show that problems (2.1.13) and (2.0.1) can be made equivalent in the sense of having the same optimal solutions. To establish the equivalence, we determine conditions under which feasible solutions to (2.0.1) and (2.1.13) exist when an arbitrarily chosen subset of coefficients, represented by the index set $\mathcal{Z}$, is constrained to have value zero. Given $b_n = 0$ for $n \in \mathcal{Z}$ and with $\mathcal{Y}$ denoting the complement of $\mathcal{Z}$, (2.1.3) becomes

$$\mathbf{b}_{\mathcal{Y}}^T \mathbf{Q}_{\mathcal{Y}\mathcal{Y}} \mathbf{b}_{\mathcal{Y}} - 2\mathbf{f}_{\mathcal{Y}}^T \mathbf{b}_{\mathcal{Y}} \leq \beta, \tag{2.1.14}$$

where $\mathbf{b}_{\mathcal{Y}}$ is the $|\mathcal{Y}|$-dimensional vector formed from the entries of $\mathbf{b}$ indexed by $\mathcal{Y}$ (similarly for other vectors), and $\mathbf{Q}_{\mathcal{Y}\mathcal{Y}}$ is the $|\mathcal{Y}| \times |\mathcal{Y}|$ matrix formed from the rows and columns of $\mathbf{Q}$ indexed by $\mathcal{Y}$ (similarly for other matrices). We consider minimizing the left-hand side of (2.1.14) with respect to $\mathbf{b}_{\mathcal{Y}}$. If the minimum value is greater than $\beta$, then (2.1.14) cannot be satisfied for any value of $\mathbf{b}_{\mathcal{Y}}$ and a feasible solution with $b_n = 0$, $n \in \mathcal{Z}$ cannot exist. It is straightforward to show by differentiation that the minimum occurs at $\mathbf{b}_{\mathcal{Y}} = \left(\mathbf{Q}_{\mathcal{Y}\mathcal{Y}}\right)^{-1}\mathbf{f}_{\mathcal{Y}}$, and consequently the condition for feasibility is

$$-\mathbf{f}_{\mathcal{Y}}^T \left(\mathbf{Q}_{\mathcal{Y}\mathcal{Y}}\right)^{-1} \mathbf{f}_{\mathcal{Y}} \leq \beta. \tag{2.1.15}$$

We refer to an index set $\mathcal{Y}$ (equivalently its complement $\mathcal{Z}$) as being feasible if (2.1.15) is satisfied.

Similarly in the case of problem (2.1.13), a subset $\mathcal{Y}$ is feasible if and only if the modified constraint

$$\frac{\mathbf{s}_{\mathcal{Y}}^T \mathbf{b}_{\mathcal{Y}}}{\sqrt{\mathbf{b}_{\mathcal{Y}}^T \mathbf{R}_{\mathcal{Y}\mathcal{Y}} \mathbf{b}_{\mathcal{Y}}}} \geq \rho$$

is satisfied when the left-hand side is maximized. The maximizing values of $\mathbf{b}_{\mathcal{Y}}$ are proportional to $(\mathbf{R}_{\mathcal{Y}\mathcal{Y}})^{-1} \mathbf{s}_{\mathcal{Y}}$ and correspond to the whitened matched filter for the partial signal $\mathbf{s}_{\mathcal{Y}}$ (a.k.a. the restricted-length matched filter in [45]). The resulting feasibility condition is $\sqrt{\mathbf{s}_{\mathcal{Y}}^T (\mathbf{R}_{\mathcal{Y}\mathcal{Y}})^{-1} \mathbf{s}_{\mathcal{Y}}} \geq \rho$, or after squaring,

$$\mathbf{s}_{\mathcal{Y}}^T (\mathbf{R}_{\mathcal{Y}\mathcal{Y}})^{-1} \mathbf{s}_{\mathcal{Y}} \geq \rho^2. \tag{2.1.16}$$

Condition (2.1.16) is identical to (2.1.15) for all $\mathcal{Y}$ with the identifications $\mathbf{Q} = \mathbf{R}$, $\mathbf{f} = \mathbf{s}$, and $\beta = -\rho^2$. It follows that an index set $\mathcal{Y}$ is feasible for problem (2.1.13) exactly when it is feasible for problem (2.0.1), and therefore the optimal index sets for (2.0.1) and (2.1.13) coincide.

One application of the basic detection problem above is in minimum-variance distortionless response (MVDR) beamforming in array processing (see [91] for background). In this context, the target signal $\mathbf{s}$ is defined by a direction of interest, $\mathbf{R}$ is the correlation matrix of the array output, and the mean-squared value of the array output is minimized subject to a unit-gain constraint on signals propagating in the chosen direction. To fit the present formulation, the mean-squared output is bounded instead of being minimized, which is equivalent to bounding the SNR as in (2.1.13). Section 4.2.3 presents an example of designing sparse MVDR beamformers.

In the problems discussed in this section, the assumption of stationarity is not necessary for equivalence with the abstract problem (2.0.1). In the absence of stationarity, the values of $\mathbf{Q}$, $\mathbf{f}$, and $\beta$ may vary with time, resulting in a succession of instances of (2.0.1).

It has been shown in this section that several filter design problems can be formulated in the form of (2.0.1). Accordingly, in the remainder of this chapter and in Chapter 3, we focus on the solution of (2.0.1). To apply the methods to be developed to a specific design problem, it suffices to determine the values of the parameters $\mathbf{Q}$, $\mathbf{f}$, $\beta$ or $\mathbf{Q}$, $\mathbf{c}$, $\gamma$ using the expressions provided in this section.

## 2.2 Special cases

In general, problem (2.0.1) is a difficult combinatorial optimization problem for which no polynomial-time algorithm is known. Efficient and exact solutions do exist however when the matrix $\mathbf{Q}$ has special structure. In this section, we discuss several such examples in which $\mathbf{Q}$ is diagonal, block-diagonal, or banded.

The methods presented in this section solve (2.0.1) by determining for each $K = 1, 2, \ldots$ whether a feasible solution with $K$ zero-valued coefficients exists. To derive a condition for the existence of a solution with a given number of zero components, we start from (2.1.15), which specifies whether a solution exists when a specific subset $\mathcal{Z}$ of coefficients is constrained to have zero value. Condition (2.1.15) may be extended to take into account all possible subsets of a given size using an argument similar to that made in deriving (2.1.15). Specifically, if the minimum value of the left-hand side of (2.1.15) taken over all subsets $\mathcal{Y}$ of size $N - K$ is greater than $\beta$, then no such subset $\mathcal{Y}$ is feasible and there can be no solution with $K$ zero-valued entries. After a sign change, this gives the condition

$$\max_{|\mathcal{Y}|=N-K} \left\{ \mathbf{f}_{\mathcal{Y}}^{T} (\mathbf{Q}_{\mathcal{Y}\mathcal{Y}})^{-1} \mathbf{f}_{\mathcal{Y}} \right\} \geq -\beta \tag{2.2.1}$$

for the existence of a feasible solution with $K$ zero-valued components. The number of subsets $\mathcal{Y}$ of size $N - K$ is $\binom{N}{K}$, which can be very large, and in the general case a tractable way of maximizing over all choices of $\mathcal{Y}$ is not apparent. However, for the special cases considered in this section, (2.2.1) can be evaluated efficiently.

We will find it convenient to express the conditions in (2.1.15) and (2.2.1) in terms of the set $\mathcal{Z}$ rather than $\mathcal{Y}$, especially when $\mathcal{Z}$ is smaller than $\mathcal{Y}$. With $b_n = 0$ for $n \in \mathcal{Z}$, the quadratic constraint (2.1.1) becomes

$$\begin{aligned} &\begin{bmatrix} (\mathbf{b}_{\mathcal{Y}} - \mathbf{c}_{\mathcal{Y}})^T & -\mathbf{c}_{\mathcal{Z}}^T \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{\mathcal{Y}\mathcal{Y}} & \mathbf{Q}_{\mathcal{Y}\mathcal{Z}} \\ \mathbf{Q}_{\mathcal{Z}\mathcal{Y}} & \mathbf{Q}_{\mathcal{Z}\mathcal{Z}} \end{bmatrix} \begin{bmatrix} \mathbf{b}_{\mathcal{Y}} - \mathbf{c}_{\mathcal{Y}} \\ -\mathbf{c}_{\mathcal{Z}} \end{bmatrix} \\ &\qquad = (\mathbf{b}_{\mathcal{Y}} - \mathbf{c}_{\mathcal{Y}})^T \mathbf{Q}_{\mathcal{Y}\mathcal{Y}} (\mathbf{b}_{\mathcal{Y}} - \mathbf{c}_{\mathcal{Y}}) - 2\mathbf{c}_{\mathcal{Z}}^T \mathbf{Q}_{\mathcal{Z}\mathcal{Y}} (\mathbf{b}_{\mathcal{Y}} - \mathbf{c}_{\mathcal{Y}}) + \mathbf{c}_{\mathcal{Z}}^T \mathbf{Q}_{\mathcal{Z}\mathcal{Z}} \mathbf{c}_{\mathcal{Z}} \leq \gamma, \end{aligned} \tag{2.2.2}$$

where $\mathbf{Q}_{\mathcal{Y}\mathcal{Z}}$ denotes the submatrix of $\mathbf{Q}$ with rows indexed by $\mathcal{Y}$ and columns indexed by $\mathcal{Z}$. As in the derivation of (2.1.15), we minimize the left-hand side of (2.2.2) with respect to $\mathbf{b}_{\mathcal{Y}}$ to obtain a condition for feasibility. The minimizer is given by $\mathbf{b}_{\mathcal{Y}} - \mathbf{c}_{\mathcal{Y}} = (\mathbf{Q}_{\mathcal{Y}\mathcal{Y}})^{-1} \mathbf{Q}_{\mathcal{Y}\mathcal{Z}} \mathbf{c}_{\mathcal{Z}}$, resulting in

$$\mathbf{c}_{\mathcal{Z}}^T(\mathbf{Q}/\mathbf{Q}_{\mathcal{YY}})\mathbf{c}_{\mathcal{Z}} \leq \gamma, \tag{2.2.3}$$

where $\mathbf{Q}/\mathbf{Q}_{\mathcal{YY}}$ is the Schur complement of $\mathbf{Q}_{\mathcal{YY}}$, defined as [92]

$$\mathbf{Q}/\mathbf{Q}_{\mathcal{YY}} = \mathbf{Q}_{\mathcal{ZZ}} - \mathbf{Q}_{\mathcal{ZY}}(\mathbf{Q}_{\mathcal{YY}})^{-1}\mathbf{Q}_{\mathcal{YZ}} = \left(\left(\mathbf{Q}^{-1}\right)_{\mathcal{ZZ}}\right)^{-1}. \tag{2.2.4}$$

Condition (2.2.3) is equivalent to (2.1.15). Similarly, the counterpart to (2.2.1) is

$$\min_{|\mathcal{Z}|=K} \left\{\mathbf{c}_{\mathcal{Z}}^T(\mathbf{Q}/\mathbf{Q}_{\mathcal{YY}})\mathbf{c}_{\mathcal{Z}}\right\} \leq \gamma. \tag{2.2.5}$$

### 2.2.1 Diagonal Q

The first example we consider is that of diagonal $\mathbf{Q}$, which arises in certain special cases of the problems presented in Section 2.1. For example, in least-squares filter design with uniform weighting ($W(\omega) = 1$ in (2.1.5)), (2.1.6a) implies that $\mathbf{Q} = \mathbf{I}$. In the estimation problem, if the observations $y[n]$ are white, then $\mathbf{Q}$ in (2.1.10a) is proportional to $\mathbf{I}$. Similarly, $\mathbf{R}$ is proportional to $\mathbf{I}$ in the detection problem when the noise is white.

Assuming $\mathbf{Q}$ is diagonal, $\mathbf{Q}/\mathbf{Q}_{\mathcal{YY}} = \mathbf{Q}_{\mathcal{ZZ}}$ and (2.2.5) simplifies to

$$\min_{|\mathcal{Z}|=K} \left\{\sum_{n\in\mathcal{Z}} Q_{nn}c_n^2\right\} \leq \gamma. \tag{2.2.6}$$

The solution to the minimization is to choose $\mathcal{Z}$ to correspond to the $K$ smallest values of $Q_{nn}c_n^2$. Letting $\Sigma_K(\{Q_{nn}c_n^2\})$ denote the sum of the $K$ smallest $Q_{nn}c_n^2$, (2.2.6) becomes

$$\Sigma_K\left(\{Q_{nn}c_n^2\}\right) \leq \gamma. \tag{2.2.7}$$

Problem (2.0.1) can be solved in the diagonal case by checking condition (2.2.7) for successively increasing values of $K$. The minimum zero-norm is given by $N - K^*$, where $K^*$ is the largest value of $K$ for which (2.2.7) holds. One particular optimal solution results from setting $b_n = c_n$ for $n$ corresponding to the $N - K^*$ largest $Q_{nn}c_n^2$, and $b_n = 0$ otherwise. This solution has an intuitive interpretation in the context of the problems discussed in Section 2.1. In least-squares filter design with $W(\omega) = 1$, we have $f_n = d[n]$ from (2.1.6b) and $c_n = f_n$. Thus the solution is to match the $N - K^*$ largest values of the desired impulse response $d[n]$ and have zeros in the remaining positions. In the estimation problem with white observations, $c_n \propto f_n = \phi_{xy}[n]$, and hence the cross-correlation between $x[n]$ and $y[n]$ plays the role of the desired impulse response. Similarly, in the detection problem with white noise, the largest values of the signal $s[n]$ are matched. If $y[n]$ or $\eta[n]$ is white but non-stationary, the matrices $\mathbf{Q}$ and $\mathbf{R}$ remain diagonal and the solution takes into account any weighting due to a time-varying variance.

### 2.2.2 Block-diagonal Q

A generalization of the diagonal structure considered in Section 2.2.1 is the case of block-diagonal $\mathbf{Q}$. In the problems discussed in Section 2.1, $\mathbf{Q}$ often represents a covariance matrix and is therefore block-diagonal if the underlying random process can be partitioned into subsets of variables with the property that variables from different subsets are uncorrelated. This may occur for example in a sensor array in which the sensors occur in clusters separated by large distances. We note that the presence of block-diagonal structure precludes stationarity except in the pure diagonal case addressed in Section 2.2.1. This is because stationarity implies that $\mathbf{Q}$ is Toeplitz, whereas block-diagonality implies that every diagonal of $\mathbf{Q}$ other than the main diagonal includes at least one zero-valued entry, and hence the only matrices satisfying both properties are multiples of $\mathbf{I}$.

We assume that $\mathbf{Q}$ has the following form:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_1 & & & \\ & \mathbf{Q}_2 & & \\ & & \ddots & \\ & & & \mathbf{Q}_L \end{bmatrix}, \tag{2.2.8}$$

where each block $\mathbf{Q}_b$, $b = 1, 2, \ldots, L$, is of dimension $N_b \times N_b$, $\sum_b N_b = N$, and indices have been permuted if necessary to convert $\mathbf{Q}$ to block-diagonal form. For an arbitrary index set $\mathcal{Y}$, let $\mathcal{Y}_b$ be the set of indices in $\mathcal{Y}$ that correspond to the $b$th block. Then

$$\mathbf{Q}_{\mathcal{Y}\mathcal{Y}} = \begin{bmatrix} \mathbf{Q}_{\mathcal{Y}_1\mathcal{Y}_1} & & & \\ & \mathbf{Q}_{\mathcal{Y}_2\mathcal{Y}_2} & & \\ & & \ddots & \\ & & & \mathbf{Q}_{\mathcal{Y}_L\mathcal{Y}_L} \end{bmatrix},$$

which is also block-diagonal. Hence the maximization in (2.2.1) can be rewritten as

$$\begin{aligned} \max \quad & \sum_{b=1}^{L} \mathbf{f}_{\mathcal{Y}_b}^T (\mathbf{Q}_{\mathcal{Y}_b \mathcal{Y}_b})^{-1} \mathbf{f}_{\mathcal{Y}_b} \\ \text{s.t.} \quad & \sum_{b=1}^{L} |\mathcal{Y}_b| = N - K. \end{aligned} \tag{2.2.9}$$

The maximization in (2.2.9) can be solved via dynamic programming. To derive the dynamic programming recursion, define $V_g(M)$ to be the maximum value over all subsets $\mathcal{Y}$ of size $M$ that are confined to the first $g$ blocks, i.e.,

$$\begin{aligned} V_g(M) = \max \quad & \sum_{b=1}^{g} \mathbf{f}_{\mathcal{Y}_b}^T (\mathbf{Q}_{\mathcal{Y}_b \mathcal{Y}_b})^{-1} \mathbf{f}_{\mathcal{Y}_b} \\ \text{s.t.} \quad & \sum_{b=1}^{g} |\mathcal{Y}_b| = M \end{aligned} \tag{2.2.10}$$

for $g = 1, 2, \dots, L$. The maximum value in (2.2.9) is thus $V_L(N - K)$. Also define $v_b(M_b)$ to be the maximum value over subsets of size $M_b$ restricted to the $b$th block,

$$v_b(M_b) = \max_{|\mathcal{Y}_b| = M_b} \mathbf{f}_{\mathcal{Y}_b}^T (\mathbf{Q}_{\mathcal{Y}_b \mathcal{Y}_b})^{-1} \mathbf{f}_{\mathcal{Y}_b}, \quad b = 1, \dots, L, \quad M_b = 0, 1, \dots, N_b. \tag{2.2.11}$$

It follows that $V_1(M) = v_1(M)$. For $g = 2, \dots, L$, $V_g(M)$ may be computed through the following recursion:

$$V_g(M) = \max_{M_g = 0, 1, \dots, \min(M, N_g)} \left\{ v_g(M_g) + V_{g-1}(M - M_g) \right\}. \tag{2.2.12}$$

Equation (2.2.12) states that the maximum value after $g$ blocks may be obtained by optimally allocating $M_g$ indices to the $g$th block, optimally allocating the remaining $M - M_g$ indices to the first $g - 1$ blocks, and then maximizing over all choices of $M_g$ between 0 and $\min(M, N_g)$ ($M_g$ cannot exceed $N_g$, the total number of indices in block $g$).

The dynamic programming procedure outlined above involves carrying out the recursion in (2.2.12) as well as computing the values of $v_b(M_b)$ in (2.2.11) for each block. We consider first the computational complexity of the recursion. We assume that (2.2.9) is to be evaluated for $N - K = 0, 1, \dots, M_0$ and that $M_0$ grows proportionally with $N$. This requires the calculation of $V_g(M)$ for $g = 2, \dots, L$ and $M = 0, \dots, M_0$ at most. For fixed $g$

and $M$, the maximization in (2.2.12) requires at most $N_g + 1$ additions and comparisons. Therefore the total number of operations is proportional to

$$\sum_{g=2}^{L} \sum_{M=0}^{M_0} (N_g + 1) = (M_0 + 1)(N - N_1 + L - 1) \sim \mathcal{O}(N^2).$$

In comparison, the computational complexity of computing $v_b(M_b)$ can be much higher. Assuming again that the values of $V_L(0), \ldots, V_L(M_0)$ are to be determined, we require the values of $v_b(M_b)$ for $b = 1, \ldots, L$ and $M_b = 0, 1, \ldots, \min(M_0, N_b)$. In the worst case, if $M_0 \geq N_b$ for all $b$, subsets of all possible sizes need to be considered within each block, resulting in $2^{N_b}$ subsets for the $b$th block. Since the matrix inversion in (2.2.11) takes $\mathcal{O}(N_b^3)$ operations and dominates the computation of each $v_b(M_b)$, the total number of operations is $\mathcal{O}\left(\sum_{b=1}^{L} N_b^3 2^{N_b}\right)$. Thus the block sizes $N_b$ must be small in an absolute sense in order for the dynamic programming algorithm to be efficient. However, even if the block sizes are not small, using dynamic programming in the block-diagonal case still offers computational savings relative to an exhaustive evaluation of (2.2.1) for $N - K = 0, \ldots, M_0$. The latter requires on the order of $2^N$ matrix inversions, one for each subset up to size $M_0$, for a total complexity that scales as $N^3 2^N$. It can be shown that

$$2^N \geq \sum_{b=1}^{L} 2^{N_b} \quad \text{for} \quad N \geq 2, \quad \sum_{b=1}^{L} N_b = N, \quad N_b \geq 1,$$

which implies that dynamic programming has a lower order of growth.

### 2.2.3 Banded Q

Another generalization of the diagonal case is to consider banded matrices. We assume in this subsection that the non-zero entries of $\mathbf{Q}$ are restricted to a band of width $2W + 1$ centered on the main diagonal with $W < N$, again after a permutation of indices if necessary. In the applications discussed in Section 2.1, $\mathbf{Q}$ matrices with banded structure can occur if the correlation distance of the random processes $x[n]$, $y[n]$ or $\eta[n]$ is small. Specifically, a bandwidth of $2W + 1$ for $\mathbf{Q}$ implies that any two variables with indices differing by more than $W$ are uncorrelated.

The maximization problem in (2.2.1) with banded $\mathbf{Q}$ was considered by Sestok in [2], who developed a dynamic programming algorithm to exploit the banded structure. This algorithm is similar in spirit to the one presented in Section 2.2.2 for block-diagonal $\mathbf{Q}$. In the block-diagonal case, the quadratic form in (2.2.1) can be decomposed into a sum of quadratic forms of smaller dimension if the subset $\mathcal{Y}$ spans multiple blocks. The algorithm of [2] relies upon a similar decomposition in the banded case: if $\mathbf{Q}$ has bandwidth $2W + 1$, a decomposition is possible if $\mathcal{Y}$ can be partitioned into multiple subsets such that any two indices taken from different subsets differ by more than $W$.

Sestok showed that the computational complexity of the dynamic programming algorithm is $\mathcal{O}(N^5)$ in the tridiagonal case, i.e., for $W = 1$. While a generalization to larger bandwidths was described in [2], the increase in computational complexity was not explicitly characterized. We now claim that even in the pentadiagonal case ($W = 2$), the complexity grows exponentially with $N$. Our argument is based on the number of quadratic forms that must be evaluated in the course of the algorithm. Specifically, we need to consider quadratic forms as in (2.1.15) for all subsets $\mathcal{Y}$ up to a certain size and with the property that when the indices in $\mathcal{Y}$ are placed in order, no two adjacent indices differ by more than $W$. The values of these quadratic forms play a role analogous to the values of $v_b(M_b)$ in (2.2.11) in the block-diagonal case. For $W = 1$, the subsets in question are composed of consecutive indices and there are $\mathcal{O}(N^2)$ of them. For $W = 2$, the number of subsets grows as $N \cdot 2^{\alpha N}$, where $\alpha$ is a constant between 0 and 1. A detailed counting argument is presented in Appendix A.1. As a consequence, the complexity of the dynamic programming algorithm of [2] increases exponentially with $N$ for $W = 2$, and by extension, for $W > 2$ as well. As in the block-diagonal case, the rate of growth for dynamic programming is still lower than that of exhaustive evaluation of (2.2.1), which involves on the order of $2^N$ subsets.

A variation on the banded case is that of $\mathbf{Q}^{-1}$ being banded. Unlike in the diagonal or block-diagonal cases, $\mathbf{Q}$ having a certain bandwidth does not imply that $\mathbf{Q}^{-1}$ has the same bandwidth, and vice versa. If $\mathbf{Q}^{-1}$ has low bandwidth, we may use the alternative condition in (2.2.5) instead of (2.2.1). Sestok's algorithm then applies with $\mathbf{Q}$ replaced by $\mathbf{Q}^{-1}$, $\mathbf{f}$ by $\mathbf{c}$, and maximization by minimization. In the case of (2.2.5), it is the number of zero coefficients, $K$, that is incremented as opposed to the number of non-zero coefficients $M$.

### 2.2.4 Challenges in generalizing to unstructured $\mathbf{Q}$

In Sections 2.2.1–2.2.3, we have seen several special cases in which the structure of the matrix $\mathbf{Q}$ allows for an efficient solution to problem (2.0.1). It is natural to ask whether instances with unstructured matrices can be solved through a transformation into one of the special cases. In particular, given that any symmetric matrix can be diagonalized by a unitary transformation, one may be led to consider the possibility of exploiting such transformations to reduce the general problem to the diagonal case of Section 2.2.1. In this subsection, we give some indications as to why this approach to generalization does not appear to be straightforward.

In the context of the applications in Section 2.1, one way of reducing an instance to the diagonal case is to apply whitening. In the estimation problem, the whitening is done on the observations $y[n]$, while in the detection problem, it is the noise $\eta[n]$ that is whitened. The process of whitening, however, requires additional processing of the input, often in the form of a prefilter. The task then shifts to designing an efficient whitening prefilter that does not add significantly to the total implementation cost. Moreover, since the whitening is likely to be imperfect, further measures may be needed. There are also applications in which cascade structures are not applicable, e.g. arrays.

A different approach that we explore in greater depth is to solve a sparsity maximization problem such as (2.0.1) by first transforming the feasible set into one that is easier to optimize over and then inverting the transformation to map the solution found in the transformed space to one in the original space. For this procedure to guarantee an optimal solution to the original problem, it is necessary that the transformation preserve the ordering of vectors by the number of non-zero components. We give a negative result stating that the only invertible linear transformations that preserve ordering by sparsity in a global sense are composed of diagonal scalings and permutations. The transformations in this class are therefore rather limited; in particular, most dense matrices cannot be transformed into diagonal, block-diagonal, or banded matrices using diagonal scalings and permutations alone.

To state the result more precisely, let $\mathbf{T} : \mathbb{R}^N \mapsto \mathbb{R}^M$ be a linear transformation that maps the original ellipsoid specified by (2.1.1), which we denote by $\mathcal{E}_{\mathbf{Q}}$, to its image $\mathbf{T}(\mathcal{E}_{\mathbf{Q}})$. We assume that $\mathcal{E}_{\mathbf{Q}}$ is full-dimensional, i.e., it has non-zero extent along every axis, which corresponds to all of the eigenvalues of $\mathbf{Q}$ being finite. The assumption of full-dimensionality requires that rank$(\mathbf{T}) = N$, as otherwise $\mathbf{T}(\mathcal{E}_\mathbf{Q})$ would be contained in a subspace of dimension less than $N$ and could not be mapped back to $\mathcal{E}_\mathbf{Q}$ through a linear transformation. Then for $M = N$, $\mathbf{T}$ has an inverse, and for $M > N$, $\mathbf{T}$ has a left-inverse, both of which will be denoted as $\mathbf{T}^{-1}$.

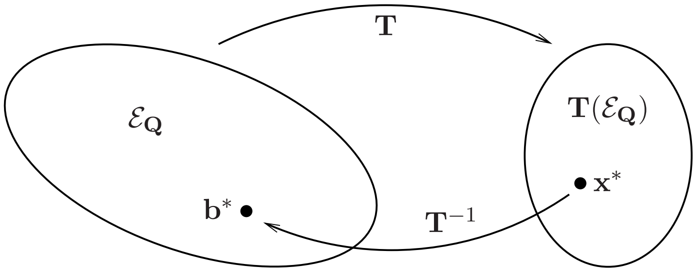


Figure 2-3: Solving problem (2.0.1) through a linear transformation.

We consider solving (2.0.1) by first determining an optimal solution $\mathbf{x}^*$ to the transformed problem

$$\min_{\mathbf{x}\in\mathbf{T}(\mathcal{E}_\mathbf{Q})} \quad \|\mathbf{x}\|_0\,,$$

and then computing a solution to (2.0.1) as $\mathbf{b}^* = \mathbf{T}^{-1}\mathbf{x}^*$. This process is represented graphically in Fig. 2-3. Requiring $\mathbf{b}^*$ to be optimal over $\mathcal{E}_\mathbf{Q}$ is the same as the condition

$$\left\|\mathbf{T}^{-1}\mathbf{x}\right\|_0 \geq \left\|\mathbf{T}^{-1}\mathbf{x}^*\right\|_0 \quad \forall\, \mathbf{x} \in \mathbf{T}(\mathcal{E}_\mathbf{Q}),$$

since $\mathcal{E}_\mathbf{Q}$ can be equivalently thought of as the image of $\mathbf{T}(\mathcal{E}_\mathbf{Q})$ under $\mathbf{T}^{-1}$. Given that $\mathbf{x}^*$ is not known a priori, to guarantee optimality it is natural to impose a similar order-preserving condition for arbitrary pairs of vectors in $\mathbf{T}(\mathcal{E}_\mathbf{Q})$, i.e.,

$$\|\mathbf{x}_1\|_0 \geq \|\mathbf{x}_2\|_0 \quad \Longrightarrow \quad \left\|\mathbf{T}^{-1}\mathbf{x}_1\right\|_0 \geq \left\|\mathbf{T}^{-1}\mathbf{x}_2\right\|_0 \quad \forall\, \mathbf{x}_1, \mathbf{x}_2 \in \mathbf{T}(\mathcal{E}_\mathbf{Q}).$$

We show that if the previous condition is extended to all of $\mathbb{R}^M$, i.e.,

$$\|\mathbf{x}_1\|_0 \geq \|\mathbf{x}_2\|_0 \quad \Longrightarrow \quad \left\|\mathbf{T}^{-1}\mathbf{x}_1\right\|_0 \geq \left\|\mathbf{T}^{-1}\mathbf{x}_2\right\|_0 \quad \forall\, \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^M, \tag{2.2.13}$$

then $\mathbf{T}$ must be a composition of a diagonal scaling and a permutation.

**Theorem 1.** *If $\mathbf{T} : \mathbb{R}^N \mapsto \mathbb{R}^M$ is a linear transformation with left-inverse $\mathbf{T}^{-1}$ satisfying*

(2.2.13)*, then* $M$ *is necessarily equal to* $N$ *and* $\mathbf{T}$ *is a composition of a diagonal scaling and a permutation of the coordinates.*

The proof of Theorem 1 can be found in Appendix A.2. Since the only effect of a permutation is to relabel coordinates, we may assume without loss of generality that the permutation in $\mathbf{T}$ is the identity permutation. This leaves diagonal scalings as the only invertible linear transformations that preserve ordering by sparsity in the sense of (2.2.13). Since diagonal scalings are insufficient to transform a dense $\mathbf{Q}$ matrix into one that is diagonal, block-diagonal, or banded, it is not possible to reduce problem (2.0.1) in its general form to one of the special cases using a linear transformation of this type.

### 2.2.5 Generalization to separable non-quadratic constraint functions

It is possible to generalize the special cases in Sections 2.2.1–2.2.3 to a larger class of sparsity maximization problems involving constraint functions that are not necessarily quadratic. For instance, a generalization of the diagonal case is to have a single constraint that can be separated into a sum of univariate functions $F_n$, each taking as input a single coefficient $b_n$.[1] The problem takes the form

$$\min_{\mathbf{b}} \quad \|\mathbf{b}\|_0 \qquad \text{s.t.} \qquad \sum_{n=0}^{N-1} F_n(b_n) \leq \gamma, \tag{2.2.14}$$

where $F_n(b_n) = Q_{nn}(b_n - c_n)^2$ in the quadratic case. For index sets $\mathcal{Z}$ and $\mathcal{Y}$ defined as before, the analogue to (2.2.6) is

$$\min_{|\mathcal{Z}|=K} \left\{ \sum_{n\in\mathcal{Z}} F_n(0) + \sum_{n\in\mathcal{Y}} \min_{b_n} F_n(b_n) \right\} \leq \gamma, \tag{2.2.15}$$

where the second term on the left-hand side is zero in the quadratic case. The minimum value of the left-hand side of (2.2.15) is attained by choosing $\mathcal{Z}$ to correspond to the $K$ smallest differences $F_n(0) - \min F_n(b_n)$, leading to a generalization of (2.2.7).

A similar generalization of the block-diagonal case would involve a constraint function that is block-separable, i.e., expressible as $\sum_{b=1}^{L} F_b(\mathbf{b}_b)$, where the vectors $\mathbf{b}_b$ are composed of disjoint subsets of coefficients. The banded case can be generalized by preserving the

[1] The separability can be with respect to a generalized notion of summation, e.g., a product of non-negative functions is also regarded as separable.

key property of decomposability given subsets that are separated by more than a certain number of indices.

The results of this section can be summarized as follows: In some special cases, specifically the diagonal, block-diagonal, and banded cases, problem (2.0.1) admits solutions that are both efficient and exact. Hence the methods discussed in this section should be used when the design parameters are such that the matrix $\mathbf{Q}$ belongs to one of the special classes. At the same time, the simplifications exploited in these special cases do not appear to be generalizable in a straightforward manner. In particular, it was argued in Section 2.2.4 that a dense $\mathbf{Q}$ matrix cannot be transformed into a diagonal, block-diagonal, or banded matrix by means of a sparsity-preserving linear transformation. Hence (2.0.1) remains a difficult problem in its general form.

## 2.3 Low-complexity algorithm for the general case

We now begin our discussion of the general case in which the matrix $\mathbf{Q}$ does not have any of the properties identified in the previous section. In keeping with the emphasis in this chapter on low-complexity algorithms, in this section we present a heuristic algorithm for solving (2.0.1) that we refer to as successive thinning. Optimal algorithms are discussed later in Chapter 3.

The basic idea in the algorithm is to successively thin a pre-designed, usually non-sparse filter by constraining more and more coefficients to zero while re-optimizing the remaining non-zero coefficients to compensate. Similar approaches were proposed in [2] for a slightly different problem formulation, in [41, 42] for channel equalization, and more generally for subset selection in regression [93]. Here the algorithm is adapted to problem (2.0.1) and an efficient implementation is described that does not require multiple matrix inversions.

As will be seen in Chapter 4, the successive thinning algorithm discussed in this section produces solutions that are in many instances either optimally sparse or close to optimal. Optimality or near-optimality is certified by running the branch-and-bound algorithm described in Chapter 3, which does guarantee an optimal solution. Thus the successive thinning algorithm is useful as a method for obtaining sparse solutions with relatively little computation.

We first give an overview of the algorithm before entering into a more detailed description. In Section 2.2 it was suggested that problem (2.0.1) may be solved by determining for each $K = 1, 2, \ldots, N$ whether a feasible solution with $K$ zero-valued coefficients exists, which is equivalent to checking condition (2.2.5). As $K$ increases from 1 (or decreases from $N$), the complexity of evaluating (2.2.5) grows as least as fast as $\binom{N}{K}$, the number of subsets of size $K$, and eventually becomes prohibitive. Successive thinning can be viewed as a simplification of the foregoing procedure. For $K = 1$, the successive thinning algorithm carries out the minimization in (2.2.5) exactly. We denote by $\mathcal{Z}^{(1)}$ the minimizing subset (in this case a single index). For $K = 2$, we restrict the minimization to only those pairs of indices that include $\mathcal{Z}^{(1)}$. Let $\mathcal{Z}^{(2)}$ represent the minimizer over this restricted collection of subsets of size 2. More generally for larger values of $K$, the subsets considered in the minimization are constrained to contain $\mathcal{Z}^{(K-1)}$, the minimizer for the previous value of $K$, thus limiting the search to adding one new index to $\mathcal{Z}^{(K-1)}$. Thus the algorithm resembles the class of greedy algorithms [94] in that decisions regarding zero-valued coefficients made in previous iterations are never revisited. The algorithm terminates when the minimum value corresponding to $\mathcal{Z}^{(K+1)}$ exceeds $\gamma$ for some $K$, at which point the last feasible subset $\mathcal{Z}^{(K)}$ becomes the final subset of zero-valued coefficients. Given $\mathcal{Z} = \mathcal{Z}^{(K)}$, we may then solve for the values of the non-zero coefficients to produce a feasible solution with zero-norm equal to $N - K$. It is often desirable in this last step to choose values that maximize performance.

The successive thinning algorithm greatly reduces the number of subsets that are explored compared to evaluating (2.2.5) exactly. The number of subsets evaluated in the $K$th stage is at most $N - K + 1$, corresponding to the $N - (K - 1)$ choices for the index to be added to $\mathcal{Z}^{(K-1)}$. Since the number of stages can be at most $N$, the total number of subsets grows only quadratically with $N$.

Successive thinning is guaranteed to result in a maximally sparse solution when the matrix $\mathbf{Q}$ is diagonal. From Section 2.2.1, the solution to the minimization in (2.2.5) in the diagonal case is to choose $\mathcal{Z}$ to correspond to the $K$ smallest $Q_{nn}c_n^2$. Since the subset of the $K$ smallest $Q_{nn}c_n^2$ is contained in the subset of the $K + 1$ smallest, the nesting property assumed by the algorithm is satisfied and the algorithm finds the true minimizing subsets. In other cases however, successive thinning does not appear to guarantee an optimal solution. Nevertheless, the examples in Section 4.1 demonstrate that the algorithm often yields optimal or near-optimal designs.

We now describe the algorithm in more detail. We use $\mathcal{Z}$ as above to represent the subset of coefficients constrained to zero. The complement of $\mathcal{Z}$ is now partitioned into two subsets $\mathcal{U}$ and $\mathcal{F}$. The subset $\mathcal{U}$ consists of those coefficients for which a value of zero is no longer feasible because of the zero-value constraints on coefficients in $\mathcal{Z}$, which restrict the feasible set. The subset $\mathcal{F}$ consists of the remaining coefficients for which a value of zero is still feasible. Each iteration of the algorithm is characterized by the assignment of variables to the subsets $\mathcal{Z}$, $\mathcal{U}$, and $\mathcal{F}$. For example, in the beginning no coefficients are constrained to zero, i.e., $\mathcal{Z} = \mathcal{U} = \emptyset$ and $\mathcal{F} = \{1, \ldots, N\}$. In subsequent iterations, both $\mathcal{Z}$ and $\mathcal{U}$ grow while $\mathcal{F}$ shrinks, giving rise to increasingly constrained versions of the original problem that we refer to as subproblems. To simplify the algorithm, we exploit the fact that every subproblem can be reduced to a lower-dimensional instance of the original problem (2.0.1). It is shown in Appendix A.3 that a subproblem defined by $(\mathcal{Z},\mathcal{U},\mathcal{F})$ can be expressed in the following form:

$$\begin{aligned} \min_{\mathbf{b}_{\mathcal{F}}} \quad & |\mathcal{U}| + \|\mathbf{b}_{\mathcal{F}}\|_0 \\ \text{s.t.} \quad & \left(\mathbf{b}_{\mathcal{F}} - \mathbf{c}_{\text{eff}}\right)^T \mathbf{Q}_{\text{eff}} \left(\mathbf{b}_{\mathcal{F}} - \mathbf{c}_{\text{eff}}\right) \leq \gamma_{\text{eff}}, \end{aligned} \tag{2.3.1}$$

where

$$\mathbf{Q}_{\text{eff}} = \mathbf{Q}_{\mathcal{Y}\mathcal{Y}}/\mathbf{Q}_{\mathcal{U}\mathcal{U}} = \mathbf{Q}_{\mathcal{F}\mathcal{F}} - \mathbf{Q}_{\mathcal{F}\mathcal{U}} \left(\mathbf{Q}_{\mathcal{U}\mathcal{U}}\right)^{-1} \mathbf{Q}_{\mathcal{U}\mathcal{F}}, \tag{2.3.2a}$$

$$\mathbf{c}_{\text{eff}} = \mathbf{c}_{\mathcal{F}} + \left(\mathbf{Q}_{\text{eff}}\right)^{-1}\left(\mathbf{Q}_{\mathcal{F}\mathcal{Z}} - \mathbf{Q}_{\mathcal{F}\mathcal{U}}(\mathbf{Q}_{\mathcal{U}\mathcal{U}})^{-1}\mathbf{Q}_{\mathcal{U}\mathcal{Z}}\right)\mathbf{c}_{\mathcal{Z}}, \tag{2.3.2b}$$

$$\gamma_{\text{eff}} = \gamma - \mathbf{c}_{\mathcal{Z}}^T(\mathbf{Q}/\mathbf{Q}_{\mathcal{Y}\mathcal{Y}})\mathbf{c}_{\mathcal{Z}}. \tag{2.3.2c}$$

In (2.3.1), the variables $b_n$ for $n \in \mathcal{Z}$ are absent because they have been set to zero and the variables $b_n$, $n \in \mathcal{U}$ have also been eliminated. The term $|\mathcal{U}|$ accounts for the cost of the non-zero variables in $\mathcal{U}$ and is a constant with respect to $\mathbf{b}_{\mathcal{F}}$. Hence subproblem (2.3.1) is of the same form as the original problem (2.0.1) with $|\mathcal{F}|$ variables instead of $N$ and effective parameters $\mathbf{Q}_{\text{eff}}$, $\mathbf{c}_{\text{eff}}$, and $\gamma_{\text{eff}}$ given by (2.3.2). As a consequence, each iteration of the algorithm after the first can be treated as if it were the first iteration acting on a lower-dimensional instance of (2.0.1).

In the remainder of this section, we use a superscript $K$ to label quantities associated with iteration $K$. In particular, $\mathbf{Q}^{(K)}$, $\mathbf{c}^{(K)}$, and $\gamma^{(K)}$ represent the parameters of the subproblem in iteration $K$. We also define $\mathbf{R}^{(K)} = \left(\mathbf{Q}^{(K)}\right)^{-1}$ and will find it more convenient to specify the computations in terms of $\mathbf{R}$ rather than $\mathbf{Q}$. The algorithm is initialized by setting $\mathcal{Z}^{(0)} = \mathcal{U}^{(0)} = \emptyset$, $\mathcal{F}^{(0)} = \{1, \ldots, N\}$, $\mathbf{R}^{(0)} = \mathbf{Q}^{-1}$, $\mathbf{c}^{(0)} = \mathbf{c}$, and $\gamma^{(0)} = \gamma$.

The first task in each iteration is to update the subsets $\mathcal{Z}$, $\mathcal{U}$, and $\mathcal{F}$. First we identify the coefficients in $\mathcal{F}^{(K)}$ that no longer yield feasible solutions when constrained to a value of zero. Determining whether a feasible solution exists when a single coefficient $b_n$ is set to zero can be done by specializing condition (2.2.3), which specifies when a subset $\mathcal{Z}$ of zero-valued coefficients is feasible. With $\mathcal{Z} = \{n\}$, (2.2.3) simplifies to

$$\frac{c_n^2}{\left(\mathbf{Q}^{-1}\right)_{nn}} \leq \gamma, \tag{2.3.3}$$

using the second definition of the Schur complement in (2.2.4). For the $K$th subproblem, $c_n$, $\left(\mathbf{Q}^{-1}\right)_{nn}$, and $\gamma$ in (2.3.3) are replaced by $c_n^{(K)}$, $R_{nn}^{(K)}$, and $\gamma^{(K)}$. The indices $n$ for which (2.3.3) is not satisfied correspond to coefficients for which a zero value is infeasible. We remove these indices from $\mathcal{F}^{(K)}$ and add them to $\mathcal{U}^{(K)}$ to form

$$\mathcal{U}^{(K+1)} = \mathcal{U}^{(K)} \cup \left\{ n \in \mathcal{F}^{(K)} : \frac{\left(c_n^{(K)}\right)^2}{R_{nn}^{(K)}} > \gamma^{(K)} \right\}. \tag{2.3.4}$$

If no indices remain in $\mathcal{F}^{(K)}$ after this removal, the filter cannot be thinned any further and the algorithm terminates. Otherwise, a new index $m \in \mathcal{F}^{(K)}$ is selected to be added to $\mathcal{Z}^{(K)}$, resulting in

$$\mathcal{Z}^{(K+1)} = \mathcal{Z}^{(K)} \cup \{m\}. \tag{2.3.5}$$

As described earlier, the index $m$ is chosen to minimize the left-hand side of (2.2.5) over all $\mathcal{Z}^{(K+1)}$ of the form in (2.3.5). In terms of the parameters of the current subproblem, this is equivalent to choosing

$$m = \arg \min_{n \in \mathcal{F}^{(K)}} \frac{\left(c_n^{(K)}\right)^2}{R_{nn}^{(K)}}. \tag{2.3.6}$$

The indices remaining in $\mathcal{F}^{(K)}$ after removing $m$ form the new subset $\mathcal{F}^{(K+1)}$.

The second task is to calculate the values of the new parameters $\mathbf{R}^{(K+1)}$, $\mathbf{c}^{(K+1)}$, and $\gamma^{(K+1)}$ from the current parameters $\mathbf{R}^{(K)}$, $\mathbf{c}^{(K)}$, and $\gamma^{(K)}$. We can adapt the results of Appendix A.3 for this purpose. In the present scenario, the $K$th (i.e., current) subproblem plays the role of the original problem, $\mathcal{Z} = \{m\}$ to represent the additional zero-value constraint, $\mathcal{U}$ is composed of the indices added to $\mathcal{U}^{(K)}$ in (2.3.4), and $\mathcal{F} = \mathcal{F}^{(K+1)}$. Equation

(2.3.2a) then gives $\mathbf{Q}^{(K+1)}$ in terms of $\mathbf{Q}^{(K)}$ after making the appropriate replacements. It can be shown that the equivalent recursion for $\mathbf{R}$ is

$$\mathbf{R}^{(K+1)} = \mathbf{R}^{(K)}_{\mathcal{F}^{(K+1)}\mathcal{F}^{(K+1)}} - \frac{1}{R^{(K)}_{mm}} \mathbf{R}^{(K)}_{\mathcal{F}^{(K+1)},m} \mathbf{R}^{(K)}_{m,\mathcal{F}^{(K+1)}}. \tag{2.3.7}$$

Similarly, (2.3.2b) shows how $\mathbf{c}^{(K+1)}$ may be determined from $\mathbf{c}^{(K)}$ and $\mathbf{Q}^{(K)}$. The equivalent formula using $\mathbf{R}^{(K)}$ instead of $\mathbf{Q}^{(K)}$ is

$$\mathbf{c}^{(K+1)} = \mathbf{c}^{(K)}_{\mathcal{F}^{(K+1)}} - \frac{c^{(K)}_{m}}{R^{(K)}_{mm}} \mathbf{R}^{(K)}_{\mathcal{F}^{(K+1)},m}. \tag{2.3.8}$$

Note that neither (2.3.7) nor (2.3.8) require the inversion of a matrix. Lastly, (2.3.2c) gives the following recursion for $\gamma$:

$$\gamma^{(K+1)} = \gamma^{(K)} - \frac{\left(c^{(K)}_{m}\right)^2}{R^{(K)}_{mm}}. \tag{2.3.9}$$

This completes the operations in iteration $K$. A summary of the algorithm is given below.

**Algorithm 1** Successive thinning for problem (2.0.1)

```
Input: Parameters $\mathbf{Q}$, $\mathbf{c}$, $\gamma$
Output: Sparse solution $\mathbf{b}$ to (2.0.1)
  Initialize: $K = 0$, $\mathcal{Z}^{(0)} = \mathcal{U}^{(0)} = \emptyset$, $\mathcal{F}^{(0)} = \{1, \dots, N\}$, $\mathbf{R}^{(0)} = \mathbf{Q}^{-1}$, $\mathbf{c}^{(0)} = \mathbf{c}$, $\gamma^{(0)} = \gamma$.
  Update $\mathcal{U}$ according to (2.3.4) and remove indices added to $\mathcal{U}^{(0)}$ from $\mathcal{F}^{(0)}$.
  while $\mathcal{F}^{(K)} \neq \emptyset$ do
    Determine $m$ from (2.3.6).
    $\mathcal{Z}^{(K+1)} = \mathcal{Z}^{(K)} \cup \{m\}$, $\mathcal{F}^{(K+1)} = \mathcal{F}^{(K)} \backslash m$.
    Update $\mathbf{R}$, $\mathbf{c}$, $\gamma$ using (2.3.7)–(2.3.9).
    $K \leftarrow K + 1$.
    Update $\mathcal{U}$ according to (2.3.4) and remove indices added to $\mathcal{U}^{(K)}$ from $\mathcal{F}^{(K)}$.
  Return solution: Compute $\mathbf{b}_{\mathcal{Y}^{(K)}}$ from (2.3.10), $b_n = 0$ for $n \in \mathcal{Z}^{(K)}$.
```

Once the successive thinning algorithm has terminated with a final subset $\mathcal{Z}^{(K)}$ of zero-valued coefficients, it remains to determine the values of the non-zero coefficients $\mathbf{b}_{\mathcal{Y}^{(K)}}$. We choose $\mathbf{b}_{\mathcal{Y}^{(K)}}$ specifically to maximize the margin in the quadratic constraint subject to $b_n = 0$ for $n \in \mathcal{Z}^{(K)}$, i.e., to minimize the left-hand side of (2.2.2). The solution that was given in Section 2.2 is

$$\mathbf{b}_{\mathcal{Y}^{(K)}} = \mathbf{c}_{\mathcal{Y}^{(K)}} + \left(\mathbf{Q}_{\mathcal{Y}^{(K)}\mathcal{Y}^{(K)}}\right)^{-1} \mathbf{Q}_{\mathcal{Y}^{(K)}\mathcal{Z}^{(K)}} \mathbf{c}_{\mathcal{Z}^{(K)}}. \tag{2.3.10}$$

# Chapter 3

# Sparse filter design under a quadratic constraint: Optimal algorithm for the general case

In Sections 2.2 and 2.3, low-complexity algorithms were presented for solving the sparse filter design problem (2.0.1) exactly in some special cases and approximately in the general case of unstructured $\mathbf{Q}$ matrices. In this chapter, we turn our attention to developing a general optimal algorithm for (2.0.1) based on a standard approach to combinatorial optimization known as branch-and-bound. An overview of the branch-and-bound procedure as it applies to (2.0.1) is given in Section 3.1. Additional background on branch-and-bound can be found in [95]. A detailed description of our branch-and-bound algorithm is deferred until the end of this chapter in Section 3.7.

Branch-and-bound is a general strategy applicable to a wide range of combinatorial optimization problems, and as such can be highly computationally intensive. Our emphasis is on reducing the complexity of branch-and-bound in the specific context of problem (2.0.1). Two key factors in reducing complexity are the availability of nearly optimal solutions and the availability and quality of lower bounds on the optimal cost of (2.0.1). We elaborate further on these points in Section 3.1. As mentioned in Section 2.3, nearly optimal solutions can often be provided by the successive thinning algorithm. The development of lower bounds is the subject of this chapter. We note that neither of the branch-and-bound algorithms in [26] and [33] make much use of bounds, while [3] uses a commercial solver that does not exploit the properties of problem (2.0.1). In Section 3.2, we present first some lower bounds that can be computed with minimal effort. To derive stronger bounds, in Section 3.3 we discuss the technique of linear relaxation, while in Section 3.4 we discuss an alternative method, referred to as diagonal relaxation, in which $\mathbf{Q}$ is replaced by a diagonal matrix. Significant attention is given to analyzing the approximation properties of the two relaxations and the quality of the resulting lower bounds. Numerical experiments in Section 3.6 support our analysis and demonstrate that the lower bounds from diagonal relaxations are often substantially tighter than those from linear relaxations. Computational efficiency in solving relaxations is also important for reducing the overall algorithm complexity. Techniques for solving diagonal relaxations efficiently are described in Section 3.5.

## 3.1 Branch-and-bound

In this section, the branch-and-bound procedure is reviewed in the context of problem (2.0.1). For ease of presentation, we reformulate (2.0.1) as a mixed integer optimization problem. For each coefficient $b_n$, we introduce a binary-valued indicator variable $i_n$ with the property that $i_n = 0$ if $b_n = 0$ and $i_n = 1$ otherwise. The sum of the indicator variables is therefore equal to the zero-norm $\|\mathbf{b}\|_0$. Using this fact, problem (2.0.1) can be restated as follows:

$$\begin{aligned} \min_{\mathbf{b},\mathbf{i}} \quad & \sum_{n=1}^{N} i_n \\ \text{s.t.} \quad & (\mathbf{b}-\mathbf{c})^T \mathbf{Q} (\mathbf{b}-\mathbf{c}) \leq \gamma, \\ & |b_n| \leq B_n i_n \quad \forall\, n, \\ & i_n \in \{0,1\} \quad \forall\, n, \end{aligned} \tag{3.1.1}$$

where $B_n$ is a positive constant for each $n$. The second constraint in (3.1.1) ensures that $i_n$ behaves as an indicator variable, specifically by requiring that $b_n = 0$ if $i_n = 0$ and also forcing $i_n$ to zero if $b_n = 0$ because the sum of the $i_n$ is being minimized. When $i_n = 1$, the second constraint becomes a bound on the absolute value of $b_n$. The constants $B_n$ are chosen to be large enough so that these bounds on $|b_n|$ do not further restrict the set of feasible $\mathbf{b}$ from that in (2.0.1). Specific values for $B_n$ will be chosen later in Section 3.3

when we discuss linear relaxation.

The branch-and-bound procedure solves problem (3.1.1) by dividing it successively into subproblems with fewer variables. The first two subproblems are formed by selecting an indicator variable $i_n$ and fixing it to zero in the first subproblem and to one in the second. Each of the two subproblems, if not solved directly, is subdivided into two more subproblems by fixing a second indicator variable to zero or one. This process, referred to as branching, produces a binary tree of subproblems as depicted in Fig. 3-1.

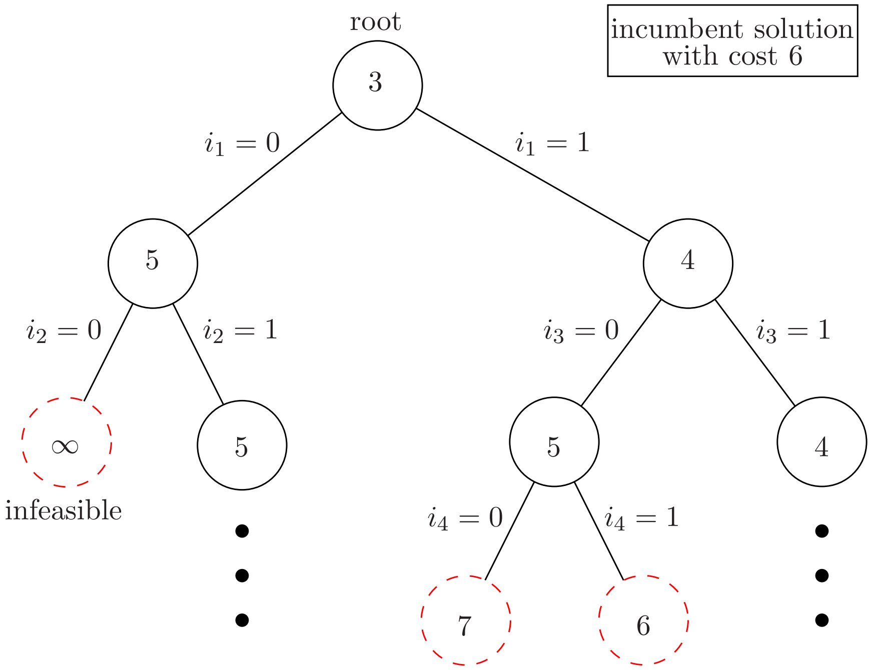


Figure 3-1: Example of a branch-and-bound tree. Each node represents a subproblem and the branch labels indicate the indicator variables that are fixed in going from a parent to a child. The number labelling each node is a lower bound on the optimal cost of the corresponding subproblem. Given an incumbent solution with a cost of 6, the subproblems marked by dashed circles need not be considered any further.

The bounding part of the algorithm consists of computing a lower bound on the optimal cost of each subproblem that is not solved directly. Infeasible subproblems can be regarded as having a lower bound of $+\infty$. Since a child subproblem is related to its parent by the addition of one constraint, the lower bound for the child cannot be less than that for the parent. This non-decreasing property of the lower bounds is illustrated in Fig. 3-1. In addition, feasible solutions may be obtained for certain subproblems. The algorithm keeps a record of the feasible solution with the lowest cost thus far, referred to as the incumbent solution.

To avoid an exhaustive enumeration of all $2^N$ potential subproblems, the following observation is employed: If the lower bound for a subproblem is equal to or higher than the cost of the incumbent solution, then the subproblem cannot lead to better solutions and can thus be eliminated from the tree along with all of its descendants. This operation is referred to as pruning the tree.

Although in worst-case examples the complexity of branch-and-bound remains exponential in $N$ [95], for typical instances the situation can be much improved. The efficiency of a branch-and-bound algorithm depends strongly on the availability and quality of lower bounds for the subproblems. Stronger lower bounds result in more subproblems being pruned. At the same time, the lower bounds should be efficiently computable so as not to increase the overall complexity of the algorithm. Our focus in Sections 3.2–3.4 is on developing lower bounds that can be computed efficiently. In Section 3.2, we discuss bounds that are very inexpensive to compute but have pruning power only for low-dimensional or severely constrained subproblems. Improved lower bounds can be obtained through relaxations of problem (2.0.1). Two types of relaxations are explored in Sections 3.3 and 3.4.

While our presentation in Sections 3.2–3.4 focuses on the root problem (3.1.1), all of the techniques we develop are equally applicable to arbitrary subproblems. This is because each subproblem can be reduced to a lower-dimensional instance of the root problem. In a given subproblem, the indices for which $i_n = 0$ and $i_n = 1$ correspond respectively to coefficients that have been constrained to zero and coefficients that have been designated as being non-zero in terms of cost. These two subsets correspond to the subsets $\mathcal{Z}$ and $\mathcal{U}$ defined in Section 2.3. The remaining indices make up the subset $\mathcal{F}$. Using the results of Appendix A.3, each subproblem can be expressed as in (2.3.1), which is an instance of (2.0.1) or equivalently (3.1.1), with parameters given by (2.3.2) and (A.3.6).

A second important ingredient in a branch-and-bound algorithm is the availability of an initial feasible solution that is nearly optimal. As with lower bounds, an optimal or nearly optimal incumbent solution allows for more subproblems to be eliminated compared to an incumbent solution with a higher cost. A near-optimal initial solution can often be provided by the successive thinning algorithm of Section 2.3. Other heuristic algorithms are also possible.

Two other variable elements in the branch-and-bound procedure are the rule for deciding which indicator variable to fix in a subproblem to generate its children, and the order in which open subproblems are processed. These choices are addressed in Section 3.7 where we provide a detailed description of our algorithm.

## 3.2 Low-complexity lower bounds

This section discusses lower bounds for problem (2.0.1) that require little computational effort to obtain. While the bounds tend to be weak when used in isolation, they become more powerful as part of a branch-and-bound algorithm where they can be applied inexpensively to each new subproblem, improving lower bounds incrementally as the algorithm descends the tree.

For a subproblem specified by index sets $(\mathcal{Z}, \mathcal{U}, \mathcal{F})$ as defined in Section 3.1, the number of elements in $\mathcal{U}$ is clearly a lower bound on the optimal cost of the subproblem. This lower bound may be updated by checking for coefficients in $\mathcal{F}$ that cannot yield feasible solutions when constrained to zero. As discussed in Section 2.3, such coefficients can be identified by evaluating condition (2.3.3) for $n \in \mathcal{F}$ (substituting the parameters for the current subproblem). For coefficients such that a zero value is infeasible, the corresponding indicator variables can be set to 1, thereby decreasing $|\mathcal{F}|$ and increasing $|\mathcal{U}|$. In terms of the branch-and-bound tree, this corresponds to eliminating $i_n = 0$ branches because they lead to infeasible subproblems. The resulting subproblem is of lower dimension and the reduction in dimension can be significant if $\gamma$ is relatively small so that (2.3.3) is violated for many indices $n$.

We will assume henceforth that the above test is performed on every subproblem and indicator variables are set to 1 as appropriate. Thus we need only consider subproblems for which (2.3.3) is satisfied for all $n \in \mathcal{F}$, i.e., a feasible solution exists whenever a single coefficient $b_n$ is constrained to zero.

It is only necessary to check for potential additions to the set $\mathcal{U}$ for subproblems derived from a parent by fixing an indicator variable to zero. Setting an indicator variable to one does not change the set of feasible $\mathbf{b}$, and consequently any coefficient for which a value of zero is feasible in the parent subproblem retains that property in the child subproblem.

It is possible to generalize the test to larger subsets of coefficients that are simultaneously constrained to zero values. The required computation increases dramatically however because the number of subsets grows rapidly as the subset size is increased, and because the matrix inversions in the general feasibility condition (2.2.3) become more complex. Moreover, incorporating information from tests involving larger subsets is less straightforward than simply setting certain $i_n$ to 1.

A second category of tests makes use of (2.1.15) to determine whether there exist feasible solutions with small numbers of non-zero elements. In the extreme case, the solution $\mathbf{b} = \mathbf{0}$ is feasible if $\beta = \gamma - \mathbf{c}^T \mathbf{Q} \mathbf{c} \geq 0$, which corresponds to $\mathcal{Y} = \emptyset$ in (2.1.15). Hence $\beta$ being negative implies a lower bound of at least one ($|\mathcal{U}| + 1$ for a general subproblem) on the minimum zero-norm. For a set $\mathcal{Y}$ consisting of a single index $n$, (2.1.15) becomes

$$-\frac{f_n^2}{Q_{nn}} \leq \beta. \tag{3.2.1}$$

If (3.2.1) is satisfied for some $n \in \mathcal{F}$, $\mathcal{Y} = \{n\}$ is feasible and the minimum zero-norm is 1 ($|\mathcal{U}| + 1$ in general) provided that the solution $\mathbf{b} = \mathbf{0}$ has been excluded. Otherwise, the minimum zero-norm is no less than 2 ($|\mathcal{U}| + 2$). The enumeration can be extended to larger subsets of coefficients, resulting in either an optimal solution or progressively higher lower bounds. However, the increase in computational effort is the same as for (2.2.3).

## 3.3 Linear relaxation

In the previous section, we discussed lower bounds that are simple to compute but relatively weak. Better bounds can be obtained through relaxations of problem (2.0.1).[1] These relaxations are designed to be significantly easier to solve than the original problem. Furthermore, their optimal values are guaranteed to be lower bounds on the original optimal cost. As discussed in Section 3.1, when these lower bounds are close approximations to the true optimal cost, solving relaxations can substantially decrease the complexity of a branch-and-bound algorithm directed at the original problem.

In this section, we apply a common technique known as linear relaxation to (2.0.1). More specifically, in Section 3.3.1 we derive the linear relaxation that results in the highest possible lower bound on the optimal cost of (2.0.1). An alternative type of relaxation is developed in Section 3.4.

[1] Following common usage in the field of optimization, we use the term relaxation to refer to both the technique used to relax certain constraints in a problem as well as the modified problem that results.

In general, given a relaxation of an optimization problem, it is of interest to understand the conditions under which the relaxation is either a good or a poor approximation to the original problem. The quality of approximation is often characterized by the approximation ratio, defined as the ratio between the optimal values of the relaxation and the original problem. In Section 3.3.2, we construct two classes of examples, the first showing that the (strongest possible) linear relaxation can yield an approximation ratio equal to 1 (i.e., the relaxation can be tight), and the second showing that the approximation ratio can be arbitrarily close to zero. We then derive in Section 3.3.3 an absolute bound on the optimal value of the linear relaxation in terms of the number of coefficients $N$. The bound is interpreted as a limitation on the ability of the linear relaxation to approximate many instances of (2.0.1).

### 3.3.1 Derivation of the tightest possible linear relaxation

To apply linear relaxation to (2.0.1), we start with its alternative formulation as a mixed integer optimization problem (3.1.1). A linear relaxation of (3.1.1) is obtained by relaxing the binary constraints on $i_n$, instead allowing $i_n$ to range continuously between 0 and 1. The minimization may then be carried out in two stages. In the first stage, $\mathbf{b}$ is held constant while the objective is minimized with respect to $\mathbf{i}$, resulting in $i_n = |b_n| / B_n$ for each $n$. Substituting back into (3.1.1) gives the following minimization over $\mathbf{b}$:

$$\begin{aligned} \min_{\mathbf{b}} \quad & \sum_{n=1}^{N} \frac{|b_n|}{B_n} \\ \text{s.t.} \quad & (\mathbf{b}-\mathbf{c})^T \mathbf{Q} (\mathbf{b}-\mathbf{c}) \leq \gamma. \end{aligned} \tag{3.3.1}$$

The linear relaxation in (3.3.1) is a convex optimization problem that can be solved efficiently. Since the set of feasible indicator vectors $\mathbf{i}$ is enlarged in deriving (3.3.1) from the original problem (3.1.1), the optimal value of (3.3.1) is a lower bound on that of (3.1.1). More precisely, since the optimal value of (3.1.1) must be an integer, it follows that the ceiling of the optimal value of (3.3.1) is also a lower bound.

To obtain the highest possible lower bound on the original optimal value through linear relaxation, the optimal value of (3.3.1) should be made as large as possible. The form of the objective in (3.3.1) implies that its optimal value is larger for smaller values of $B_n$. At the same time, $B_n$ must be large enough to leave the feasible set unchanged from that in (2.0.1) as discussed in Section 3.1. Specifically, this requires that $B_n \geq |b_n|$ for all $n$ whenever $\mathbf{b}$ satisfies the quadratic constraint (2.1.1). These two competing requirements imply that $B_n$ should be chosen as

$$\begin{aligned} B_n &= \max\left\{|b_n| : (\mathbf{b}-\mathbf{c})^T\mathbf{Q}(\mathbf{b}-\mathbf{c}) \leq \gamma\right\} \\ &= \max\left\{\max\left\{b_n : (\mathbf{b}-\mathbf{c})^T\mathbf{Q}(\mathbf{b}-\mathbf{c}) \leq \gamma\right\},\ \max\left\{-b_n : (\mathbf{b}-\mathbf{c})^T\mathbf{Q}(\mathbf{b}-\mathbf{c}) \leq \gamma\right\}\right\}. \end{aligned} \tag{3.3.2}$$

The inner maximizations in (3.3.2) can be solved in closed form as shown in Appendix B.1, yielding

$$\max\left\{b_n : (\mathbf{b}-\mathbf{c})^T\mathbf{Q}(\mathbf{b}-\mathbf{c}) \leq \gamma\right\} = \sqrt{\gamma\left(\mathbf{Q}^{-1}\right)_{nn}} + c_n, \tag{3.3.3a}$$

$$\max\left\{-b_n : (\mathbf{b}-\mathbf{c})^T\mathbf{Q}(\mathbf{b}-\mathbf{c}) \leq \gamma\right\} = \sqrt{\gamma\left(\mathbf{Q}^{-1}\right)_{nn}} - c_n. \tag{3.3.3b}$$

Hence (3.3.2) can be simplified to

$$B_n = \sqrt{\gamma\left(\mathbf{Q}^{-1}\right)_{nn}} + |c_n|. \tag{3.3.4}$$

The value of $B_n$ can be decreased even further if it can be made to depend on the sign of $b_n$. For example, if $b_n$ is known to be positive, $B_n$ only needs to be greater than or equal to the quantity in (3.3.3a) without regard to (3.3.3b), while the reverse is true if $b_n$ is negative. Unless $c_n = 0$, one of the quantities in (3.3.3a) and (3.3.3b) is smaller than the value in (3.3.4). The key to allowing $B_n$ to depend on the sign of $b_n$ is to separate $b_n$ into its positive and negative parts. We express each $b_n$ as

$$b_n = b_n^+ - b_n^-, \quad b_n^+,\ b_n^- \geq 0. \tag{3.3.5}$$

Under the condition that one of $b_n^+$, $b_n^-$ is always zero, the representation in (3.3.5) is unique, $b_n = b_n^+$ when $b_n > 0$, and $b_n = -b_n^-$ when $b_n < 0$. Hence $b_n^+$ and $b_n^-$ can be interpreted as the positive and negative parts of $b_n$. We assign to each pair $b_n^+$, $b_n^-$ corresponding pairs of binary-valued indicator variables $i_n^+$, $i_n^-$ and positive constants $B_n^+$, $B_n^-$. We then consider the following generalization of (3.1.1):

$$\begin{aligned}\min_{\mathbf{b}^+,\mathbf{b}^-,\mathbf{i}^+,\mathbf{i}^-} \quad & \sum_{n=1}^{N} \left(i_n^+ + i_n^-\right) \\ \text{s.t.} \quad & (\mathbf{b}^+ - \mathbf{b}^- - \mathbf{c})^T \mathbf{Q} (\mathbf{b}^+ - \mathbf{b}^- - \mathbf{c}) \leq \gamma, \\ & 0 \leq b_n^+ \leq B_n^+ i_n^+, \quad 0 \leq b_n^- \leq B_n^- i_n^- \qquad \forall\, n, \\ & i_n^+ \in \{0,1\}, \quad i_n^- \in \{0,1\} \qquad \forall\, n.\end{aligned} \tag{3.3.6}$$

The first constraint is the quadratic constraint (2.1.1) rewritten in terms of $\mathbf{b}^+$ and $\mathbf{b}^-$. The second line of constraints ensures that $i_n^+$ and $i_n^-$ act as indicator variables for $b_n^+$ and $b_n^-$ respectively. Furthermore, the condition that at least one of $b_n^+$, $b_n^-$ is zero for every $n$ is automatically satisfied at an optimal solution to (3.3.6). Otherwise, if $b_n^+$ and $b_n^-$ are both non-zero for some $n$, both could be decreased by $\min\{b_n^+, b_n^-\}$ without affecting the first constraint while driving the smaller of $b_n^+$, $b_n^-$ to zero and allowing one of $i_n^+$, $i_n^-$ to be decreased from one to zero, contradicting optimality. It follows from this property that at least one of $i_n^+$, $i_n^-$ is zero at an optimal solution, with both being zero if $b_n^+ = b_n^- = 0$. Thus the objective function in (3.3.6) behaves exactly like the zero-norm $\|\mathbf{b}\|_0$.

To guarantee that (3.3.6) is a valid reformulation of (2.0.1), the constants $B_n^+$, $B_n^-$ should be large enough to not constrain $\mathbf{b}^+$, $\mathbf{b}^-$ any further than the quadratic constraint already does. This requirement is met by choosing

$$B_n^+ = \sqrt{\gamma\left(\mathbf{Q}^{-1}\right)_{nn}} + c_n, \tag{3.3.7a}$$

$$B_n^- = \sqrt{\gamma\left(\mathbf{Q}^{-1}\right)_{nn}} - c_n, \tag{3.3.7b}$$

since (3.3.3a) and (3.3.3b) represent the largest possible values of $b_n^+$ and $b_n^-$ respectively under the quadratic constraint.

As before with (3.1.1), a linear relaxation of (3.3.6) is obtained by replacing the binary constraints on $i_n^+$ and $i_n^-$ with unit-interval constraints and then minimizing with respect to $\mathbf{i}^+$ and $\mathbf{i}^-$ while holding $\mathbf{b}^+$ and $\mathbf{b}^-$ constant. The resulting linear relaxation is given by

$$\begin{aligned}\min_{\mathbf{b}^+,\mathbf{b}^-} \quad & \sum_{n=1}^{N} \left(\frac{b_n^+}{B_n^+} + \frac{b_n^-}{B_n^-}\right) \\ \text{s.t.} \quad & (\mathbf{b}^+ - \mathbf{b}^- - \mathbf{c})^T \mathbf{Q} (\mathbf{b}^+ - \mathbf{b}^- - \mathbf{c}) \leq \gamma, \qquad \mathbf{b}^+ \geq \mathbf{0}, \quad \mathbf{b}^- \geq \mathbf{0}.\end{aligned} \tag{3.3.8}$$

Using a standard technique based on the representation in (3.3.5) to replace the absolute value functions in the first linear relaxation (3.3.1) with linear functions (see [96]), it can be seen that (3.3.1) is a special case of (3.3.8) with $B_n^+ = B_n^- = B_n$. Since $B_n$ must satisfy (3.3.4) for (3.3.1) to be a valid relaxation of (2.0.1) while $B_n^+$ and $B_n^-$ can be chosen as in (3.3.7), the optimal value of (3.3.8) is at least as large as that of (3.3.1). Hence (3.3.8) is a stronger relaxation than (3.3.1).

An alternative interpretation of the linear relaxation in (3.3.1) is as a weighted $\ell^1$ relaxation of the $\ell^0$ minimization in (2.0.1). In (3.3.8), the weights are also allowed to depend on the signs of the coefficients. The values of $B_n^+$ and $B_n^-$ in (3.3.7) correspond to the choice of weights that gives the tightest relaxation in this class. For this reason, we will use the term linear relaxation to refer henceforth to (3.3.8) with $B_n^+$ and $B_n^-$ given by (3.3.7).

Fig. 3-2 shows a two-dimensional graphical representation of the linear relaxation as an asymetrically-weighted $\ell^1$ minimization. The values of $B_n^\pm$ are given by the maximum extent of the feasible ellipsoid along the positive and negative coordinate directions and can be found graphically as indicated in Fig. 3-2. The asymmetric diamond shape represents a level contour of the $\ell^1$ norm weighted by $1/B_n^\pm$. The solution to the weighted $\ell^1$ minimization can be visualized by inflating the diamond until it just touches the feasible ellipsoid. The point of tangency is the optimal solution and the tangent contour corresponds to the optimal value. In Section 3.3.2, we will draw upon this geometric intuition to construct best-case and worst-case examples in terms of the strength of the bound provided by the linear relaxation.

Lemaréchal and Oustry [97] have shown that a common semidefinite relaxation technique is equivalent to linear relaxation when applied to sparsity maximization problems such as (2.0.1). As a consequence, the properties of the linear relaxation (3.3.8) to be discussed in Sections 3.3.2 and 3.3.3 also apply to this type of semidefinite relaxation.

In Section 3.3.2, we will also require the dual of the linear relaxation (3.3.8), given by

$$\begin{aligned} \max_{\boldsymbol{\mu}} \quad & \mathbf{c}^T\boldsymbol{\mu} - \sqrt{\gamma\boldsymbol{\mu}^T\mathbf{Q}^{-1}\boldsymbol{\mu}} \\ \text{s.t.} \quad & -\mathbf{g}^- \leq \boldsymbol{\mu} \leq \mathbf{g}^+, \end{aligned} \tag{3.3.9}$$

with $g_n^+ = 1/B_n^+$ and $g_n^- = 1/B_n^-$ for all $n$. The derivation of the dual problem can be found in Appendix B.2. Since the primal problem (3.3.8) is convex and has a strictly feasible solution $\mathbf{b}^+ - \mathbf{b}^- = \mathbf{c}$, and the dual has a strictly feasible solution $\boldsymbol{\mu} = \mathbf{0}$, by Slater's

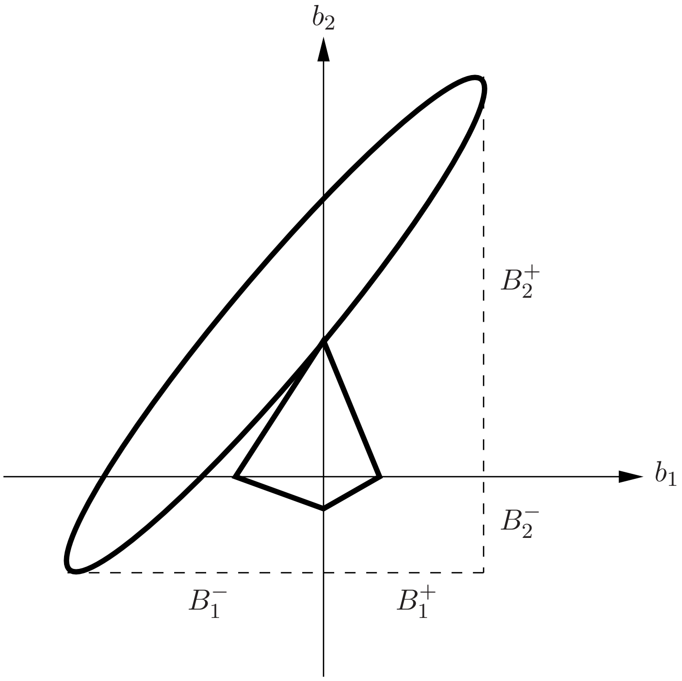


Figure 3-2: Interpretation of the linear relaxation as a weighted $\ell^1$ minimization and a graphical representation of its solution.

condition the optimal values of the primal and dual are equal [98]. The dual is a nonlinear optimization problem with upper and lower bound constraints on each of the variables and is generally easier to solve than the primal because of the nature of the constraints.

While this thesis focuses on linear relaxation as a means of bounding the optimal cost of problem (2.0.1), linear relaxation can also be used to generate a feasible solution. Once the linear relaxation (3.3.8) has been solved, we may define a subset $\mathcal{Z}$ of zero-valued coefficients by setting a threshold between 0 and 1 and including in $\mathcal{Z}$ those indices $n$ for which both $i_n^+ = b_n^+/B_n^+$ and $i_n^- = b_n^-/B_n^-$ fall below the threshold. In effect, the fractional $i_n^\pm$ values returned by the linear relaxation are rounded up to 1 or down to 0 based on the threshold. We then use condition (2.2.3) to determine whether the subset $\mathcal{Z}$ is feasible. To obtain the sparsest solution possible through this method, the threshold can be adjusted until $\mathcal{Z}$ is a minimal feasible subset. This rounding technique has been applied to other integer optimization problems as a way of obtaining a solution with provable approximation guarantees [95]. The present case is complicated by the fact that the minimum threshold required for feasibility is difficult to predict. It may still be possible however to establish an approximation guarantee for a solution produced in this manner.

### 3.3.2 Best-case and worst-case examples

We now investigate the approximation properties of the linear relaxation (3.3.8). In this subsection, we construct two classes of examples, the first of which shows that the approximation ratio can equal 1, the highest possible value, while the second shows that the ratio can be arbitrarily close to zero for large $N$. The examples contribute to an understanding of when the linear relaxation is expected to be a good approximation to problem (2.0.1) and when it is expected to be poor. Together these examples also imply that it is not possible to establish a non-trivial constant bound on the approximation ratio that holds for all instances of the problem.

Throughout this subsection, we set $\mathbf{c} = \mathbf{e}$, a vector of all ones, and $\gamma = 1$, which can be regarded as a normalization. For the best-case examples, we wish to construct instances of (2.0.1), and more specifically ellipsoids parameterized by $\mathbf{Q}$, for which the optimal value of the linear relaxation is large. Based on the intuition of Fig. 3-2, this can be done by choosing all but one of the ellipsoid axes to be short and orienting the remaining major axis so that it is nearly parallel to the level surfaces of the $\ell^1$ norm. This gives the $\ell^1$ diamond more room to grow before intersecting the ellipsoid. In addition, to ensure that (2.3.3) is satisfied, the ellipsoid should contain a point at which $b_n = 0$ for each $n$.

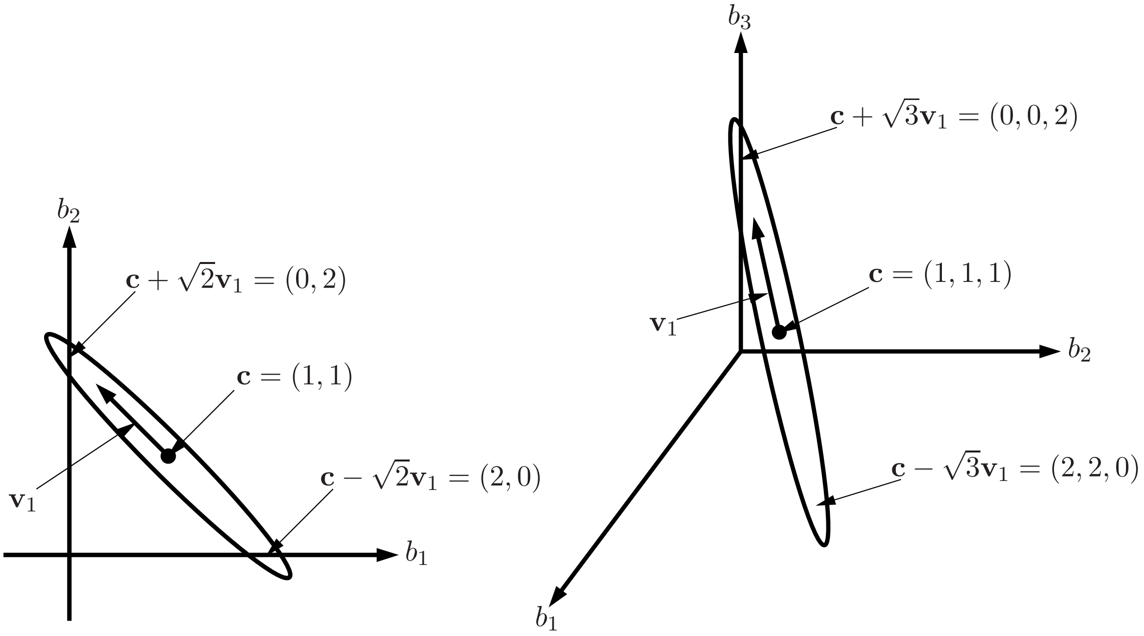
To translate the geometric intuition into an algebraic specification, we assume that $\mathbf{Q}$ has the following eigendecomposition:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{V}_\perp \end{bmatrix} \begin{bmatrix} \lambda_1 & \\ & \lambda_2 \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{v}_1^T \\ \mathbf{V}_\perp^T \end{bmatrix}, \tag{3.3.10}$$

where $\lambda_1 = 1/N$ and $\lambda_2$ is very large relative to $\lambda_1$ so that the minor axes of the ellipsoid are small. For $N$ even, the eigenvector $\mathbf{v}_1$ corresponding to $\lambda_1$ is chosen to have half of its components equal to $+1/\sqrt{N}$ and the other half equal to $-1/\sqrt{N}$. For $N$ odd, $(N+1)/2$ components of $\mathbf{v}_1$ are equal to $+1/\sqrt{N}$ and $(N-1)/2$ components are equal to $-1/\sqrt{N}$. The matrix $\mathbf{V}_\perp$ of eigenvectors corresponding to $\lambda_2$ is chosen so that $\begin{bmatrix} \mathbf{v}_1 & \mathbf{V}_\perp \end{bmatrix}$ is an orthogonal matrix, i.e.,

$$\begin{bmatrix} \mathbf{v}_1 & \mathbf{V}_\perp \end{bmatrix} \begin{bmatrix} \mathbf{v}_1^T \\ \mathbf{V}_\perp^T \end{bmatrix} = \mathbf{v}_1 \mathbf{v}_1^T + \mathbf{V}_\perp \mathbf{V}_\perp^T = \mathbf{I}. \tag{3.3.11}$$

Ellipsoids that correspond to these choices of $\mathbf{Q}$, $\mathbf{c}$, and $\gamma$ are sketched in Fig. 3-3 for the cases $N = 2$ and $N = 3$.



$b_2$
$\mathbf{c} + \sqrt{2}\mathbf{v}_1 = (0, 2)$
$\mathbf{c} = (1, 1)$
$\mathbf{v}_1$
$\mathbf{c} - \sqrt{2}\mathbf{v}_1 = (2, 0)$
$b_1$
$b_3$
$\mathbf{c} + \sqrt{3}\mathbf{v}_1 = (0, 0, 2)$
$\mathbf{c} = (1, 1, 1)$
$\mathbf{v}_1$
$b_2$
$\mathbf{c} - \sqrt{3}\mathbf{v}_1 = (2, 2, 0)$
$b_1$

(a)

(b)

Figure 3-3: Ellipsoids corresponding to the first class of examples for (a) $N = 2$ and (b) $N = 3$.

Given the values of $\lambda_1$ and $\mathbf{v}_1$ above, the points $\mathbf{b} = \mathbf{c} \pm \sqrt{N}\mathbf{v}_1$ are feasible for problem (2.0.1) as can be verified by substitution into (2.1.1). Furthermore, if $N$ is even, both $\mathbf{c} + \sqrt{N}\mathbf{v}_1$ and $\mathbf{c} - \sqrt{N}\mathbf{v}_1$ have $N/2$ zero-valued components. If $N$ is odd, one of these points has $(N-1)/2$ non-zero components and the other has $(N+1)/2$ non-zero components. Assuming that $\lambda_2$ is large enough, i.e., the minor ellipsoid axes are short enough, the points $\mathbf{c} \pm \sqrt{N}\mathbf{v}_1$ have the greatest number of zero components and hence the minimum zero-norm in (2.0.1) is equal to $N/2$ for $N$ even and $(N-1)/2$ for $N$ odd. In Appendix B.3, we show that the linear relaxation leads to a lower bound on (2.0.1) that is also equal to $N/2$ or $(N-1)/2$, thus proving that the approximation ratio can equal 1.

We now present a second class of examples to show that the approximation ratio can be arbitrarily close to zero for large $N$. The approximation ratio cannot be exactly equal to zero since that would require the optimal value of the linear relaxation to be zero, which occurs only if $\mathbf{b} = \mathbf{0}$ is a feasible solution to (2.0.1), i.e., only if the minimum zero-norm in the original problem is also equal to zero. Thus our goal in the following construction is to have the optimal value of the linear relaxation be less than 1, so that the lower bound on (2.0.1) is equal to 1 after taking the ceiling. In addition, given our assumption that the feasible range for each coefficient $b_n$ includes zero, i.e., that (2.3.3) is satisfied for all $n$, the optimal cost in (2.0.1) can be no greater than $N-1$. Accordingly we require in the construction that the optimal cost of (2.0.1) be equal to $N-1$. The difference of $N-2$ between the true optimal cost and the lower bound provided by linear relaxation is essentially the worst possible, and the approximation ratio of $1/(N-1)$ clearly approaches zero as $N$ increases.

To achieve the goals laid out above, we refer again to the geometry of Fig. 3-2. The optimal value of the linear relaxation can be made small by orienting the major axis of the ellipsoid so that it points toward the origin, thus limiting the growth of the $\ell^1$ diamond. At the same time, the minor axes should be chosen large enough for the ellipsoid to intersect the hyperplanes $b_n = 0$ for all $n$, but also small enough to not intersect any of the hyperplanes defined by two components being equal to zero. Assuming again that $\mathbf{Q}$ has the eigendecomposition shown in (3.3.10), these requirements can be met by choosing $\lambda_1 = 1/(N-1)$, $\mathbf{v}_1 = (1/\sqrt{N})\mathbf{e}$, and $\lambda_2 = (N-1)/2$. In Appendix B.4, we verify that this choice of parameters leads to an optimal value of $N-1$ for (2.0.1) and an optimal value less than 1 for its linear relaxation.

### 3.3.3 Absolute upper bound

The examples in the previous subsection demonstrate that there does not exist a non-trivial constant bound on the ratio between the optimal value of the linear relaxation (3.3.8) and the optimal value of the original problem. It is possible however to obtain an absolute upper bound on the optimal value of the linear relaxation in terms of $N$, the total number of coefficients. We use the fact that any feasible solution to the primal form (3.3.8) of the linear relaxation provides an upper bound on the optimal value. Choosing $\mathbf{b}^+ - \mathbf{b}^- = \mathbf{c}$, i.e., $b_n^+ = c_n$, $b_n^- = 0$ for $c_n \geq 0$ and $b_n^+ = 0$, $b_n^- = |c_n|$ for $c_n < 0$, we obtain an upper bound of

$$\sum_{n:c_n>0} \frac{c_n}{B_n^+} + \sum_{n:c_n<0} \frac{|c_n|}{B_n^-} = \sum_{n=1}^{N} \frac{|c_n|}{\sqrt{\gamma\left(\mathbf{Q}^{-1}\right)_{nn}} + |c_n|}, \tag{3.3.12}$$

using (3.3.7). Since we have assumed that (2.3.3) is satisfied for all $n$, each of the fractions on the right-hand side of (3.3.12) is no greater than $1/2$. As a consequence, the optimal value of the linear relaxation can be no larger than $N/2$. This upper bound can be strengthened by scaling the solution $\mathbf{b}^+ - \mathbf{b}^- = \mathbf{c}$, which is in the center of the feasible set, so that it lies on the boundary instead. A scale factor that meets this criterion is

$$\theta = 1 - \sqrt{\frac{\gamma}{\mathbf{c}^T\mathbf{Q}\mathbf{c}}}, \tag{3.3.13}$$

which is less than 1 and also greater than 0 provided that $\mathbf{b}^+ - \mathbf{b}^- = \mathbf{0}$ is not a feasible solution to (3.3.6). Thus the upper bound in (3.3.12) can be reduced by the factor $\theta$.

It is apparent from (3.3.12) that the lower bound resulting from the linear relaxation cannot be tight if the optimal cost in (2.0.1) is greater than $\lceil N/2 \rceil$. We infer that it is unlikely for the linear relaxation to be a good approximation to (2.0.1) in most instances, since if it were, this would imply that the optimal value of (2.0.1) is not much greater than $N/2$ in most cases, a fact that is considered unlikely. The situation is exacerbated if the factor $\theta$ in (3.3.13) is small. This motivates the consideration of a second type of relaxation as we describe next.

## 3.4 Diagonal relaxation

As an alternative to linear relaxations, this section introduces relaxations of problem (2.0.1) in which the matrix $\mathbf{Q}$ is replaced by a diagonal matrix, an approach we refer to as diagonal relaxation. As seen in Section 2.2.1, the solution to the problem of sparse design is straightforward in the diagonal case, thus making it attractive as a relaxation of the problem when $\mathbf{Q}$ is non-diagonal. In Section 3.4.1, we show how to obtain diagonal relaxations of (2.0.1), in particular the tightest possible diagonal relaxation.

As with linear relaxations, we are interested in understanding how well diagonal relaxations can approximate the original problem. It is clear that if $\mathbf{Q}$ is already diagonal, the diagonal relaxation and the original problem coincide and the approximation ratio is equal to 1. In Section 3.4.2, we exhibit worst-case examples in which the approximation ratio is equal to zero. While this implies that diagonal relaxation is no better than linear relaxation in terms of the range of approximation ratios encountered over all possible instances, diagonal relaxation can yield a more favorable approximation for certain classes of instances. Several such examples are illustrated by means of numerical comparisons in Section 3.6. To complement the numerical results, in this section we analyze how the quality of approximation depends on properties of the matrix $\mathbf{Q}$, or equivalently of the ellipsoid $\mathcal{E}_{\mathbf{Q}}$ corresponding to $\mathbf{Q}$. In Section 3.4.4, the approximation quality is characterized based on the condition number of $\mathbf{Q}$, while in Section 3.4.5, the case of diagonally dominant $\mathbf{Q}$ is considered. In Section 3.4.6, we analyze the case in which the axes of the ellipsoid $\mathcal{E}_{\mathbf{Q}}$ are nearly aligned with the coordinate axes, which can be viewed as the geometric counterpart to the diagonally dominant case. To strengthen some of our results, we exploit the invariance of both problem (2.0.1) and its diagonal relaxation to diagonal scaling transformations, properties that are derived in Section 3.4.3.

### 3.4.1 Derivation of the tightest possible diagonal relaxation

To obtain a diagonal relaxation, the quadratic constraint in (2.1.1) is replaced with a similar constraint involving a positive definite diagonal matrix $\mathbf{D}$:

$$(\mathbf{b}-\mathbf{c})^T\mathbf{D}(\mathbf{b}-\mathbf{c}) = \sum_{n=1}^{N} D_{nn}(b_n - c_n)^2 \leq \gamma. \tag{3.4.1}$$

Geometrically, constraint (3.4.1) specifies an ellipsoid, denoted as $\mathcal{E}_{\mathbf{D}}$, with axes that are aligned with the coordinate axes. Since the relaxation is intended to provide a lower bound for the original problem, we require that the coordinate-aligned ellipsoid $\mathcal{E}_{\mathbf{D}}$ enclose the original ellipsoid $\mathcal{E}_{\mathbf{Q}}$ so that minimizing over $\mathcal{E}_{\mathbf{D}}$ yields a lower bound on the minimum over $\mathcal{E}_{\mathbf{Q}}$. Because of symmetry, the two ellipsoids can be made concentric without any loss in the quality of the relaxation. Then the nesting of the ellipsoids is equivalent to $\mathbf{Q}-\mathbf{D}$ being positive semidefinite, which we write as $\mathbf{Q}-\mathbf{D}\succeq \mathbf{0}$ or $\mathbf{Q}\succeq\mathbf{D}$. Sufficiency follows from the inequality

$$(\mathbf{b}-\mathbf{c})^T\mathbf{D}(\mathbf{b}-\mathbf{c}) \leq (\mathbf{b}-\mathbf{c})^T\mathbf{Q}(\mathbf{b}-\mathbf{c}) \quad \forall\,\mathbf{b}, \tag{3.4.2}$$

so if $\mathbf{b}\in\mathcal{E}_{\mathbf{Q}}$, i.e., $\mathbf{b}$ satisfies (2.1.1), then it also satisfies (3.4.1) and belongs to $\mathcal{E}_{\mathbf{D}}$. The condition $\mathbf{Q}\succeq\mathbf{D}$ is necessary because $\mathbf{Q}\not\succeq\mathbf{D}$ implies the existence of a vector $\mathbf{b}$ that violates the inequality in (3.4.2), and by scaling $\mathbf{b}-\mathbf{c}$ so that the right side of (3.4.2) is equal to $\gamma$, we see that $\mathbf{b}$ satisfies (2.1.1) but does not satisfy (3.4.1).

For every $\mathbf{D}$ satisfying $\mathbf{0}\preceq\mathbf{D}\preceq\mathbf{Q}$, minimizing $\|\mathbf{b}\|_0$ subject to (3.4.1) results in a lower bound for problem (2.0.1). Thus the set of diagonal relaxations is parameterized by $\mathbf{D}$, as shown graphically in Fig. 3-4. As with linear relaxations in Section 3.3.1, we are interested in finding a diagonal relaxation that is as tight as possible, i.e., a matrix $\mathbf{D}_d$ such that the minimum zero-norm associated with $\mathbf{D}_d$ is maximal among all valid choices of $\mathbf{D}$. To see

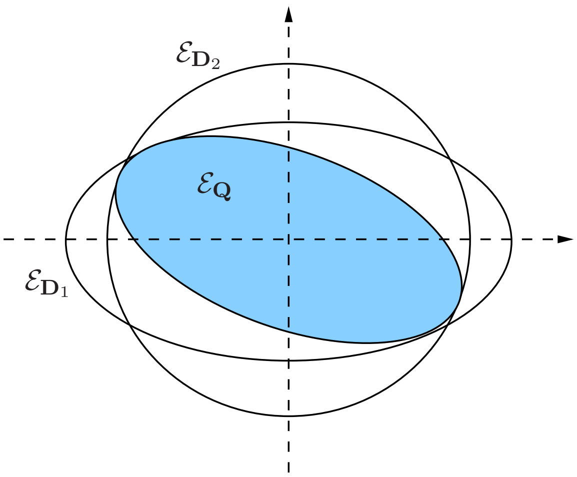

$\mathcal{E}_{\mathbf{D}_2}$
$\mathcal{E}_{\mathbf{Q}}$
$\mathcal{E}_{\mathbf{D}_1}$

Figure 3-4: Two different diagonal relaxations.

how such a relaxation may be obtained, recall from Section 2.2.1 that constraint (3.4.1) admits a feasible solution with $K$ zero-valued elements if and only if the condition in (2.2.7) is met (with $D_{nn}$ in place of $Q_{nn}$). Based on (2.2.7), the tightest diagonal relaxation may be determined by solving the following optimization problem starting from $K = 0$:

$$\begin{aligned} \max_{\mathbf{D}} \quad & \Sigma_K\big(\{D_{nn}c_n^2\}\big) \\ \text{s.t.} \quad & \mathbf{0} \preceq \mathbf{D} \preceq \mathbf{Q}, \\ & \mathbf{D} \text{ diagonal.} \end{aligned} \tag{3.4.3}$$

Denote by $E_d(K)$ the optimal value of (3.4.3). If $E_d(K)$ is less than or equal to $\gamma$, then condition (2.2.7) holds for every $\mathbf{D}$ satisfying the constraints in (3.4.3), and consequently a feasible solution $\mathbf{b}$ with $K$ zero-valued coefficients exists for every such $\mathbf{D}$. We conclude that no diagonal relaxation can give a minimum zero-norm greater than $N - K$. The value of $K$ is then incremented by 1 and (3.4.3) is re-solved. If on the other hand $E_d(K)$ is greater than $\gamma$ for some $K = K_d + 1$, then according to (2.2.7) there exists a $\mathbf{D}_d$ for which it is not feasible to have a solution with $K_d + 1$ zero elements. When combined with the conclusions drawn for $K \leq K_d$, this implies that the minimum zero-norm with $\mathbf{D} = \mathbf{D}_d$ is equal to $N - K_d$. It follows that $N - K_d$ is the tightest lower bound achievable with a diagonal relaxation.

The foregoing procedure determines both the tightest possible diagonal relaxation and its optimal value at the same time. For convenience, we will refer to the overall procedure as solving the diagonal relaxation. The term diagonal relaxation will refer henceforth to the tightest diagonal relaxation.

The main computational burden in solving the diagonal relaxation lies in solving (3.4.3) for multiple values of $K$. Problem (3.4.3) can be recast as a semidefinite optimization problem to which efficient interior-point algorithms as well as other simplifications may be applied. A detailed discussion of the solution of (3.4.3) can be found in Section 3.5.

As with the linear relaxation, our main interest in the diagonal relaxation is as a method of bounding the optimal cost of the original problem (2.0.1). However, the solution of the diagonal relaxation also suggests a heuristic for obtaining a feasible solution to (2.0.1). The procedure described above terminates with a matrix $\mathbf{D}^*$ such that the sum of the $K^*$ smallest $D^*_{nn}c_n^2$ is no greater than $\gamma$. This implies that the index set $\mathcal{Z}$ corresponding to the $K^*$ smallest $D^*_{nn}c_n^2$ is feasible for the relaxed problem. Using condition (2.2.3), we can check whether $\mathcal{Z}$ is also feasible for the original problem (2.0.1). If it is, (2.0.1) is solved because the zero-norm $N - K^*$ of this solution is equal to the lower bound provided by the diagonal relaxation. If not, $\mathcal{Z}$ is reduced in size to correspond to the $K^* - 1$ smallest $D^*_{nn}c_n^2$ and the feasibility test is repeated. The size of $\mathcal{Z}$ is successively decreased in this manner until $\mathcal{Z}$ becomes feasible, yielding a solution with zero-norm equal to $N - |\mathcal{Z}|$. If $|\mathcal{Z}|$ is only slightly smaller than $K^*$, then this solution is close to optimal.

### 3.4.2 Worst-case examples

In the remainder of this section, the approximation properties of the diagonal relaxation are explored. In this subsection, we show that the approximation ratio associated with diagonal relaxation can equal zero. Intuitively, the diagonal relaxation is expected to result in a poor approximation when the original ellipsoid $\mathcal{E}_{\mathbf{Q}}$ is far from being coordinate-aligned. This occurs for example if $\mathcal{E}_{\mathbf{Q}}$ is dominated by a single long axis that has equal components in all coordinate directions, thus forcing the enclosing coordinate-aligned ellipsoid $\mathcal{E}_{\mathbf{D}}$ to be much larger than $\mathcal{E}_{\mathbf{Q}}$. To show that the approximation ratio can actually equal zero in these instances, we use the examples in Section 3.3.2 in which $\mathbf{c} = \mathbf{e}$, $\gamma = 1$, and the eigenvector $\mathbf{v}_1$ has equal-magnitude components. The objective function in (3.4.3) reduces in this case to the sum of the $K$ smallest diagonal entries of $\mathbf{D}$, and the maximum value $E_d(K)$ is compared to 1 to determine whether there exists a feasible solution to the diagonal relaxation with $K$ zero-valued components. We make use of the following lemma, which holds for the case $\mathbf{c} = \mathbf{e}$:

**Lemma 1.** *If $\mathbf{c} = \mathbf{e}$, the optimal value $E_d(K)$ of (3.4.3) is bounded from below by $K\lambda_{\min}(\mathbf{Q})$, where $\lambda_{\min}(\mathbf{Q})$ is the smallest eigenvalue of $\mathbf{Q}$. This lower bound is tight if the eigenvector $\mathbf{v}_1$ corresponding to $\lambda_{\min}(\mathbf{Q})$ has components of equal magnitude.*

*Proof.* The matrix $\mathbf{D} = \lambda_{\min}(\mathbf{Q})\mathbf{I}$ satisfies $\mathbf{D} \preceq \mathbf{Q}$ and is therefore a feasible solution to (3.4.3). Hence the corresponding objective value $K\lambda_{\min}(\mathbf{Q})$ is a lower bound on $E_d(K)$. To show that this lower bound can be tight, first note that for any $\mathbf{D}$ satisfying $\mathbf{D} \preceq \mathbf{Q}$,

$$\mathbf{v}_1^T \mathbf{D} \mathbf{v}_1 = \sum_{n=0}^{N-1} D_{nn} \left(\mathbf{v}_1\right)_n^2 \leq \mathbf{v}_1^T \mathbf{Q} \mathbf{v}_1 = \lambda_{\min}(\mathbf{Q}) \left\|\mathbf{v}_1\right\|_2^2 . \tag{3.4.4}$$

If $\mathbf{v}_1$ has equal-magnitude components, e.g. $\left|\left(\mathbf{v}_1\right)_n\right| = 1/\sqrt{N}$ for all $n$ assuming that $\mathbf{v}_1$ is normalized to have unit 2-norm, (3.4.4) reduces to

$$\sum_{n=0}^{N-1} D_{nn} \leq N\lambda_{\min}(\mathbf{Q}). \tag{3.4.5}$$

Since (3.4.5) holds for any $\mathbf{D}$ such that $\mathbf{D} \preceq \mathbf{Q}$ and is met with equality for $\mathbf{D} = \lambda_{\min}(\mathbf{Q})\mathbf{I}$, $\lambda_{\min}(\mathbf{Q})\mathbf{I}$ is an optimal solution to (3.4.3) when $\mathbf{c} = \mathbf{e}$, $K = N$, and $\mathbf{v}_1$ has equal-magnitude components. This proves that the lower bound of $K\lambda_{\min}(\mathbf{Q})$ is tight in the case $K = N$. Using the fact that the average of the $K$ smallest $D_{nn}$ for $K < N$ is no greater than the average of all $N$ diagonal entries, it follows from (3.4.5) that

$$\Sigma_K\left(\{D_{nn}\}\right) \leq K\lambda_{\min}(\mathbf{Q}), \quad K = 1, 2, \ldots, N-1, \tag{3.4.6}$$

for all $\mathbf{D}$ such that $\mathbf{D} \preceq \mathbf{Q}$. Since the solution $\mathbf{D} = \lambda_{\min}(\mathbf{Q})\mathbf{I}$ also satisfies (3.4.6) with equality, $\lambda_{\min}(\mathbf{Q})\mathbf{I}$ is an optimal solution to (3.4.3) for all $K$ under the assumptions of the lemma. Hence $K\lambda_{\min}(\mathbf{Q})$ is a tight lower bound on $E_d(K)$. □

In the first class of examples in Section 3.3.2, $\lambda_{\min}(\mathbf{Q}) = \lambda_1 = 1/N$ and the corresponding eigenvector $\mathbf{v}_1$ satisfies the property of having equal-magnitude components. It follows from Lemma 1 that $E_d(K)$ is given by $K\lambda_1 = K/N$ for all values of $K$. $E_d(K)$ does not exceed $\gamma = 1$ for any $K$, and hence the solution $\mathbf{b} = \mathbf{0}$ is feasible for the diagonal relaxation. Since the minimum zero-norm in the unrelaxed problem (2.0.1) is either $N/2$

or $(N-1)/2$ depending on whether $N$ is even or odd, the approximation ratio is zero. In addition, given that the linear relaxation results in an approximation ratio of 1 for these examples, we observe that there is no strict dominance relationship between the linear and diagonal relaxations (diagonal relaxations are clearly dominant in the case of diagonal $\mathbf{Q}$).

In the second class of examples in Section 3.3.2, $\lambda_{\min}(\mathbf{Q}) = \lambda_1 = 1/(N-1)$ assuming that $N \geq 3$, and the eigenvector $\mathbf{v}_1$ again has equal-magnitude components. $E_d(K)$ is equal to $K/(N-1)$ and does not exceed $\gamma = 1$ for any $K < N$ but does for $K = N$. Therefore the minimum zero-norm under the diagonal relaxation is equal to 1 and the resulting lower bound on the optimal cost in (2.0.1) is the same as that given by linear relaxation. As mentioned in Section 3.3.2, the difference of $N-2$ between the optimal cost of (2.0.1) and the lower bound provided by both relaxations is essentially the largest possible assuming that (2.3.3) is satisfied.

The examples in this subsection demonstrate that it is not possible to place a non-trivial constant bound on the approximation ratio associated with diagonal relaxation that holds for all instances. However, for $\mathbf{Q}$ matrices with certain properties, the diagonal relaxation tends to be a good approximation to the original problem. We analyze some of these cases in detail in Sections 3.4.4–3.4.6. Numerical evidence of the approximation quality of diagonal relaxations is presented in Section 3.6.

### 3.4.3 Invariance under diagonal scaling

Before proceeding to the main analytical results in Sections 3.4.4–3.4.6, we first show that both problem (2.0.1) and its diagonal relaxation are invariant to diagonal scalings of the ellipsoid $\mathcal{E}_{\mathbf{Q}}$. More precisely, we show that the optimal value of (2.0.1) and the optimal value $E_d(K)$ of (3.4.3) are invariant. This invariance property will be used to strengthen certain bounds in Sections 3.4.4–3.4.6.

To show that the optimal value of (2.0.1) is invariant under diagonal scaling of the feasible set $\mathcal{E}_{\mathbf{Q}}$, we let $\mathbf{S}$ be an arbitrary invertible diagonal matrix. The image of $\mathcal{E}_{\mathbf{Q}}$ under $\mathbf{S}$ is given by

$$\{\mathbf{y} = \mathbf{S}\mathbf{b} : (\mathbf{b}-\mathbf{c})^T\mathbf{Q}(\mathbf{b}-\mathbf{c}) \leq \gamma\} = \{\mathbf{y} : (\mathbf{y}-\mathbf{S}\mathbf{c})^T\mathbf{S}^{-1}\mathbf{Q}\mathbf{S}^{-1}(\mathbf{y}-\mathbf{S}\mathbf{c}) \leq \gamma\},$$

i.e., an ellipsoid with center $\mathbf{S}\mathbf{c}$ and shape matrix $\mathbf{S}^{-1}\mathbf{Q}\mathbf{S}^{-1}$. The minimization of the zero-norm over $\mathbf{S}(\mathcal{E}_{\mathbf{Q}})$ reads

$$\min_{\mathbf{y}} \quad \|\mathbf{y}\|_0 \qquad \text{s.t.} \qquad (\mathbf{y} - \mathbf{Sc})^T \mathbf{S}^{-1}\mathbf{Q}\mathbf{S}^{-1}(\mathbf{y} - \mathbf{Sc}) \leq \gamma. \tag{3.4.7}$$

By substituting $\mathbf{y} = \mathbf{Sb}$, simplifying the constraint, and recognizing that $\|\mathbf{Sb}\|_0 = \|\mathbf{b}\|_0$ because $\mathbf{S}$ is diagonal, we recover problem (2.0.1). This establishes that (2.0.1) and its scaled counterpart (3.4.7) have the same optimal value.

Next we consider the diagonal relaxation of the scaled problem (3.4.7). The tightest possible diagonal relaxation can be found by solving

$$\begin{aligned} \max_{\mathbf{D}} \quad & \Sigma_K\big(\{D_{nn}(S_{nn}c_n)^2\}\big) \\ \text{s.t.} \quad & \mathbf{0} \preceq \mathbf{D} \preceq \mathbf{S}^{-1}\mathbf{Q}\mathbf{S}^{-1}, \\ & \mathbf{D} \text{ diagonal}, \end{aligned} \tag{3.4.8}$$

which is equivalent to

$$\begin{aligned} \max_{\mathbf{D}} \quad & \Sigma_K\big(\{(S_{nn}D_{nn}S_{nn})c_n^2\}\big) \\ \text{s.t.} \quad & \mathbf{0} \preceq \mathbf{SDS} \preceq \mathbf{Q}, \\ & \mathbf{D} \text{ diagonal}, \end{aligned}$$

since $\mathbf{S}$ is invertible. By absorbing the diagonal scaling by $\mathbf{S}$ into the matrix $\mathbf{D}$, we recover problem (3.4.3). This shows that the optimal value $E_d(K)$ of (3.4.3) is invariant under diagonal scaling. As a consequence, the value of $K_d$, the largest $K$ such that $E_d(K) \leq \gamma$, is also invariant.

Recall from Section 2.2.4 that diagonal scalings are essentially the only invertible linear transformations that preserve globally the ordering of vectors by sparsity. We infer from the invariance of $E_d(K)$ that it is not possible to obtain a more favorable diagonal relaxation of problem (2.0.1) by first applying a linear transformation in this class to the feasible set. However, some of the bounds that we derive in Sections 3.4.4–3.4.6 do change under diagonal scaling and can therefore be strengthened by an appropriate choice of scaling.

### 3.4.4 Condition number bound on the approximation ratio

In this subsection, the quality of approximation of the diagonal relaxation is characterized in terms of the condition number of the matrix $\mathbf{Q}$. Geometrically, the condition number $\kappa(\mathbf{Q})$ corresponds to the ratio between the longest and shortest axes of the ellipsoid $\mathcal{E}_{\mathbf{Q}}$. We expect intuitively that the diagonal relaxation will yield a good approximation when the condition number is low. A small value for $\kappa(\mathbf{Q})$ implies that the ellipsoid $\mathcal{E}_{\mathbf{Q}}$ is nearly spherical and can therefore be enclosed by a coordinate-aligned ellipsoid $\mathcal{E}_{\mathbf{D}}$ of comparable size. This is illustrated in Fig. 3-5 in the two-dimensional case. Given that $\mathcal{E}_{\mathbf{Q}}$ can be approximated well by $\mathcal{E}_{\mathbf{D}}$ in terms of volume, one would expect a close approximation in terms of sparsity as well. Our purpose in this subsection is to formalize the geometric intuition by deriving a bound on the approximation ratio based on the conditioning of $\mathbf{Q}$.

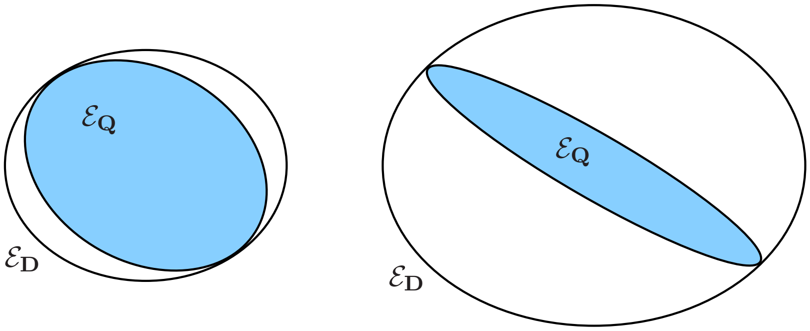

$\mathcal{E}_{\mathbf{Q}}$
$\mathcal{E}_{\mathbf{D}}$
$\mathcal{E}_{\mathbf{Q}}$
$\mathcal{E}_{\mathbf{D}}$

Figure 3-5: Diagonal relaxations for two ellipsoids with contrasting condition numbers.

Our first step is to bound the optimal value of (2.0.1) using the eigenvalues of $\mathbf{Q}$. Toward this end we consider condition (2.2.5), which determines whether there exists a solution to (2.0.1) with $K$ zero-valued coefficients. Define $E_0(K)$ to be the left-hand side of (2.2.5) and denote by $K^*$ the maximum value of $K$ for which (2.2.5) is satisfied, i.e., the maximum number of zero-valued coefficients that is feasible for problem (2.0.1). By bounding $E_0(K)$ in terms of the smallest and largest eigenvalues of $\mathbf{Q}$, $\lambda_{\min}(\mathbf{Q})$ and $\lambda_{\max}(\mathbf{Q})$, we derive the following bounds on $K^*$:

**Lemma 2.** *The maximum number of zero-valued coefficients in problem (2.0.1), $K^*$, is bounded from above by*

$$\overline{K} = \max\{K : \lambda_{\min}(\mathbf{Q})\Sigma_K\big(\{c_n^2\}\big) \leq \gamma\} \tag{3.4.9}$$

*and from below by*

$$\underline{K} = \max\{K : \lambda_{\max}(\mathbf{Q})\Sigma_K\left(\{c_n^2\}\right) \leq \gamma\}. \tag{3.4.10}$$

*Furthermore,*

$$\frac{\overline{K}}{\underline{K}} \leq \frac{\lceil(\underline{K}+1)\kappa(\mathbf{Q})\rceil - 1}{\underline{K}} \approx \kappa(\mathbf{Q}).$$

The proof of Lemma 2 is provided in Appendix B.5.

Next we relate $K_d$, the maximum number of zero-valued coefficients in the diagonal relaxation of (2.0.1), to the bounds in Lemma 2. Specifically, we show that $K_d$ is a tighter upper bound on $K^*$ than $\overline{K}$ in (3.4.9). To prove that $K_d \leq \overline{K}$, we recall that $K_d$ is the largest value of $K$ for which $E_d(K) \leq \gamma$, where $E_d(K)$ is the optimal value of (3.4.3). An upper bound on $K_d$ can be obtained through a lower bound on $E_d(K)$. $E_d(K)$ can be bounded from below by observing that $\mathbf{D} = \lambda_{\min}(\mathbf{Q})\mathbf{I}$ is a feasible solution to (3.4.3), and hence $E_d(K)$ is no smaller than the corresponding objective value $\lambda_{\min}(\mathbf{Q})\Sigma_K\left(\{c_n^2\}\right)$. As a consequence, the largest value of $K$ for which $\lambda_{\min}(\mathbf{Q})\Sigma_K\left(\{c_n^2\}\right) \leq \gamma$ is an upper bound on $K_d$, i.e.,

$$K_d = \max\{K : E_d(K) \leq \gamma\} \leq \max\{K : \lambda_{\min}(\mathbf{Q})\Sigma_K\left(\{c_n^2\}\right) \leq \gamma\} = \overline{K},$$

which is the desired result.

Combining the preceding inequality with Lemma 2 allows us to bound the ratio between $K_d$ and $K^*$ by the condition number $\kappa(\mathbf{Q})$.

**Theorem 2.** *The maximum number $K_d$ of zero-valued coefficients in the diagonal relaxation of problem (2.0.1) satisfies $K^* \leq K_d \leq \overline{K}$ with $\overline{K}$ as given in (3.4.9). Furthermore,*

$$\frac{K_d}{K^*} \leq \frac{\overline{K}}{\underline{K}} \leq \frac{\lceil(\underline{K}+1)\kappa(\mathbf{Q})\rceil - 1}{\underline{K}} \approx \kappa(\mathbf{Q}).$$

Both Lemma 2 and Theorem 2 can be strengthened slightly by exploiting the invariance of the optimal values of (2.0.1) and its diagonal relaxation to diagonal scalings. We consider the scaled problem (3.4.7) in which the feasible ellipsoid $\mathcal{E}_{\mathbf{Q}}$ is scaled by a diagonal matrix $\mathbf{S}$, resulting in $\mathbf{Q}$ being replaced by $\mathbf{S}^{-1}\mathbf{Q}\mathbf{S}^{-1}$ and $\mathbf{c}$ by $\mathbf{S}\mathbf{c}$. Following the same development that led to Theorem 2, we see that the definitions of $\overline{K}$ in (3.4.9) and $\underline{K}$ in (3.4.10) change, as does the condition number. However, as shown in Section 3.4.3, the values of $K^*$ and

$K_d$ do not change. Theorem 2 can therefore be generalized as follows:

**Corollary 1.** *For any invertible diagonal matrix* $\mathbf{S}$*, define*

$$\overline{K}(\mathbf{S}) = \max\{K : \lambda_{\min}(\mathbf{S}^{-1}\mathbf{Q}\mathbf{S}^{-1})\Sigma_K\big(\{S_{nn}c_n^2\}\big) \leq \gamma\},$$
$$\underline{K}(\mathbf{S}) = \max\{K : \lambda_{\max}(\mathbf{S}^{-1}\mathbf{Q}\mathbf{S}^{-1})\Sigma_K\big(\{S_{nn}c_n^2\}\big) \leq \gamma\}.$$

*Then we have the ordering* $\underline{K}(\mathbf{S}) \leq K^* \leq K_d \leq \overline{K}(\mathbf{S})$ *with*

$$\frac{K_d}{K^*} \leq \frac{\overline{K}(\mathbf{S})}{\underline{K}(\mathbf{S})} \leq \frac{\left\lceil(\underline{K}(\mathbf{S})+1)\kappa(\mathbf{S}^{-1}\mathbf{Q}\mathbf{S}^{-1})\right\rceil - 1}{\underline{K}(\mathbf{S})} \approx \kappa(\mathbf{S}^{-1}\mathbf{Q}\mathbf{S}^{-1}).$$

In particular, $\mathbf{S}$ can be chosen to minimize $\kappa(\mathbf{S}^{-1}\mathbf{Q}\mathbf{S}^{-1})$, i.e., as an optimal diagonal preconditioner for $\mathbf{Q}$, to obtain a tighter bound on $K_d/K^*$.

Theorem 2 and Corollary 1 provide a theoretical explanation for the dependence of the approximation ratio for diagonal relaxation on the condition number. This dependence is observed in the numerical experiments of Section 3.6. However, the results in the present subsection do not explain the additional dependence on the distribution of eigenvalues that is also observed in Section 3.6. Specifically, distributions in which most of the eigenvalues of $\mathbf{Q}$ are small and comparable are favored. The dependence on eigenvalue distribution can be explained by the following geometric intuition: Assuming that $\mathcal{E}_{\mathbf{Q}}$ is not close to spherical, i.e., $\kappa(\mathbf{Q})$ is relatively large, it is preferable for most of the ellipsoid axes to be long rather than short, and for the long axes to be comparable in length. Such an ellipsoid tends to require a comparatively smaller coordinate-aligned enclosing ellipsoid, and consequently the diagonal relaxation tends to be a better approximation. For example, in three dimensions, a severely oblate spheroid can on average be enclosed in a smaller coordinate-aligned ellipsoid than an equally severely prolate spheroid. Recalling that the eigenvalues of $\mathbf{Q}$ are inversely proportional to the axis lengths of the ellipsoid $\mathcal{E}_{\mathbf{Q}}$, this argument based on volume explains the preference for certain eigenvalue distributions seen in Section 3.6.

In contrast, the diagonal relaxation tends not to perform well in the case of many large eigenvalues and few small eigenvalues. The examples used in Sections 3.3.2 and 3.4.2 represent extreme cases in this latter category since in both examples there are $N-1$ equally large eigenvalues and a single small eigenvalue. The shape of the feasible ellipsoid is largely determined by the eigenvector $\mathbf{v}_1$ associated with the small eigenvalue, and since in both cases $\mathbf{v}_1$ was chosen to have equal-magnitude coordinates, a relatively large coordinate-aligned ellipsoid is needed to enclose the original ellipsoid.

In the absence of a rigorous explanation for the observed dependence on the eigenvalue distribution, we give instead an informal argument based on the results of this subsection. We focus specifically on the bound $\underline{K}$ given in (3.4.10), which tends to be conservative. The value of $\underline{K}$ depends in turn on the upper bound on $E_0(K)$ given in (B.5.2). One of the inequalities used in the derivation is

$$\lambda_{\max}\left(\left(\left(\mathbf{Q}^{-1}\right)_{\mathcal{Z}\mathcal{Z}}\right)^{-1}\right) \|\mathbf{c}_{\mathcal{Z}}\|_2^2 \leq \lambda_{\max}(\mathbf{Q}) \|\mathbf{c}_{\mathcal{Z}}\|_2^2 , \tag{3.4.11}$$

which holds for all subsets $\mathcal{Z}$. However, to obtain an upper bound on $E_0(K)$, it suffices to bound the left-hand side of (3.4.11) only for the subset $\mathcal{Z}$ that minimizes the left-hand side among all subsets of size $K$. For $|\mathcal{Z}| = K$, the range of possible values for the left-hand side of (3.4.11) is given by (see [92])

$$\lambda_K(\mathbf{Q})\Sigma_K\left(\{c_n^2\}\right) \leq \lambda_{\max}\left(\left(\left(\mathbf{Q}^{-1}\right)_{\mathcal{Z}\mathcal{Z}}\right)^{-1}\right) \|\mathbf{c}_{\mathcal{Z}}\|_2^2 \leq \lambda_{\max}(\mathbf{Q})\Sigma^K\left(\{c_n^2\}\right),$$

where $\lambda_K(\mathbf{Q})$ denotes the $K$th smallest eigenvalue of $\mathbf{Q}$ and $\Sigma^K$ denotes the sum of the $K$ largest elements of a sequence. Thus the upper bound of $\lambda_{\max}(\mathbf{Q})\Sigma_K\left(\{c_n^2\}\right)$ used in (B.5.2) can be much larger than the minimum of the left-hand side of (3.4.11) over all $\mathcal{Z}$ of size $K$, especially if $\lambda_K(\mathbf{Q}) \ll \lambda_{\max}(\mathbf{Q})$. If we assume instead that

$$\min_{|\mathcal{Z}|=K} \lambda_{\max}\left(\left(\left(\mathbf{Q}^{-1}\right)_{\mathcal{Z}\mathcal{Z}}\right)^{-1}\right) \|\mathbf{c}_{\mathcal{Z}}\|_2^2 = c\lambda_K(\mathbf{Q})\Sigma_K\left(\{c_n^2\}\right)$$

where $c$ is a constant not much larger than 1, then (3.4.10) is replaced with

$$\underline{K} = \max\{K : c\lambda_K(\mathbf{Q})\Sigma_K\left(\{c_n^2\}\right) \leq \gamma\}.$$

Following the same reasoning leading to Theorem 2, one arrives at an approximate bound of $c\lambda_{\underline{K}+1}(\mathbf{Q})/\lambda_{\min}(\mathbf{Q})$ in place of $\kappa(\mathbf{Q})$ in Theorem 2, and similarly for Corollary 1. The ratio $\lambda_{\underline{K}+1}(\mathbf{Q})/\lambda_{\min}(\mathbf{Q})$ can be viewed as a partial condition number involving only the $\underline{K}+1$ smallest eigenvalues of $\mathbf{Q}$. Thus if most of the eigenvalues of $\mathbf{Q}$ are small and comparable, the partial condition number $\lambda_{\underline{K}+1}(\mathbf{Q})/\lambda_{\min}(\mathbf{Q})$ is small whereas the full condition number may be much larger.

### 3.4.5 The diagonally dominant case

We now consider the case in which the matrix $\mathbf{Q}$ is diagonally dominant. The notion of diagonal dominance used here will be made precise shortly. It is expected in this case that the original problem can be well-approximated by its diagonal relaxation. Our goal in this subsection is to determine analytically how the quality of approximation depends on the chosen measure of diagonal dominance.

As in Section 3.4.4, our strategy is to bound the ratio $K_d/K^*$ by determining an upper bound on $K_d$ and a lower bound on $K^*$. This can be done by obtaining a lower bound on $E_d(K)$ and an upper bound on $E_0(K)$ respectively. Since $E_d(K)$ is defined as the maximum value of (3.4.3), any feasible solution to (3.4.3) provides a lower bound on $E_d(K)$. We wish to choose a feasible solution that is likely to approximate $E_d(K)$ well given that $\mathbf{Q}$ is diagonally dominant. To do this, we use the following lemma, which determines the optimal value of (3.4.3) under the additional restriction that $\mathbf{D}$ is a multiple of a fixed diagonal matrix.

**Lemma 3.** *Fix a positive definite diagonal matrix $\mathbf{D}_0$, and let $\mathbf{D} = \alpha\mathbf{D}_0$. Then the optimal value of* (3.4.3)*, $E_d(K)$, satisfies*

$$E_d(K) \geq \lambda_{\min}\left(\mathbf{D}_0^{-1/2}\mathbf{Q}\mathbf{D}_0^{-1/2}\right) \cdot \Sigma_K\left(\{(\mathbf{D}_0)_{nn}c_n^2\}\right).$$

*Proof.* With $\mathbf{D} = \alpha\mathbf{D}_0$, (3.4.3) reduces to

$$\begin{aligned} \max_{\alpha} \quad & \alpha\Sigma_K\left(\{(\mathbf{D}_0)_{nn}c_n^2\}\right) \\ \text{s.t.} \quad & \mathbf{0} \preceq \alpha\mathbf{D}_0 \preceq \mathbf{Q}. \end{aligned} \tag{3.4.12}$$

Since $\mathbf{D}$ is restricted to be a multiple of $\mathbf{D}_0$ in (3.4.12), the optimal value of (3.4.12) is a lower bound on $E_d(K)$. Noting that $\Sigma_K\left(\{(\mathbf{D}_0)_{nn}c_n^2\}\right)$ is not a function of $\alpha$ and that $\mathbf{D}_0$ is invertible, (3.4.12) has the same optimal solution as

$$\begin{aligned} \max_{\alpha} \quad & \alpha \\ \text{s.t.} \quad & \mathbf{0} \preceq \alpha\mathbf{I} \preceq \mathbf{D}_0^{-1/2}\mathbf{Q}\mathbf{D}_0^{-1/2}. \end{aligned} \tag{3.4.13}$$

The solution to (3.4.13) is to set $\alpha$ equal to the smallest eigenvalue of $\mathbf{D}_0^{-1/2}\mathbf{Q}\mathbf{D}_0^{-1/2}$.

Multiplying by $\Sigma_K\big(\{(\mathbf{D}_0)_{nn}c_n^2\}\big)$ results in the desired bound. □

Motivated by the diagonal case in which the optimal solution to (3.4.3) is to set $\mathbf{D} = \mathbf{Q}$, for the diagonally dominant case we let $\mathbf{D}_0 = \operatorname{Diag}(\mathbf{Q})$ in Lemma 3, where $\operatorname{Diag}(\mathbf{Q})$ denotes the diagonal matrix with diagonal entries equal to those of $\mathbf{Q}$. Thus $\mathbf{D}$ is chosen to be an optimally scaled version of $\operatorname{Diag}(\mathbf{Q})$.

Next we relate the scale factor $\alpha = \lambda_{\min}\big(\operatorname{Diag}(\mathbf{Q})^{-1/2}\mathbf{Q}\operatorname{Diag}(\mathbf{Q})^{-1/2}\big)$ to an explicit measure of the diagonal dominance of $\mathbf{Q}$. By the Gershgorin circle theorem [92], every eigenvalue of a symmetric matrix $\mathbf{A}$ lies in one of the intervals

$$\left[A_{mm} - \sum_{n\neq m} |A_{mn}|\,, A_{mm} + \sum_{n\neq m} |A_{mn}|\right]$$

for some $m$. Applying the theorem to $\operatorname{Diag}(\mathbf{Q})^{-1/2}\mathbf{Q}\operatorname{Diag}(\mathbf{Q})^{-1/2}$ yields

$$\lambda_{\min}\big(\operatorname{Diag}(\mathbf{Q})^{-1/2}\mathbf{Q}\operatorname{Diag}(\mathbf{Q})^{-1/2}\big) \geq 1 - \max_m \sum_{n\neq m} \frac{|Q_{mn}|}{\sqrt{Q_{mm}Q_{nn}}}, \tag{3.4.14}$$

noting that $\operatorname{Diag}(\mathbf{Q})^{-1/2}\mathbf{Q}\operatorname{Diag}(\mathbf{Q})^{-1/2}$ has unit diagonal entries. Combining Lemma 3 and (3.4.14),

$$E_d(K) \geq \left(1 - \max_m \sum_{n\neq m} \frac{|Q_{mn}|}{\sqrt{Q_{mm}Q_{nn}}}\right) \Sigma_K\big(\{Q_{nn}c_n^2\}\big). \tag{3.4.15}$$

We assume in this subsection that $\mathbf{Q}$ is sufficiently diagonally dominant so that the lower bound in (3.4.15) is positive and is also an improvement over the previous bound of $\lambda_{\min}(\mathbf{Q})\Sigma_K\big(\{c_n^2\}\big)$ used in Theorem 2.

We now determine an upper bound on $E_0(K)$. Since $E_0(K)$ is defined as the minimum value of the left-hand side of (2.2.5), for any subset $\mathcal{Z}_0$ of size $K$ we have

$$\begin{aligned} E_0(K) = \min_{|\mathcal{Z}|=K} \left\{\mathbf{c}_{\mathcal{Z}}^T(\mathbf{Q}/\mathbf{Q}_{\mathcal{Y}\mathcal{Y}})\mathbf{c}_{\mathcal{Z}}\right\} &\leq \mathbf{c}_{\mathcal{Z}_0}^T(\mathbf{Q}/\mathbf{Q}_{\mathcal{Y}_0\mathcal{Y}_0})\mathbf{c}_{\mathcal{Z}_0} \\ &= \mathbf{c}_{\mathcal{Z}_0}^T\left(\mathbf{Q}_{\mathcal{Z}_0\mathcal{Z}_0} - \mathbf{Q}_{\mathcal{Z}_0\mathcal{Y}_0}\left(\mathbf{Q}_{\mathcal{Y}_0\mathcal{Y}_0}\right)^{-1}\mathbf{Q}_{\mathcal{Y}_0\mathcal{Z}_0}\right)\mathbf{c}_{\mathcal{Z}_0} \\ &\leq \mathbf{c}_{\mathcal{Z}_0}^T\mathbf{Q}_{\mathcal{Z}_0\mathcal{Z}_0}\mathbf{c}_{\mathcal{Z}_0}. \end{aligned} \tag{3.4.16}$$

We wish to choose $\mathcal{Z}_0$ so that the right-hand side of (3.4.16) is a close approximation to $E_0(K)$. Recall from Section 2.2.1 that if $\mathbf{Q}$ is diagonal, the solution to the combinatorial minimization in (2.2.5) is to choose $\mathcal{Z}$ to correspond to the $K$ smallest values of $Q_{nn}c_n^2$. For the diagonally dominant case we assume that this choice of $\mathcal{Z}$, which we denote as $\mathcal{Z}_K$, results in a good approximation to the true minimum $E_0(K)$ and we therefore set $\mathcal{Z}_0 = \mathcal{Z}_K$ in (3.4.16).

To relate the bound in (3.4.16) to the measure of diagonal dominance defined in (3.4.14), we rewrite the right-hand side of (3.4.16) (with $\mathcal{Z}_0 = \mathcal{Z}_K$) as

$$\begin{aligned}\mathbf{c}_{\mathcal{Z}_K}^T \mathbf{Q}_{\mathcal{Z}_K\mathcal{Z}_K} \mathbf{c}_{\mathcal{Z}_K} = (\mathrm{Diag}(\mathbf{Q}_{\mathcal{Z}_K\mathcal{Z}_K})^{1/2}\mathbf{c}_{\mathcal{Z}_K})^T \,\mathrm{Diag}(\mathbf{Q}_{\mathcal{Z}_K\mathcal{Z}_K})^{-1/2}\mathbf{Q}_{\mathcal{Z}_K\mathcal{Z}_K}\,\mathrm{Diag}(\mathbf{Q}_{\mathcal{Z}_K\mathcal{Z}_K})^{-1/2}& \\ \times\,(\mathrm{Diag}(\mathbf{Q}_{\mathcal{Z}_K\mathcal{Z}_K})^{1/2}\mathbf{c}_{\mathcal{Z}_K}).&\end{aligned}$$

Bounding the right-hand side in terms of the largest eigenvalue and then applying the Gershgorin circle theorem,

$$\begin{aligned}\mathbf{c}_{\mathcal{Z}_K}^T \mathbf{Q}_{\mathcal{Z}_K\mathcal{Z}_K} \mathbf{c}_{\mathcal{Z}_K} &\le \lambda_{\max}\left(\mathrm{Diag}(\mathbf{Q}_{\mathcal{Z}_K\mathcal{Z}_K})^{-1/2}\mathbf{Q}_{\mathcal{Z}_K\mathcal{Z}_K}\,\mathrm{Diag}(\mathbf{Q}_{\mathcal{Z}_K\mathcal{Z}_K})^{-1/2}\right)\sum_{n\in\mathcal{Z}_K} Q_{nn}c_n^2 \\ &\le \left(1 + \max_{m\in\mathcal{Z}_K}\sum_{\substack{n\in\mathcal{Z}_K\\ n\neq m}}\frac{|Q_{mn}|}{\sqrt{Q_{mm}Q_{nn}}}\right)\sum_{n\in\mathcal{Z}_K} Q_{nn}c_n^2 \\ &= \left(1 + \max_{m\in\mathcal{Z}_K}\sum_{\substack{n\in\mathcal{Z}_K\\ n\neq m}}\frac{|Q_{mn}|}{\sqrt{Q_{mm}Q_{nn}}}\right)\Sigma_K\big(\{Q_{nn}c_n^2\}\big), \qquad (3.4.17)\end{aligned}$$

where the last line follows from the definition of $\mathcal{Z}_K$. Combining (3.4.17) with (3.4.16), we obtain

$$E_0(K) \le \left(1 + \max_{m\in\mathcal{Z}_K}\sum_{\substack{n\in\mathcal{Z}_K\\ n\neq m}}\frac{|Q_{mn}|}{\sqrt{Q_{mm}Q_{nn}}}\right)\Sigma_K\big(\{Q_{nn}c_n^2\}\big). \qquad (3.4.18)$$

Based on the bounds in (3.4.15) and (3.4.18),

$$\underline{K}_{dd} = \max\left\{K : \left(1 + \max_{m\in\mathcal{Z}_K}\sum_{\substack{n\in\mathcal{Z}_K\\ n\neq m}}\frac{|Q_{mn}|}{\sqrt{Q_{mm}Q_{nn}}}\right)\Sigma_K\big(\{Q_{nn}c_n^2\}\big) \le \gamma\right\} \qquad (3.4.19)$$

is a lower bound on $K^*$, while

$$\overline{K}_{dd} = \max\left\{K : \left(1 - \max_m \sum_{n\neq m} \frac{|Q_{mn}|}{\sqrt{Q_{mm}Q_{nn}}}\right) \Sigma_K\left(\{Q_{nn}c_n^2\}\right) \leq \gamma\right\} \tag{3.4.20}$$

is an upper bound on $K_d$. The following theorem summarizes the relationships among $K^*$, $K_d$, $\underline{K}_{dd}$, and $\overline{K}_{dd}$. The proof of the bound on the ratio $\overline{K}_{dd}/\underline{K}_{dd}$ is similar to the proof of Lemma 2.

**Theorem 3.** *Assume that* $\mathbf{Q}$ *is diagonally dominant in the sense that*

$$\max_m \sum_{n\neq m} \frac{|Q_{mn}|}{\sqrt{Q_{mm}Q_{nn}}} < 1.$$

*With* $\underline{K}_{dd}$ *and* $\overline{K}_{dd}$ *as defined in* (3.4.19) *and* (3.4.20)*, we have* $\underline{K}_{dd} \leq K^* \leq K_d \leq \overline{K}_{dd}$ *and*

$$\frac{\overline{K}_{dd}}{\underline{K}_{dd}} \leq \frac{\lceil(\underline{K}_{dd}+1)r_{dd}\rceil - 1}{\underline{K}_{dd}+1} \approx r_{dd},$$

*where*

$$r_{dd} = \left(1 + \max_{m\in\mathcal{Z}_{\underline{K}_{dd}+1}} \sum_{\substack{n\in\mathcal{Z}_{\underline{K}_{dd}+1}\\ n\neq m}} \frac{|Q_{mn}|}{\sqrt{Q_{mm}Q_{nn}}}\right) \Bigg/ \left(1 - \max_m \sum_{n\neq m} \frac{|Q_{mn}|}{\sqrt{Q_{mm}Q_{nn}}}\right).$$

The ratio $r_{dd}$ plays the same role in Theorem 3 as the condition number $\kappa(\mathbf{Q})$ does in Theorem 2. If $\mathbf{Q}$ is strongly diagonally dominant, $r_{dd}$ is only slightly greater than 1 and therefore $\overline{K}_{dd}$ is not much larger than $\underline{K}_{dd}$. Unlike with Theorem 2, there is no benefit to generalizing Theorem 3 by means of diagonal scaling transformations because the measure of diagonal dominance that is used remains unchanged when $\mathbf{Q}$ is replaced by $\mathbf{S}^{-1}\mathbf{Q}\mathbf{S}^{-1}$.

### 3.4.6 The nearly coordinate-aligned case

In this subsection, we analyze in the same manner as in Section 3.4.5 the case in which the eigenvectors of $\mathbf{Q}$ are close to the standard basis vectors, i.e., the ellipsoid $\mathcal{E}_{\mathbf{Q}}$ is nearly coordinate-aligned. More specifically, we assume that $\mathbf{Q}$ is diagonalized as $\mathbf{Q} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T$, where the eigenvalues $\lambda_n(\mathbf{Q})$ and the orthogonal matrix $\mathbf{V}$ of eigenvectors are ordered in such a way that $\boldsymbol{\Delta} \equiv \mathbf{V} - \mathbf{I}$ is small. It is expected that the diagonal relaxation gives a close approximation in the nearly coordinate-aligned case. Our aim in this subsection is to bound the ratio between $K^*$ and $K_d$ in terms of a measure of the size of $\mathbf{\Delta}$.

To derive an upper bound on $K_d$, we use Lemma 3 to obtain a lower bound on $E_d(K)$ as was done in Section 3.4.5. In the present case, we set $\mathbf{D}_0 = \mathbf{\Lambda}$, which corresponds geometrically to restricting the coordinate-aligned ellipsoid $\mathcal{E}_{\mathbf{D}}$ to be of the same shape as $\mathcal{E}_{\mathbf{Q}}$, a reasonable choice in the nearly coordinate-aligned case. This leads to the following bound:

$$E_d(K) \geq \lambda_{\min}\big(\mathbf{\Lambda}^{-1/2}\mathbf{Q}\mathbf{\Lambda}^{-1/2}\big)\Sigma_K\big(\{\lambda_n(\mathbf{Q})c_n^2\}\big). \tag{3.4.21}$$

Given that $\mathbf{V} \approx \mathbf{I}$, the matrix $\mathbf{\Lambda}^{-1/2}\mathbf{Q}\mathbf{\Lambda}^{-1/2}$ is also approximately equal to $\mathbf{I}$ and its smallest eigenvalue is close to 1. The following lemma makes this precise by providing a lower bound on $\lambda_{\min}\big(\mathbf{\Lambda}^{-1/2}\mathbf{Q}\mathbf{\Lambda}^{-1/2}\big)$ in terms of the spectral radius $\rho(\mathbf{\Delta})$ and the condition number $\kappa(\mathbf{Q})$. We also derive an upper bound on the largest eigenvalue for later use.

**Lemma 4.** *Assume that $\mathbf{Q}$ has a diagonalization $\mathbf{Q} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$ such that $\mathbf{\Delta} = \mathbf{V} - \mathbf{I}$ is small in the sense that $\kappa(\mathbf{Q})\rho(\mathbf{\Delta}) < 1$. Then*

$$\begin{aligned}\lambda_{\min}\big(\mathbf{\Lambda}^{-1/2}\mathbf{Q}\mathbf{\Lambda}^{-1/2}\big) &\geq 1 - \kappa(\mathbf{Q})\rho(\mathbf{\Delta}),\\ \lambda_{\max}\big(\mathbf{\Lambda}^{-1/2}\mathbf{Q}\mathbf{\Lambda}^{-1/2}\big) &\leq 1 + \kappa(\mathbf{Q})\rho(\mathbf{\Delta}) + \kappa(\mathbf{Q})\rho^2(\mathbf{\Delta}).\end{aligned}$$

The proof of Lemma 4 can be found in Appendix B.6. Combining (3.4.21) and Lemma 4, we obtain

$$E_d(K) \geq \left(1 - \kappa(\mathbf{Q})\rho(\mathbf{\Delta})\right)\Sigma_K\big(\{\lambda_n(\mathbf{Q})c_n^2\}\big). \tag{3.4.22}$$

As with the bound in (3.4.15), we assume in this subsection that $\kappa(\mathbf{Q})\rho(\mathbf{\Delta})$ is small enough for the lower bound in (3.4.22) to be stronger than the bound of $\lambda_{\min}(\mathbf{Q})\Sigma_K\big(\{c_n^2\}\big)$ used in Theorem 2.

The dependence of the bound in (3.4.22) on the condition number $\kappa(\mathbf{Q})$ can be explained by the following geometric phenomenon: If $\kappa(\mathbf{Q})$ is close to 1, i.e., the original ellipsoid $\mathcal{E}_{\mathbf{Q}}$ is close to spherical, and the misalignment between the ellipsoid axes and the coordinate axes is small, then a coordinate-aligned ellipsoid only needs to be slightly larger in order to enclose $\mathcal{E}_{\mathbf{Q}}$. In the limit of $\kappa(\mathbf{Q}) = 1$, $\mathcal{E}_{\mathbf{Q}}$ is spherical and thus already coordinate-aligned. This agrees with (3.4.22) since in the spherical case $\mathbf{V}$ can be chosen equal to $\mathbf{I}$ and $\rho(\mathbf{\Delta}) = 0$. On the other hand, if $\kappa(\mathbf{Q})$ is large, even a small misalignment between the ellipsoid and coordinate axes results in a much larger coordinate-aligned enclosing ellipsoid. This behavior is illustrated in Fig. 3-6 in the two-dimensional case.

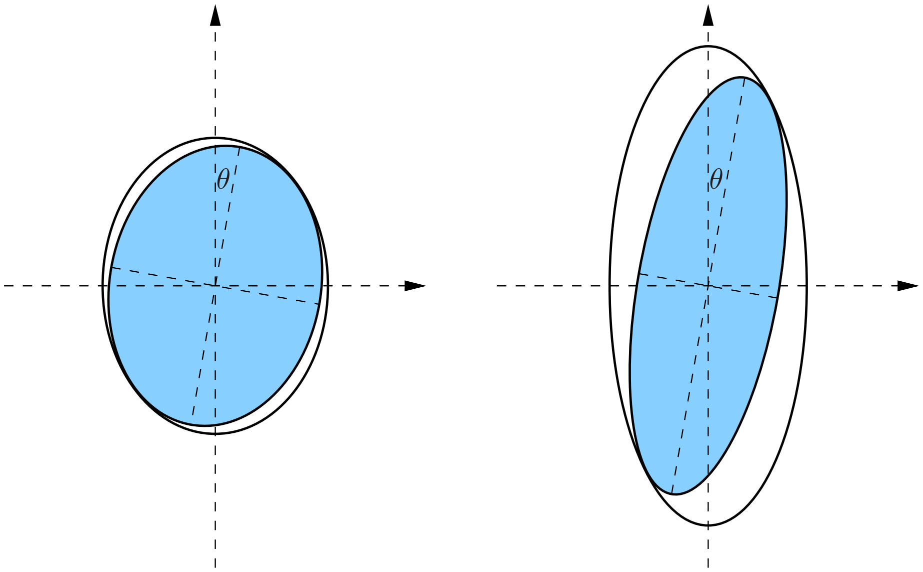


Figure 3-6: Relationship between the approximation quality of the diagonal relaxation and the condition number of $\mathbf{Q}$ in the nearly coordinate-aligned case. For the same small angular offset $\theta$ between the axes of the original ellipsoid and the coordinate axes, the coordinate-aligned enclosing ellipsoid on the right is comparatively larger.

We now determine an upper bound on $E_0(K)$ in the nearly coordinate-aligned case. We again make use of the bound in (3.4.16), this time choosing $\mathcal{Z}_0$ to correspond to the $K$ smallest $\lambda_n(\mathbf{Q})c_n^2$. We refer to this subset as $\mathcal{Z}'_K$. To relate the right-hand side of (3.4.16) to the proximity measure $\kappa(\mathbf{Q})\rho(\boldsymbol{\Delta})$ appearing in (3.4.22), we rewrite the first as follows:

$$\begin{aligned}\mathbf{c}_{\mathcal{Z}'_K}^T \mathbf{Q}_{\mathcal{Z}'_K\mathcal{Z}'_K}\mathbf{c}_{\mathcal{Z}'_K} &= \begin{bmatrix}\mathbf{c}_{\mathcal{Z}'_K}^T & \mathbf{0}\end{bmatrix} \begin{bmatrix} \mathbf{Q}_{\mathcal{Z}'_K\mathcal{Z}'_K} & \mathbf{Q}_{\mathcal{Z}'_K\mathcal{Y}'_K}\\ \mathbf{Q}_{\mathcal{Y}'_K\mathcal{Z}'_K} & \mathbf{Q}_{\mathcal{Y}'_K\mathcal{Y}'_K}\end{bmatrix}\begin{bmatrix}\mathbf{c}_{\mathcal{Z}'_K}\\ \mathbf{0}\end{bmatrix}\\ &= \left(\boldsymbol{\Lambda}^{1/2}\begin{bmatrix}\mathbf{c}_{\mathcal{Z}'_K}\\ \mathbf{0}\end{bmatrix}\right)^T \boldsymbol{\Lambda}^{-1/2}\mathbf{Q}\boldsymbol{\Lambda}^{-1/2}\left(\boldsymbol{\Lambda}^{1/2}\begin{bmatrix}\mathbf{c}_{\mathcal{Z}'_K}\\ \mathbf{0}\end{bmatrix}\right).\end{aligned}$$

Bounding the right-hand side in terms of $\lambda_{\max}\big(\boldsymbol{\Lambda}^{-1/2}\mathbf{Q}\boldsymbol{\Lambda}^{-1/2}\big)$ and combining with Lemma 4 and (3.4.16), we arrive at

$$E_0(K) \leq \big(1 + \kappa(\mathbf{Q})\rho(\boldsymbol{\Delta}) + \kappa(\mathbf{Q})\rho^2(\boldsymbol{\Delta})\big)\, \Sigma_K\big(\{\lambda_n(\mathbf{Q})c_n^2\}\big). \tag{3.4.23}$$

Equations (3.4.22) and (3.4.23) imply that

$$\underline{K}_{naa} = \max\left\{K : \left(1 + \kappa(\mathbf{Q})\rho(\mathbf{\Delta}) + \kappa(\mathbf{Q})\rho^2(\mathbf{\Delta})\right) \Sigma_K\left(\{\lambda_n(\mathbf{Q})c_n^2\}\right) \leq \gamma\right\} \tag{3.4.24}$$

is a lower bound on $K^*$ and

$$\overline{K}_{naa} = \max\left\{K : (1 - \kappa(\mathbf{Q})\rho(\mathbf{\Delta})) \Sigma_K\left(\{\lambda_n(\mathbf{Q})c_n^2\}\right) \leq \gamma\right\} \tag{3.4.25}$$

is an upper bound on $K_d$. The following result is analogous to Theorem 3, with a similar proof.

**Theorem 4.** *Assume that $\mathbf{Q}$ has a diagonalization $\mathbf{Q} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$ such that $\mathbf{\Delta} = \mathbf{V} - \mathbf{I}$ is small in the sense that $\kappa(\mathbf{Q})\rho(\mathbf{\Delta}) < 1$. With $\underline{K}_{naa}$ and $\overline{K}_{naa}$ as defined in (3.4.24) and (3.4.25), we have $\underline{K}_{naa} \leq K^* \leq K_d \leq \overline{K}_{naa}$ and*

$$\frac{\overline{K}_{naa}}{\underline{K}_{naa}} \leq \frac{\lceil(\underline{K}_{naa} + 1)r_{naa}\rceil - 1}{\underline{K}_{naa} + 1} \approx r_{naa},$$

*where*

$$r_{naa} = \frac{1 + \kappa(\mathbf{Q})\rho(\mathbf{\Delta}) + \kappa(\mathbf{Q})\rho^2(\mathbf{\Delta})}{1 - \kappa(\mathbf{Q})\rho(\mathbf{\Delta})}.$$

Theorem 4 characterizes the quality of approximation in terms of the ratio $r_{naa}$. If the ellipsoid $\mathcal{E}_{\mathbf{Q}}$ is nearly coordinate-aligned and if the condition number $\kappa(\mathbf{Q})$ is low, $r_{naa}$ is close to 1 and the approximation ratio is guaranteed to be small. As with Theorem 2, there is a potential benefit to considering diagonal scaling transformations since the quantity corresponding to $\kappa(\mathbf{Q})\rho(\mathbf{\Delta})$ may decrease when $\mathbf{Q}$ is transformed into $\mathbf{S}^{-1}\mathbf{Q}\mathbf{S}^{-1}$ for certain choices of $\mathbf{S}$.

## 3.5 Efficient solution of diagonal relaxations

In Section 3.4, we introduced the diagonal relaxation of problem (2.0.1) and analyzed its approximation properties. It was seen that the diagonal relaxation can provide good lower bounds on the optimal cost of (2.0.1) for certain classes of instances, and more evidence of this will be presented in Section 3.6. However, to be useful as part of a branch-and-bound algorithm, the diagonal relaxation must also be efficiently solvable. To some extent this is ensured by the ability to reformulate the core optimization problem (3.4.3) as a semidefinite program, allowing the use of efficient solvers such as SDPT3 [99, 100] and SeDuMi [101] (both accessible via the `cvx` interface [102]). In our experience however, these general-purpose solvers are not as efficient at solving (3.4.3) as a more specialized solver. In this section, we describe some techniques that exploit the structure of the problem at hand and thereby increase efficiency when combined with standard interior-point algorithms for semidefinite optimization.

We begin in Section 3.5.1 by rescaling problem (3.4.3) to normalize the vector $\mathbf{c}$ and simplify the presentation in the remainder of the section. In Section 3.5.2, we discuss how to make more efficient the search for $K_d$, the largest value of $K$ such that the optimal value of (3.4.3) is no greater than $\gamma$. We then focus on the rescaled version of problem (3.4.3). In Section 3.5.3, (3.4.3) is reformulated as a standard semidefinite optimization problem and a primal-dual algorithm is introduced to solve it. The primal-dual nature of the algorithm is particularly suited to the fact that the optimal value is used only in a threshold test. Later subsections describe how to improve the efficiency of specific aspects of the primal-dual algorithm, namely the determination of initial solutions, search directions, and step sizes.

### 3.5.1 Normalization of the vector c

In Section 3.4.3, it was shown that the optimal value $E_d(K)$ of (3.4.3) does not change when a diagonal scaling $\mathbf{S}$ is applied to the ellipsoid $\mathcal{E}_{\mathbf{Q}}$. To facilitate the presentation in the remainder of the section, we now fix a particular choice for $\mathbf{S}$ that normalizes the ellipsoid center $\mathbf{c}$. Denoting by $\mathcal{S}$ the set of $n$ for which $c_n = 0$ and by $\mathcal{T}$ the complement of $\mathcal{S}$, $\mathbf{S}$ is chosen as follows:

$$S_{nn} = \begin{cases} \dfrac{1}{c_n}, & n \in \mathcal{T}, \\ 1, & n \in \mathcal{S}. \end{cases} \tag{3.5.1}$$

We also define $\overline{\mathbf{Q}} = \mathbf{S}^{-1}\mathbf{Q}\mathbf{S}^{-1}$, i.e.,

$$\overline{Q}_{mn} = \begin{cases} c_m c_n Q_{mn}, & m, n \in \mathcal{T}, \\ c_m Q_{mn}, & m \in \mathcal{T},\ n \in \mathcal{S}, \\ c_n Q_{mn}, & m \in \mathcal{S},\ n \in \mathcal{T}, \\ Q_{mn}, & m, n \in \mathcal{S}. \end{cases} \tag{3.5.2}$$

With $\mathbf{S}$ given by (3.5.1), problem (3.4.8) becomes

$$\begin{aligned} \max_{\mathbf{D}} \quad & \Sigma_{K-|\mathcal{S}|}\big(\{D_{nn} : n \in \mathcal{T}\}\big) \\ \text{s.t.} \quad & \mathbf{0} \preceq \mathbf{D} \preceq \overline{\mathbf{Q}}, \\ & \mathbf{D} \text{ diagonal.} \end{aligned} \tag{3.5.3}$$

The objective function no longer depends on $\mathbf{c}$ and involves only the $K - |\mathcal{S}|$ smallest elements in $\{D_{nn} : n \in \mathcal{T}\}$ because the terms $D_{nn}c_n^2$ for $n \in \mathcal{S}$ are always zero.

For $K \leq |\mathcal{S}|$, the optimal value of (3.5.3) is equal to zero. For $K > |\mathcal{S}|$, it is possible to reformulate (3.5.3) to eliminate the variables $D_{nn}$ for $n \in \mathcal{S}$ and reduce the dimensionality of the problem. By expressing the positive semidefinite constraints $\mathbf{0} \preceq \mathbf{D} \preceq \overline{\mathbf{Q}}$ in terms of submatrices defined by the index sets $\mathcal{S}$ and $\mathcal{T}$, we have the equivalent constraints (from [92])

$$\mathbf{0} \preceq \mathbf{D}_{\mathcal{S}\mathcal{S}} \preceq \overline{\mathbf{Q}}_{\mathcal{S}\mathcal{S}}, \tag{3.5.4a}$$

$$\mathbf{0} \preceq \mathbf{D}_{\mathcal{T}\mathcal{T}} \preceq \overline{\mathbf{Q}}_{\mathcal{T}\mathcal{T}} - \overline{\mathbf{Q}}_{\mathcal{T}\mathcal{S}} \left(\overline{\mathbf{Q}}_{\mathcal{S}\mathcal{S}} - \mathbf{D}_{\mathcal{S}\mathcal{S}}\right)^{-1} \overline{\mathbf{Q}}_{\mathcal{S}\mathcal{T}}. \tag{3.5.4b}$$

Suppose that a pair $(\mathbf{D}_{\mathcal{S}\mathcal{S}}, \mathbf{D}_{\mathcal{T}\mathcal{T}})$ satisfies the constraints in (3.5.4), i.e., it is feasible for (3.5.3). Then the pair $(\mathbf{0}, \mathbf{D}_{\mathcal{T}\mathcal{T}})$ has the same objective value as $(\mathbf{D}_{\mathcal{S}\mathcal{S}}, \mathbf{D}_{\mathcal{T}\mathcal{T}})$ since the variables $D_{nn}$, $n \in \mathcal{S}$, do not appear in the objective, and $(\mathbf{0}, \mathbf{D}_{\mathcal{T}\mathcal{T}})$ satisfies (3.5.4a). To see that $(\mathbf{0}, \mathbf{D}_{\mathcal{T}\mathcal{T}})$ also satisfies (3.5.4b) whenever $(\mathbf{D}_{\mathcal{S}\mathcal{S}}, \mathbf{D}_{\mathcal{T}\mathcal{T}})$ does, consider the following chain of inequalities based on well-known properties of semidefinite matrices (see [92]):

$$\begin{aligned} \mathbf{D}_{\mathcal{S}\mathcal{S}} \succeq \mathbf{0} \quad \implies & \quad \overline{\mathbf{Q}}_{\mathcal{S}\mathcal{S}} - \mathbf{D}_{\mathcal{S}\mathcal{S}} \preceq \overline{\mathbf{Q}}_{\mathcal{S}\mathcal{S}} \\ \implies & \quad \left(\overline{\mathbf{Q}}_{\mathcal{S}\mathcal{S}} - \mathbf{D}_{\mathcal{S}\mathcal{S}}\right)^{-1} \succeq \left(\overline{\mathbf{Q}}_{\mathcal{S}\mathcal{S}}\right)^{-1} \\ \implies & \quad \overline{\mathbf{Q}}_{\mathcal{T}\mathcal{S}} \left(\overline{\mathbf{Q}}_{\mathcal{S}\mathcal{S}} - \mathbf{D}_{\mathcal{S}\mathcal{S}}\right)^{-1} \overline{\mathbf{Q}}_{\mathcal{S}\mathcal{T}} \succeq \overline{\mathbf{Q}}_{\mathcal{T}\mathcal{S}} \left(\overline{\mathbf{Q}}_{\mathcal{S}\mathcal{S}}\right)^{-1} \overline{\mathbf{Q}}_{\mathcal{S}\mathcal{T}} \\ \implies & \quad \overline{\mathbf{Q}}_{\mathcal{T}\mathcal{T}} - \overline{\mathbf{Q}}_{\mathcal{T}\mathcal{S}} \left(\overline{\mathbf{Q}}_{\mathcal{S}\mathcal{S}} - \mathbf{D}_{\mathcal{S}\mathcal{S}}\right)^{-1} \overline{\mathbf{Q}}_{\mathcal{S}\mathcal{T}} \preceq \overline{\mathbf{Q}}_{\mathcal{T}\mathcal{T}} - \overline{\mathbf{Q}}_{\mathcal{T}\mathcal{S}} \left(\overline{\mathbf{Q}}_{\mathcal{S}\mathcal{S}}\right)^{-1} \overline{\mathbf{Q}}_{\mathcal{S}\mathcal{T}}. \end{aligned}$$

Combining the last inequality with (3.5.4b),

$$\mathbf{D}_{\mathcal{T}\mathcal{T}} \preceq \overline{\mathbf{Q}}_{\mathcal{T}\mathcal{T}} - \overline{\mathbf{Q}}_{\mathcal{T}\mathcal{S}} \left(\overline{\mathbf{Q}}_{\mathcal{S}\mathcal{S}} - \mathbf{D}_{\mathcal{S}\mathcal{S}}\right)^{-1} \overline{\mathbf{Q}}_{\mathcal{S}\mathcal{T}} \preceq \overline{\mathbf{Q}}_{\mathcal{T}\mathcal{T}} - \overline{\mathbf{Q}}_{\mathcal{T}\mathcal{S}} \left(\overline{\mathbf{Q}}_{\mathcal{S}\mathcal{S}}\right)^{-1} \overline{\mathbf{Q}}_{\mathcal{S}\mathcal{T}} = \overline{\mathbf{Q}}/\overline{\mathbf{Q}}_{\mathcal{S}\mathcal{S}},$$

which implies that (3.5.4b) holds with $\mathbf{D}_{\mathcal{S}\mathcal{S}} = \mathbf{0}$ if it holds for any other value for $\mathbf{D}_{\mathcal{S}\mathcal{S}}$. We have thus shown that $\mathbf{D}_{\mathcal{S}\mathcal{S}}$ can be set to $\mathbf{0}$ in problem (3.5.3) without loss of optimality, yielding the reduced problem

$$\begin{aligned} \max_{\mathbf{D}_{\mathcal{T}\mathcal{T}}} \quad & \Sigma_{K-|\mathcal{S}|}\big(\{D_{nn} : n \in \mathcal{T}\}\big) \\ \text{s.t.} \quad & \mathbf{0} \preceq \mathbf{D}_{\mathcal{T}\mathcal{T}} \preceq \overline{\mathbf{Q}}/\overline{\mathbf{Q}}_{\mathcal{S}\mathcal{S}}, \\ & \mathbf{D}_{\mathcal{T}\mathcal{T}} \text{ diagonal}, \end{aligned} \tag{3.5.5}$$

which involves only the variables $D_{nn}$, $n \in \mathcal{T}$.

For ease of notation, we assume henceforth that none of the entries of $\mathbf{c}$ are equal to zero, i.e., that $\mathcal{S} = \emptyset$ and $\mathcal{T} = \{1, \ldots, N\}$ in (3.5.3). Sections 3.5.3–3.5.6 concentrate on solving (3.5.3) under this assumption. If $\mathbf{c}$ does have zero-valued entries, the same methods can be used to solve (3.5.5) instead.

From a numerical standpoint, it is advisable to expand the set $\mathcal{S}$ to include not only the indices $n$ for which $c_n = 0$, but also those $n$ for which the product $Q_{nn}c_n^2$ is very small relative to $\gamma$. This ensures that none of the diagonal entries of $\overline{\mathbf{Q}}$, which are equal to $Q_{nn}c_n^2$ according to (3.5.2), are close to zero, thereby improving the conditioning of $\overline{\mathbf{Q}}$ as well as that of other matrices used in solving (3.5.3). Moreover, the removal of the variables $D_{nn}$, $n \in \mathcal{S}$ incurs a negligible loss of optimality. Under the constraint $\mathbf{D} \preceq \mathbf{Q}$, we have $D_{nn}c_n^2 \leq Q_{nn}c_n^2$ for every $n$, so if $Q_{nn}c_n^2$ is small compared to $\gamma$, the contribution of $D_{nn}c_n^2$ to the sum in (3.4.3) is also small.

### 3.5.2 Search over $K$

As discussed in Section 3.4.1, solving the diagonal relaxation involves a search over $K = 1, \ldots N$ to determine $K_d$, the largest $K$ such that $E_d(K) \leq \gamma$. Thus the computational complexity depends on the number of values of $K$ for which $E_d(K)$ needs to be evaluated, keeping in mind that each evaluation of $E_d(K)$ requires solving or re-solving problem (3.5.3). We describe in this subsection how the required number of evaluations of $E_d(K)$ may be reduced.

First we observe that $E_d(K)$ increases monotonically with $K$. This follows because the objective function in (3.5.3) increases monotonically with $K$ for every fixed $\mathbf{D}$. The monotonicity of $E_d(K)$ allows the transition point $K_d$ between $E_d(K) \leq \gamma$ and $E_d(K) > \gamma$ to be determined through a bisection search. Starting from an initial interval of $K$ values, $E_d(K)$ is evaluated at the midpoint $K_{\text{mid}}$ of the current interval (rounded to the nearest integer if necessary). If $E_d(K_{\text{mid}}) \leq \gamma$, then $E_d(K) \leq \gamma$ for all $K < K_{\text{mid}}$ and the lower limit of the new interval is set to $K_{\text{mid}} + 1$. Similarly if $E_d(K_{\text{mid}}) > \gamma$, $E_d(K) > \gamma$ for all $K > K_{\text{mid}}$ and the upper limit of the new interval is set to $K_{\text{mid}} - 1$. The process continues until the interval is empty, at which point $K_d$ is determined. The number of evaluations of $E_d(K)$ in the worst case is approximately $\log_2 N$, which is better than the worst case of $N$ evaluations for a linear search.

Furthermore, it is not necessary to initialize the bisection search with the full interval $[1, N]$. In particular, the lower limit can be increased based on an easily computed upper bound on $E_d(K)$ which we now derive. Given the constraint $\mathbf{D} \preceq \overline{\mathbf{Q}}$ in (3.5.3), the $n$th smallest eigenvalue of $\mathbf{D}$ is bounded from above by the $n$th smallest eigenvalue of $\overline{\mathbf{Q}}$ for all $n$ [92]. Since the eigenvalues of $\mathbf{D}$ are also its diagonal elements, it follows that

$$\Sigma_K\big(\{D_{nn}\}\big) \leq \sum_{n=1}^{K} \lambda_n(\overline{\mathbf{Q}}) \tag{3.5.6}$$

where the eigenvalues of $\overline{\mathbf{Q}}$ are indexed from smallest to largest. Since (3.5.6) is true for all $\mathbf{D}$ such that $\mathbf{D} \preceq \overline{\mathbf{Q}}$, it is true for the $\mathbf{D}$ that maximizes $\Sigma_K\big(\{D_{nn}\}\big)$, and hence

$$E_d(K) \leq \sum_{n=1}^{K} \lambda_n(\overline{\mathbf{Q}}). \tag{3.5.7}$$

Define $K_{\overline{\mathbf{Q}}}$ to be the largest value of $K$ for which the right-hand side of (3.5.7) is less than or equal to $\gamma$. Then (3.5.7) implies that $E_d(K_{\overline{\mathbf{Q}}}) \leq \gamma$, and hence the lower limit of the initial interval can be set to $K_{\overline{\mathbf{Q}}} + 1$, the smallest $K$ for which the relationship of $E_d(K)$ to $\gamma$ is not yet known. Note that evaluating $K_{\overline{\mathbf{Q}}}$ requires knowledge only of the eigenvalues of $\overline{\mathbf{Q}}$.

It is also possible to initialize the upper limit of the interval to a value smaller than $N$ if the diagonal relaxation is being solved in an attempt to improve upon an existing lower bound on the optimal cost of a subproblem. Suppose that the optimal cost of a subproblem is known to be bounded from below by $LB$, possibly as a result of the low-complexity techniques of Section 3.2 or a lower bound inherited from the parent subproblem. Given the existing lower bound $LB$, the solution to the diagonal relaxation represents an improvement only if $K_d < N - LB$. Otherwise if $K_d \geq N - LB$, the exact value of $K_d$ does not need to be determined. It follows that the largest value of $K$ for which $E_d(K)$ needs to be evaluated is $N - LB$. If $E_d(N - LB) \le \gamma$, then we infer that $K_d \ge N - LB$ and terminate, otherwise $K_d < N - LB$ and the search continues.

In the remainder of this section, we assume that the existing lower bound $LB$ is always non-zero, implying that it is not necessary to set $K = N$. This assumption is met in the case of the branch-and-bound algorithm described in Section 3.7, which uses the low-complexity tests of Section 3.2 to establish non-zero lower bounds.

### 3.5.3 Semidefinite reformulation and primal-dual algorithms

In the remainder of Section 3.5, we focus on solving problem (3.5.3) (under the assumption that $\mathcal{S} = \emptyset$) for a fixed value of $K$. As written, problem (3.5.3) involves the non-differentiable function $\Sigma_K$. It is shown in this subsection that the non-differentiability can be avoided by recasting (3.5.3) as a standard semidefinite optimization problem. Among the many algorithms available for solving semidefinite optimization problems, we explain how primal-dual algorithms in particular allow the threshold test $E_d(K) \le \gamma$ to be concluded with fewer iterations.

To derive a semidefinite formulation of problem (3.5.3), we begin by expressing the function $\Sigma_K$ in an alternative way. Specifically, $\Sigma_K\big(\{D_{nn}\}\big)$ is equal to the optimal value of the following linear program:

$$\begin{aligned} \min_{\mathbf{t}} \quad & \mathbf{d}^T\mathbf{t} \\ \text{s.t.} \quad & \mathbf{e}^T\mathbf{t} = K, \\ & \mathbf{0} \le \mathbf{t} \le \mathbf{e}, \end{aligned} \tag{3.5.8}$$

where $\mathbf{d} \equiv \operatorname{diag}(\mathbf{D})$. The equivalence holds because the minimum in (3.5.8) is attained with $t_n = 1$ for $n$ corresponding to the $K$ smallest $D_{nn}$ and $t_n = 0$ otherwise. The linear programming dual of problem (3.5.8) is given by

$$\begin{aligned} \max_{y_0, \mathbf{v}} \quad & K y_0 + \mathbf{e}^T\mathbf{v} \\ \text{s.t.} \quad & \mathbf{d} - y_0\mathbf{e} - \mathbf{v} \ge \mathbf{0}, \\ & \mathbf{v} \le \mathbf{0}, \end{aligned} \tag{3.5.9}$$

and its optimal value is also equal to $\Sigma_K\big(\{D_{nn}\}\big)$. Substituting (3.5.9) into (3.5.3), combining the two maximizations into one, and making the change of variables $\mathbf{w} = \mathbf{d} - y_0\mathbf{e}$, we arrive at

$$\begin{aligned} \max_{y_0,\mathbf{v},\mathbf{w}} \quad & Ky_0 + \mathbf{e}^T\mathbf{v} \\ \text{s.t.} \quad & \mathbf{0} \preceq y_0\mathbf{I} + \operatorname{Diag}(\mathbf{w}) \preceq \overline{\mathbf{Q}}, \\ & \mathbf{w} - \mathbf{v} \geq \mathbf{0}, \\ & \mathbf{v} \leq \mathbf{0}, \end{aligned} \tag{3.5.10}$$

where $\operatorname{Diag}(\mathbf{w})$ denotes a diagonal matrix with the entries of $\mathbf{w}$ along the diagonal. It will be seen in Section 3.5.5 that the change from $\mathbf{d}$ to $\mathbf{w}$ results in a convenient block structure for the matrix used to determine search directions. Problem (3.5.10) may be rewritten in the standard form consisting of a linear objective and a linear matrix inequality, i.e.,

$$\begin{aligned} \max_{y_0,\mathbf{v},\mathbf{w}} \quad & Ky_0 + \mathbf{e}^T\mathbf{v} + \mathbf{0}^T\mathbf{w} \\ \text{s.t.} \quad & \mathbf{S} \equiv \mathbf{C} - y_0\mathbf{A}_0 - \sum_{n=1}^{N} v_n\mathbf{A}_n - \sum_{n=1}^{N} w_n\mathbf{A}_{N+n} \succeq \mathbf{0}, \end{aligned} \tag{3.5.11}$$

where

$$\mathbf{A}_0 = \begin{bmatrix} \mathbf{I} & & & \\ & -\mathbf{I} & & \\ & & \mathbf{0} & \\ & & & \mathbf{0} \end{bmatrix}, \tag{3.5.12a}$$

$$\mathbf{A}_n = \begin{bmatrix} \mathbf{0} & & & \\ & \mathbf{0} & & \\ & & \mathbf{E}_n & \\ & & & \mathbf{E}_n \end{bmatrix}, \quad n = 1, \ldots, N, \tag{3.5.12b}$$

$$\mathbf{A}_n = \begin{bmatrix} \mathbf{E}_n & & & \\ & -\mathbf{E}_n & & \\ & & -\mathbf{E}_n & \\ & & & \mathbf{0} \end{bmatrix}, \quad n = N+1, \ldots, 2N, \tag{3.5.12c}$$

$$\mathbf{C} = \begin{bmatrix} \overline{\mathbf{Q}} & & & \\ & \mathbf{0} & & \\ & & \mathbf{0} & \\ & & & \mathbf{0} \end{bmatrix}, \tag{3.5.12d}$$

and $\mathbf{E}_n = \mathrm{Diag}(\mathbf{e}_n)$.

Since problem (3.5.10) is to be solved using a primal-dual algorithm, we also require the dual problem given as follows:

$$\begin{aligned} \min_{\mathbf{X}} \quad & \mathbf{C} \bullet \mathbf{X} \\ \text{s.t.} \quad & \mathbf{A}_0 \bullet \mathbf{X} = K, \\ & \mathbf{A}_n \bullet \mathbf{X} = 1, \quad n = 1, \ldots, N, \\ & \mathbf{A}_n \bullet \mathbf{X} = 0, \quad n = N+1, \ldots, 2N, \\ & \mathbf{X} \succeq \mathbf{0}, \end{aligned}$$

where $\mathbf{A} \bullet \mathbf{X} = \mathrm{tr}(\mathbf{AX})$ denotes the standard inner product between symmetric matrices. It is straightforward to verify that both the primal and the dual are strictly feasible and hence the optimal value of the dual is also equal to $E_d(K)$. The $4N \times 4N$ matrix $\mathbf{X}$ may be assumed to have the same sparsity pattern as $\mathbf{A}_n$ and $\mathbf{C}$ and can therefore be partitioned as

$$\mathbf{X} = \begin{bmatrix} \mathbf{Z} & & & \\ & \mathrm{Diag}(\mathbf{s}) & & \\ & & \mathrm{Diag}(\mathbf{t}) & \\ & & & \mathrm{Diag}(\mathbf{u}) \end{bmatrix}, \tag{3.5.13}$$

where $\mathbf{Z}$ is an $N \times N$ symmetric matrix and $\mathbf{s}$, $\mathbf{t}$, and $\mathbf{u}$ are $N$-dimensional vectors. This allows the dual problem to be rewritten with fewer decision variables as

$$\begin{aligned} \min_{\mathbf{Z},\mathbf{s},\mathbf{t},\mathbf{u}} \quad & \overline{\mathbf{Q}} \bullet \mathbf{Z} \\ \text{s.t.} \quad & \mathrm{diag}(\mathbf{Z}) - \mathbf{s} = \mathbf{t}, \\ & \mathbf{e}^T \mathbf{t} = K, \\ & \mathbf{t} + \mathbf{u} = \mathbf{e}, \\ & \mathbf{Z} \succeq \mathbf{0}, \quad \mathbf{s} \geq \mathbf{0}, \quad \mathbf{t} \geq \mathbf{0}, \quad \mathbf{u} \geq \mathbf{0}. \end{aligned} \tag{3.5.14}$$

Primal-dual algorithms solve the primal (3.5.10) and dual (3.5.14) problems simultaneously, iteratively improving solutions in both spaces. The relationship of the common optimal value $E_d(K)$ to the threshold $\gamma$ can be determined without solving either problem to optimality. First, since the primal is a maximization, the objective value of any feasible solution to the primal is by definition a lower bound on $E_d(K)$. Thus the algorithm can terminate as soon as a primal solution has been obtained with an objective value greater than $\gamma$, implying that $E_d(K) > \gamma$. Conversely, the dual is a minimization and any dual solution must have an objective value greater than or equal to $E_d(K)$. As soon as the dual objective value falls below $\gamma$, the algorithm can terminate with the conclusion $E_d(K) \leq \gamma$. Thus a primal-dual algorithm can decide whether or not $E_d(K)$ is greater than $\gamma$ in fewer iterations than a primal-only or dual-only algorithm.

The pair of semidefinite optimization problems (3.5.10) and (3.5.14) can be solved using a variety of general-purpose primal-dual algorithms. Our aim in Sections 3.5.4–3.5.6 is to show how the efficiency of these algorithms can be improved by exploiting both the algebraic structure of (3.5.10) and (3.5.14) as well as any existing solutions for previous values of $K$. We work with a particular potential-reduction algorithm from [103, 104] in which the potential function is given by

$$\varphi = (4N + 2\nu\sqrt{N}) \ln(\overline{\mathbf{Q}} \bullet \mathbf{Z} - K y_0 - \mathbf{e}^T \mathbf{v}) - \ln\det \mathbf{S} - \ln\det \mathbf{X} - 4N \ln 4N. \qquad (3.5.15)$$

The first term in (3.5.15) represents a penalty on the duality gap $\overline{\mathbf{Q}} \bullet \mathbf{Z} - K y_0 - \mathbf{e}^T \mathbf{v}$, the difference between the primal and dual objective values. The duality gap is non-negative for all feasible primal and dual solutions and is zero at an optimal pair of solutions. Consequently it is used as a measure of optimality. The second and third terms in (3.5.15) are barrier functions that enforce the constraints $\mathbf{S} \succeq \mathbf{0}$ and $\mathbf{X} \succeq \mathbf{0}$, so-called because their values become infinite as $\mathbf{S}$ or $\mathbf{X}$ approach the boundary of the positive semidefinite cone. The parameter $\nu$ controls the relative weight of the duality gap term; computational experiments indicate that choosing $\nu \sim 20$ yields faster convergence.

We use the duality gap and the number of iterations as secondary stopping criteria in addition to comparing the primal and dual objective values to the threshold $\gamma$ as previously discussed. If neither of the terminating conditions involving $\gamma$ is met first, the algorithm is terminated when the duality gap falls below a tolerance or when the number of iterations exceeds a maximum limit.

### 3.5.4 Initialization

Primal-dual interior-point algorithms require as input an initial solution in the interior of the feasible set for both the primal and the dual. Choosing an initial solution closer to the optimal solution naturally leads to faster convergence. In the present context, multiple instances of (3.5.10) and (3.5.14) are solved with the only difference being the value of $K$. It is reasonable therefore to expect that initializing based on the final solution for the previous value of $K$ results in faster convergence compared to independent initialization. We describe in this subsection how existing solutions can be modified to become feasible under a new value of $K$. We also develop initial primal and dual solutions for the first value of $K$ for which existing solutions are not available.

Consider first the case in which a previous instance of (3.5.10) and (3.5.14) has been solved, and let $K_c$ and $K_p$ denote the current and previous values of $K$. Since the feasible set for the primal (3.5.10) does not depend on $K$, no modification is needed to reuse the final primal solution for $K = K_p$ as the initial primal solution for $K = K_c$. We concentrate therefore on modifying the dual solution and consider the two cases $K_c > K_p$ and $K_c < K_p$. For the case $K_c > K_p$, we first observe that the second constraint in (3.5.14) can be equivalently replaced by the constraint $\mathbf{e}^T\mathbf{u} = N - K$. The equivalence can be seen by multiplying the third constraint in (3.5.14) from the left by $\mathbf{e}^T$. Based on this alternative set of constraints with $\mathbf{e}^T\mathbf{u} = N - K$, we construct an initial dual solution for $K = K_c$ (denoted using a subscript $c$) in terms of the final dual solution for $K = K_p$ (denoted using a subscript $p$) as follows:

$$\mathbf{u}_c = \frac{N - K_c}{N - K_p}\mathbf{u}_p, \tag{3.5.16a}$$

$$\mathbf{t}_c = \mathbf{e} - \mathbf{u}_c, \tag{3.5.16b}$$

$$\mathbf{s}_c = \mathbf{s}_p, \tag{3.5.16c}$$

$$\begin{aligned}(\mathbf{Z}_c)_{nn} &= (\mathbf{s}_c)_n + (\mathbf{t}_c)_n, \quad n = 1, \ldots, N, \\ (\mathbf{Z}_c)_{mn} &= (\mathbf{Z}_p)_{mn}, \quad m \neq n.\end{aligned} \tag{3.5.16d}$$

When $K_c > K_p$, the multiplier in front of $\mathbf{u}_p$ in (3.5.16a) is between 0 and 1 and scales

$\mathbf{u}_p$ down to satisfy the new constraint $\mathbf{e}^T\mathbf{u}_c = N - K_c$ while preserving positivity. Note that the barrier term $-\ln\det\mathbf{X}$ in (3.5.15) ensures that $\mathbf{Z}_p$, $\mathbf{s}_p$, $\mathbf{t}_p$, and $\mathbf{u}_p$ are strictly positive definite. The vector $\mathbf{t}_c$ satisfies the third constraint in (3.5.14) by construction and remains positive because $\mathbf{t}_p = \mathbf{e} - \mathbf{u}_p$ is positive and $\mathbf{u}_c < \mathbf{u}_p$ component-wise. Equation (3.5.16d) enforces the first constraint in (3.5.14) and has the effect of adding a positive definite diagonal matrix to $\mathbf{Z}_p$, thus keeping $\mathbf{Z}_c$ positive definite.

For the case $K_c < K_p$ we use instead the following initialization:

$$\mathbf{t}_c = \frac{K_c}{K_p}\mathbf{t}_p, \tag{3.5.17a}$$

$$\mathbf{u}_c = \mathbf{e} - \mathbf{t}_c, \tag{3.5.17b}$$

$$\mathbf{Z}_c = \mathbf{Z}_p, \tag{3.5.17c}$$

$$(\mathbf{s}_c)_n = (\mathbf{Z}_c)_{nn} - (\mathbf{t}_c)_n, \quad n = 1, \ldots, N. \tag{3.5.17d}$$

Now $\mathbf{t}_p$ is scaled down to satisfy the new constraint $\mathbf{e}^T\mathbf{t}_c = K_c$ while $\mathbf{u}_c$ and $\mathbf{s}_c$ are increased relative to $\mathbf{u}_p$ and $\mathbf{s}_p$. Equations (3.5.17) are consistent with all constraints in (3.5.14) for $K = K_c$. Our computational experiments indicate that initializing the solver according to (3.5.16) or (3.5.17) decreases the solution time by a factor of 2 to 3 relative to independent initialization.

For the first value of $K$, existing solutions to (3.5.10) and (3.5.14) are not available. In this case we suggest the following approach to obtain initial interior solutions. First, based on the fact that $\mathbf{D} = \lambda_{\min}(\overline{\mathbf{Q}})\mathbf{I}$ is a positive definite feasible solution to (3.5.3), we set $\mathbf{D} = (1-\epsilon)\lambda_{\min}(\overline{\mathbf{Q}})\mathbf{I}$ with $\epsilon$ a small positive constant to ensure that $\mathbf{D} \prec \overline{\mathbf{Q}}$ strictly. We then constrain $\mathbf{v}$, $\mathbf{w}$, $\mathbf{s}$, $\mathbf{t}$, and $\mathbf{u}$ to be proportional to $\mathbf{e}$ and $\mathbf{Z}$ to be proportional to $\mathbf{I}$ with constants of proportionality $v_0$, $w_0$, $s_0$, $t_0$, $u_0$, and $Z_0$ respectively. From the relation $\mathbf{d} = y_0\mathbf{e} + \mathbf{w}$ and the constraints in (3.5.14), we infer that

$$\begin{aligned} y_0 &= d_0 - w_0, \\ t_0 &= \frac{K}{N}, \\ u_0 &= \frac{N-K}{N}, \\ Z_0 &= s_0 + \frac{K}{N}, \end{aligned} \tag{3.5.18}$$

where $d_0 = (1-\epsilon)\lambda_{\min}(\overline{\mathbf{Q}})$. The remaining constants $s_0$, $v_0$, and $w_0$ are determined by approximately minimizing the potential function (3.5.15) with respect to them. Incorporating our restrictions on the variables, using (3.5.18), and neglecting constant terms, (3.5.15) becomes

$$\varphi = (4N + 2\nu\sqrt{N}) \ln \eta_0 - N \left[ \ln\left(s_0 + \frac{K}{N}\right) + \ln s_0 + \ln(w_0 - v_0) + \ln(-v_0) \right],$$

where

$$\eta_0 = \operatorname{tr}(\overline{\mathbf{Q}}) \left(s_0 + \frac{K}{N}\right) - K(d_0 - w_0) - N v_0$$

is the initial duality gap. Setting the derivatives with respect to $s_0$, $v_0$, and $w_0$ equal to zero and solving the resulting set of equations gives

$$\begin{aligned} w_0 &= \frac{2K - N}{K} v_0, \\ v_0 &= -\frac{\operatorname{tr}(\overline{\mathbf{Q}}) s_0 \left(s_0 + \frac{K}{N}\right)}{(N-K)\left(2s_0 + \frac{K}{N}\right)}, \end{aligned} \tag{3.5.19}$$

and

$$\frac{\operatorname{tr}(\overline{\mathbf{Q}})(4N + 2\nu\sqrt{N})}{\operatorname{tr}(\overline{\mathbf{Q}})\left(s_0 + \frac{K}{N}\right)\left(4s_0 + \frac{K}{N}\right) - K d_0 \left(2s_0 + \frac{K}{N}\right)} = \frac{N}{s_0\left(s_0 + \frac{K}{N}\right)}.$$

The last equation for $s_0$ may be simplified by assuming that $K d_0 \ll \operatorname{tr}(\overline{\mathbf{Q}})$ and neglecting the second term in the left-hand denominator, resulting in

$$s_0 = \frac{K}{2\nu\sqrt{N}}. \tag{3.5.20}$$

Back-substition into (3.5.19) and (3.5.18) completes the initialization. It can be verified that $w_0 - v_0 > 0$ and $v_0 < 0$.

### 3.5.5 Search directions

In each iteration of a primal-dual algorithm, search directions are determined in both the primal and dual solution spaces. The algorithm then searches along these directions to improve upon the current solutions. Search directions are usually computed by solving a system of linear equations. In this subsection, we demonstrate that in the case of problems (3.5.10) and (3.5.14), the system of equations corresponding to a particular method for computing search directions has a block structure that permits a solution through the inversion of an $N \times N$ matrix as opposed to the full $(2N+1) \times (2N+1)$ matrix. The reduction in dimension improves both speed and numerical accuracy. We also develop a method for correcting the computed directions to ensure that all iterates remain feasible and that inaccuracies do not propagate.

For the purpose of illustrating how simplifying block structure can arise, we restrict our attention to the HKM method of computing search directions [105–107]. Similar block structure can occur when using other methods, e.g. the primal method described in [104] and attributed to [108]. For notational convenience, we collect all of the primal variables into a single vector $\mathbf{y} = \begin{bmatrix} y_0 & \mathbf{v}^T & \mathbf{w}^T \end{bmatrix}^T$ and all of the dual variables into the matrix $\mathbf{X}$ defined in (3.5.13). The vector $\mathbf{y}$ and the matrix $\mathbf{X}$ represent the current primal and dual solutions while $\Delta\mathbf{y}$ and $\Delta\mathbf{X}$ represent the search directions to be determined. Under the HKM method, $\Delta\mathbf{y}$ is obtained by solving

$$\sum_{n=0}^{2N+1} \mathbf{A}_m \bullet \left(\mathbf{S}^{-1}\mathbf{A}_n\mathbf{X}\right) \Delta y_n = \rho \mathbf{A}_m \bullet \mathbf{X} - \mathbf{A}_m \bullet \mathbf{S}^{-1}, \quad m = 0, 1, \ldots, 2N, \tag{3.5.21}$$

where $\mathbf{S}$ is defined in (3.5.11), $\rho = (4N + 2\nu\sqrt{N})/\eta$, and $\eta$ is the current duality gap. The dual search direction $\Delta\mathbf{X}$ is given in terms of $\Delta\mathbf{y}$ as

$$\Delta\mathbf{X} = \mathbf{S}^{-1} - \rho\mathbf{X} + \mathbf{S}^{-1}\left(\sum_{n=0}^{2N} \Delta y_n \mathbf{A}_n\right)\mathbf{X}. \tag{3.5.22}$$

A straightforward calculation of the coefficients $\mathbf{A}_m \bullet \left(\mathbf{S}^{-1}\mathbf{A}_n\mathbf{X}\right)$ shows that equations (3.5.21) have the following block structure when written in matrix form:

$$\underbrace{\begin{bmatrix} M_{00} & \mathbf{0} & \mathbf{M}_{02} \\ \mathbf{0} & \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{02}^T & \mathbf{M}_{12} & \mathbf{M}_{22} \end{bmatrix}}_{\mathbf{M}} \underbrace{\begin{bmatrix} \Delta y_0 \\ \Delta\mathbf{v} \\ \Delta\mathbf{w} \end{bmatrix}}_{\Delta\mathbf{y}} = \underbrace{\begin{bmatrix} r_0 \\ \mathbf{r}_1 \\ \mathbf{r}_2 \end{bmatrix}}_{\mathbf{r}}. \tag{3.5.23}$$

On the left-hand side of (3.5.23), $M_{00}$ is a scalar given by

$$M_{00} = \mathbf{A}_0 \bullet \left(\mathbf{S}^{-1}\mathbf{A}_0\mathbf{X}\right) = \left(\overline{\mathbf{Q}} - \mathbf{D}\right)^{-1} \bullet \mathbf{Z} + \sum_{n=1}^{N} \frac{s_n}{D_{nn}},$$

$\mathbf{M}_{02}$ is a $1 \times N$ row vector with components

$$(\mathbf{M}_{02})_n = \mathbf{A}_0 \bullet \left(\mathbf{S}^{-1}\mathbf{A}_{N+n}\mathbf{X}\right) = \sum_{m=1}^{N} \left[\left((\overline{\mathbf{Q}} - \mathbf{D})^{-1}\right)_{mn} Z_{mn}\right] + \frac{s_n}{D_{nn}}, \quad n = 1, \dots, N,$$

$\mathbf{M}_{11}$ and $\mathbf{M}_{12}$ are $N \times N$ diagonal matrices with diagonal elements

$$\begin{aligned}(\mathbf{M}_{11})_{nn} &= \mathbf{A}_n \bullet \left(\mathbf{S}^{-1}\mathbf{A}_n\mathbf{X}\right) = \frac{t_n}{w_n - v_n} - \frac{u_n}{v_n}, \quad n = 1, \dots, N,\\ (\mathbf{M}_{12})_{nn} &= \mathbf{A}_n \bullet \left(\mathbf{S}^{-1}\mathbf{A}_{N+n}\mathbf{X}\right) = -\frac{t_n}{w_n - v_n}, \quad n = 1, \dots, N,\end{aligned}$$

and $\mathbf{M}_{22}$ is an $N \times N$ matrix with elements

$$(\mathbf{M}_{22})_{mn} = \mathbf{A}_{N+m} \bullet \left(\mathbf{S}^{-1}\mathbf{A}_{N+n}\mathbf{X}\right) = \begin{cases} \left((\overline{\mathbf{Q}} - \mathbf{D})^{-1}\right)_{nn} Z_{nn} + \dfrac{s_n}{D_{nn}} + \dfrac{t_n}{w_n - v_n}, & m = n,\\ \left((\overline{\mathbf{Q}} - \mathbf{D})^{-1}\right)_{mn} Z_{mn}, & m \neq n,\end{cases}$$

for $m$, $n = 1, \dots, N$. The right-hand side of (3.5.23) is given by

$$\begin{aligned} r_0 &= \rho K - \operatorname{tr}\left((\overline{\mathbf{Q}} - \mathbf{D})^{-1}\right) + \operatorname{tr}\left(\mathbf{D}^{-1}\right),\\ (\mathbf{r}_1)_n &= \rho - \frac{1}{w_n - v_n} + \frac{1}{v_n}, \quad n = 1, \dots, N,\\ (\mathbf{r}_2)_n &= -\left((\overline{\mathbf{Q}} - \mathbf{D})^{-1}\right)_{nn} + \frac{1}{D_{nn}} + \frac{1}{w_n - v_n}, \quad n = 1, \dots, N.\end{aligned}$$

The presence of zero blocks in (3.5.23) and the fact that $\mathbf{M}_{11}$ is diagonal allow for a more efficient solution to the $(2N+1) \times (2N+1)$ system of equations for $\Delta\mathbf{y}$. We define

$$\mathbf{M}_S = \mathbf{M}_{22} - \frac{1}{M_{00}}\mathbf{M}_{02}^T\mathbf{M}_{02} - \mathbf{M}_{12}\mathbf{M}_{11}^{-1}\mathbf{M}_{12},$$

which is the Schur complement of the upper-left block $\begin{bmatrix} M_{00} & \\ & \mathbf{M}_{11}\end{bmatrix}$. Using formulas for the inverse of a block matrix in terms of its constituent blocks, it can be shown that

$$\Delta\mathbf{w} = \mathbf{M}_S^{-1}\left(\mathbf{r}_2 - \frac{r_0}{M_{00}}\mathbf{M}_{02}^T - \mathbf{M}_{12}\mathbf{M}_{11}^{-1}\mathbf{r}_1\right), \tag{3.5.24a}$$

$$\Delta y_0 = \frac{r_0 - \mathbf{M}_{02}\Delta\mathbf{w}}{M_{00}}, \tag{3.5.24b}$$

$$\Delta\mathbf{v} = \mathbf{M}_{11}^{-1}(\mathbf{r}_1 - \mathbf{M}_{12}\Delta\mathbf{w}). \tag{3.5.24c}$$

Equations (3.5.24) require the inversion of the dense $N \times N$ matrix $\mathbf{M}_S$ and the diagonal $N \times N$ matrix $\mathbf{M}_{11}$. The gain in efficiency over the direct inversion of the full $(2N+1) \times (2N+1)$ matrix $\mathbf{M}$ can be significant since the inversion of a matrix without exploiting structure requires $\mathcal{O}(N^3)$ arithmetic operations.

The dual search direction $\Delta\mathbf{X}$ is computed using (3.5.22) once $\Delta\mathbf{y}$ has been determined from (3.5.24). Because of numerical errors, the new dual solution $\mathbf{X} + \tau_2\Delta\mathbf{X}$, where $\tau_2$ is a positive step size, may not satisfy the constraints in (3.5.14) exactly. Assuming that the old solution $\mathbf{X}$ does satisfy the constraints, this implies that the search direction $\Delta\mathbf{X}$ violates the conditions

$$\operatorname{tr}(\Delta\mathbf{Z}) - \mathbf{e}^T\Delta\mathbf{s} = 0, \tag{3.5.25a}$$

$$\operatorname{diag}(\Delta\mathbf{Z}) - \Delta\mathbf{s} - \Delta\mathbf{t} = \mathbf{0}, \tag{3.5.25b}$$

$$\Delta\mathbf{t} + \Delta\mathbf{u} = \mathbf{0}, \tag{3.5.25c}$$

which are required for $\mathbf{X} + \tau_2\Delta\mathbf{X}$ to remain feasible. Condition (3.5.25a) is derived from $\mathbf{e}^T\mathbf{t} = K$ by requiring that $\mathbf{e}^T\Delta\mathbf{t} = 0$ and combining this with (3.5.25b). To ensure the feasibility of the new dual solution, we propose correcting the nominal direction $\Delta\mathbf{X}$ so that the corrected direction $\Delta\mathbf{X}'$ does satisfy (3.5.25). The size of the correction is to be minimized in the least-squares sense to perturb $\Delta\mathbf{X}$ as little as possible. This leads to

$$\begin{aligned} \min_{\Delta\mathbf{X}'} \quad & \left\|\operatorname{diag}(\Delta\mathbf{Z}') - \operatorname{diag}(\Delta\mathbf{Z})\right\|_2^2 + \left\|\Delta\mathbf{s}' - \Delta\mathbf{s}\right\|_2^2 + \left\|\Delta\mathbf{t}' - \Delta\mathbf{t}\right\|_2^2 + \left\|\Delta\mathbf{u}' - \Delta\mathbf{u}\right\|_2^2 \\ \text{s.t.} \quad & \underbrace{\begin{bmatrix} \mathbf{e}^T & -\mathbf{e}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{I} \\ \mathbf{I} & -\mathbf{I} & -\mathbf{I} & \mathbf{0} \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} \operatorname{diag}(\Delta\mathbf{Z}') \\ \Delta\mathbf{s}' \\ \Delta\mathbf{t}' \\ \Delta\mathbf{u}' \end{bmatrix} = \mathbf{0}, \end{aligned} \tag{3.5.26}$$

where the matrix $\mathbf{A}$ is used to enforce the conditions in (3.5.25). The solution to (3.5.26) is

$$\begin{bmatrix} \operatorname{diag}(\Delta\mathbf{Z}') \\ \Delta\mathbf{s}' \\ \Delta\mathbf{t}' \\ \Delta\mathbf{u}' \end{bmatrix} = \left(\mathbf{I} - \mathbf{A}^T\left(\mathbf{A}\mathbf{A}^T\right)^{-1}\mathbf{A}\right) \begin{bmatrix} \operatorname{diag}(\Delta\mathbf{Z}) \\ \Delta\mathbf{s} \\ \Delta\mathbf{t} \\ \Delta\mathbf{u} \end{bmatrix} \tag{3.5.27}$$

using the pseudo-inverse of $\mathbf{A}$. The off-diagonal entries of $\Delta\mathbf{Z}'$ are unchanged from those of $\Delta\mathbf{Z}$. Equation (3.5.27) can be simplified to the following:

$$\begin{aligned}
\operatorname{diag}(\Delta\mathbf{Z}') &= \frac{1}{5}\left(3\operatorname{diag}(\Delta\mathbf{Z}) + 2\Delta\mathbf{s} + \Delta\mathbf{t} - \Delta\mathbf{u}\right) - S\mathbf{e},\\
\Delta\mathbf{s}' &= \frac{1}{5}\left(2\operatorname{diag}(\Delta\mathbf{Z}) + 3\Delta\mathbf{s} - \Delta\mathbf{t} + \Delta\mathbf{u}\right) + S\mathbf{e},\\
\Delta\mathbf{t}' &= \operatorname{diag}(\Delta\mathbf{Z}') - \Delta\mathbf{s}',\\
\Delta\mathbf{u}' &= -\Delta\mathbf{t}',
\end{aligned}$$

with

$$S = \frac{1}{10N}\left(\operatorname{tr}(\Delta\mathbf{Z}) - \mathbf{e}^T\Delta\mathbf{s} + 2\mathbf{e}^T\Delta\mathbf{t} - 2\mathbf{e}^T\Delta\mathbf{u}\right).$$

Thus the corrected search direction $\Delta\mathbf{X}'$ can be computed with no matrix inversions.

### 3.5.6 Plane search

We now discuss the determination of the sizes of the steps to be taken in the primal and dual search directions. Ideally, the primal and dual step sizes are chosen independently so as to maximize the decrease in the potential function (3.5.15). The resulting two-variable problem is referred to as a plane search and is discussed in detail in [103]. In this subsection, we focus on a simplification in which the ratio between the primal and dual step sizes is fixed, thereby restricting the plane search to a more efficient one-dimensional search. We have observed that the simplified search performs just as well as the full plane search for many instances of problems (3.5.10) and (3.5.14).

Let $\tau_1$ and $\tau_2$ represent the primal and dual step sizes. In our simplified method, the ratio between $\tau_1$ and $\tau_2$ is determined based on the region of permissible step sizes. This depends in turn on the potential function as we now illustrate. Following the same notation as in Section 3.5.5 and defining

$$\Delta\mathbf{S} = -\sum_{n=0}^{2N}\Delta y_n\mathbf{A}_n,$$

the change in the potential function due to steps $\tau_1\Delta\mathbf{y}$ and $\tau_2\Delta\mathbf{X}$ in the primal and dual spaces is given by

$$\begin{aligned}\Delta\phi(\tau_1,\tau_2) &\equiv \phi(\mathbf{y}+\tau_1\Delta\mathbf{y},\mathbf{X}+\tau_2\Delta\mathbf{X})-\phi(\mathbf{y},\mathbf{X})\\ &= (4N+2\nu\sqrt{N})\ln\left(1+\tau_1\frac{\overline{\mathbf{Q}}\bullet\mathbf{Z}}{\eta}-\tau_2\frac{K\Delta y_0+\mathbf{e}^T\Delta\mathbf{v}}{\eta}\right)\\ &\qquad -\ln\det\left(\mathbf{I}+\tau_1\mathbf{S}^{-1}\Delta\mathbf{S}\right)-\ln\det\left(\mathbf{I}+\tau_2\mathbf{X}^{-1}\Delta\mathbf{X}\right)\end{aligned}\tag{3.5.28}$$

after some straightforward combining of terms. By expressing the determinants as products of eigenvalues and using the fact that the addition of **I** to a matrix adds 1 to its eigenvalues, (3.5.28) can be rewritten as

$$\begin{aligned}\Delta\phi(\tau_1,\tau_2) &= (4N+2\nu\sqrt{N})\ln\left(1+\tau_1\frac{\overline{\mathbf{Q}}\bullet\mathbf{Z}}{\eta}-\tau_2\frac{K\Delta y_0+\mathbf{e}^T\Delta\mathbf{v}}{\eta}\right)\\ &\qquad -\sum_{n=1}^{4N}\ln\left(1+\tau_1\lambda_n\left(\mathbf{S}^{-1}\Delta\mathbf{S}\right)\right)-\sum_{n=1}^{4N}\ln\left(1+\tau_2\lambda_n\left(\mathbf{X}^{-1}\Delta\mathbf{X}\right)\right).\end{aligned}\tag{3.5.29}$$

The change in the potential increases to $+\infty$ if any of the arguments of the ln functions in the second line of (3.5.29) approaches zero. This corresponds to a loss of positive definiteness in either $\mathbf{S}+\tau_1\Delta\mathbf{S}$ or $\mathbf{X}+\tau_2\Delta\mathbf{X}$. To avoid this situation, the step sizes must be bounded as follows:

$$\tau_{1\,\min} = -\frac{1}{\max\left\{\lambda_n\left(\mathbf{S}^{-1}\Delta\mathbf{S}\right)\right\}} < \tau_1 < \tau_{1\,\max} = -\frac{1}{\min\left\{\lambda_n\left(\mathbf{S}^{-1}\Delta\mathbf{S}\right)\right\}},\tag{3.5.30a}$$

$$\tau_{2\,\min} = -\frac{1}{\max\left\{\lambda_n\left(\mathbf{X}^{-1}\Delta\mathbf{X}\right)\right\}} < \tau_2 < \tau_{2\,\max} = -\frac{1}{\min\left\{\lambda_n\left(\mathbf{X}^{-1}\Delta\mathbf{X}\right)\right\}}.\tag{3.5.30b}$$

Equations (3.5.30) define the region of permissible step sizes.

We now restrict $(\tau_1,\tau_2)$ to be a positive multiple of the maximum step sizes $(\tau_{1\,\max},\tau_{2\,\max})$, i.e.,

$$(\tau_1,\tau_2)=s(\tau_{1\,\max},\tau_{2\,\max}),\quad 0<s<1.$$

Since the search directions $\Delta\mathbf{y}$ and $\Delta\mathbf{X}$ are designed to be directions of descent, it is usually sufficient to consider only positive $s$ to obtain a decrease in the potential. To ensure that the decrease is sufficiently large, we use the following Armijo condition to determine $s$:

$$\Delta\phi(s\tau_{1\,\max},s\tau_{2\,\max})\le\sigma s\left(\tau_{1\,\max}\frac{\partial\Delta\phi}{\partial\tau_1}(0,0)+\tau_{2\,\max}\frac{\partial\Delta\phi}{\partial\tau_2}(0,0)\right).\tag{3.5.31}$$

Condition (3.5.31) requires that the decrease in $\phi$ be at least as large in magnitude as the decrease predicted by a linear approximation to $\phi$ at $\tau_1 = \tau_2 = 0$, scaled by a constant $\sigma$ between 0 and 1 (see [98] for background on the Armijo rule). To begin, $s$ is set to an initial value $s_0$ and condition (3.5.31) is evaluated. If (3.5.31) is not satisfied with $s = s_0$, $s$ is replaced by $\theta s$ with $0 < \theta < 1$ and (3.5.31) is re-evaluated. Once a value of $s$ is found that satisfies (3.5.31), the step sizes $\tau_1$ and $\tau_2$ are determined. The process is guaranteed to terminate as long as the partial derivatives in (3.5.31) are negative, i.e., as long as $\Delta\mathbf{y}$ and $\Delta\mathbf{X}$ are directions of descent. If $\partial\Delta\phi/\partial\tau_1$ is positive, the method can still be used with $\tau_{1\,\max}$ replaced by $\tau_{1\,\min}$ (i.e., by reversing directions), and similarly for $\tau_2$. In our numerical experiments, we have used the values $\sigma = 0.1$, $s_0 = 0.85$, and $\theta = 0.9$.

We have found that for most instances of (3.5.10) and (3.5.14), the simplified search yields step sizes close to those resulting from a full plane search, but with significantly lower complexity. On occasion however, the simplified search may not produce sufficient decreases in the potential. This situation can be remedied by re-instituting a full plane search once the number of iterations in the primal-dual algorithm has become large (e.g. over 100).

## 3.6 Numerical comparison of linear and diagonal relaxations

In Sections 3.3 and 3.4, we analyzed the quality of the lower bounds on the optimal value of problem (2.0.1) resulting from linear and diagonal relaxations. In particular, it was observed in Section 3.4.2 that neither relaxation dominates the other over all possible instances of (2.0.1). However, for many classes of instances, the bounds provided by diagonal relaxations tend to be stronger than those from linear relaxations. In this section, we present results from numerical experiments that compare the two types of relaxations and largely support the conclusion that diagonal relaxations are superior.

The experiments in this section also serve to further elucidate the approximation properties of the diagonal relaxation. Specifically, it will be seen that the distribution of eigenvalues of the matrix $\mathbf{Q}$ can play an important role in addition to properties such as the condition number and diagonal dominance that were identified in Section 3.4. To explore these effects, a large number of instances of (2.0.1) were created by randomly selecting values for $\mathbf{Q}$ and $\mathbf{c}$ according to different methods and probability distributions. The corresponding ellipsoids $\mathcal{E}_{\mathbf{Q}}$ have different properties as a result and the quality of approximation for the diagonal relaxation is shown to vary accordingly.

We use four different methods to generate instances. For all of the methods, the number of dimensions $N$ is varied between 10 and 150 and the parameter $\gamma$ is normalized to 1. The linear relaxation of each instance, and more specifically the dual form (3.3.9), is solved using the function `fmincon` in MATLAB. We use the customized solver described in Section 3.5 for the diagonal relaxation; a general-purpose solver such as SDPT3 [99, 100] or SeDuMi [101] can also be used to solve problem (3.4.3). In addition, a feasible solution for each instance is obtained using the successive thinning algorithm of Section 2.3. The ratio of the optimal value of each relaxation to the objective value of the feasible solution is used to assess the quality of the relaxation. Note that we are now defining approximation ratios in terms of the number of non-zero coefficients and not the number of zero coefficients as in Sections 3.4.4–3.4.6. We will use $R_\ell$ and $R_d$ to denote the ratios corresponding to linear and diagonal relaxations respectively. Since any feasible solution provides an upper bound on the optimal cost of (2.0.1), $R_\ell$ and $R_d$ are lower bounds on the true approximation ratios, which are difficult to compute given the large number of instances.

In the first three methods, the eigenvalues and eigenvectors of $\mathbf{Q}$ are generated separately. The eigenvectors are chosen together as an orthonormal set oriented randomly and uniformly over the unit sphere in $N$ dimensions. The eigenvalues are specified both in terms of the condition number $\kappa(\mathbf{Q})$, which determines their range, as well as their distribution within the range. For each value of $N$, $\kappa(\mathbf{Q})$ is set in turn to $\sqrt{N}$, $N$, $10N$, and $100N$. The methods differ in the choice of eigenvalue distribution. Once $\mathbf{Q}$ is determined, each component $c_n$ of the ellipsoid center is drawn uniformly from the interval $\left[-\sqrt{(\mathbf{Q}^{-1})_{nn}}, \sqrt{(\mathbf{Q}^{-1})_{nn}}\right]$ to ensure that (2.3.3) is satisfied for all $n$. This choice of $\mathbf{c}$ is in keeping with our assumption that a feasible solution exists whenever a single coefficient is constrained to zero. For each pair of $N$ and $\kappa(\mathbf{Q})$, 1000 instances are created according to the general procedure described above.

In the first method, the eigenvalues of $\mathbf{Q}$ are drawn from a distribution $f_1(\lambda) \propto 1/\lambda$ and then scaled to match the specified condition number. This distribution corresponds to $\log \lambda$ being uniformly distributed. One motivation for considering power-law eigenvalue distributions stems from the typical channel frequency responses encountered in wireline communications. A second motivation for choosing a $1/\lambda$ distribution is due to its invariance under matrix inversion (up to a possible overall scaling). While no single eigenvalue distribution can be representative of all positive definite matrices, the inverse of any positive definite matrix is also positive definite and hence the distribution $f_1(\lambda)$ can be regarded as unbiased in this sense.

In Fig. 3-7 we plot the ratios $R_\ell$ and $R_d$ as functions of $N$ and $\kappa(\mathbf{Q})$, averaged over the 1000 instances corresponding to each $(N, \kappa(\mathbf{Q}))$ pair. The linear relaxation ratio $R_\ell$ does not vary much with $N$ or $\kappa(\mathbf{Q})$ except for a slight decrease at low $N$. In contrast, $R_d$ is markedly higher for lower $\kappa(\mathbf{Q})$. This dependence agrees qualitatively with Theorem 2 and its geometric interpretation in terms of ellipsoid sphericality. For $\kappa(\mathbf{Q}) = \sqrt{N}$, approximation ratios between 0.81 and 0.92 imply that the corresponding lower bounds are quite strong. Moreover, $R_d$ also improves with increasing $N$ so that even for $\kappa(\mathbf{Q}) = 100N$ the diagonal relaxation outperforms the linear relaxation for $N \geq 20$, with the difference being substantial at large $N$.

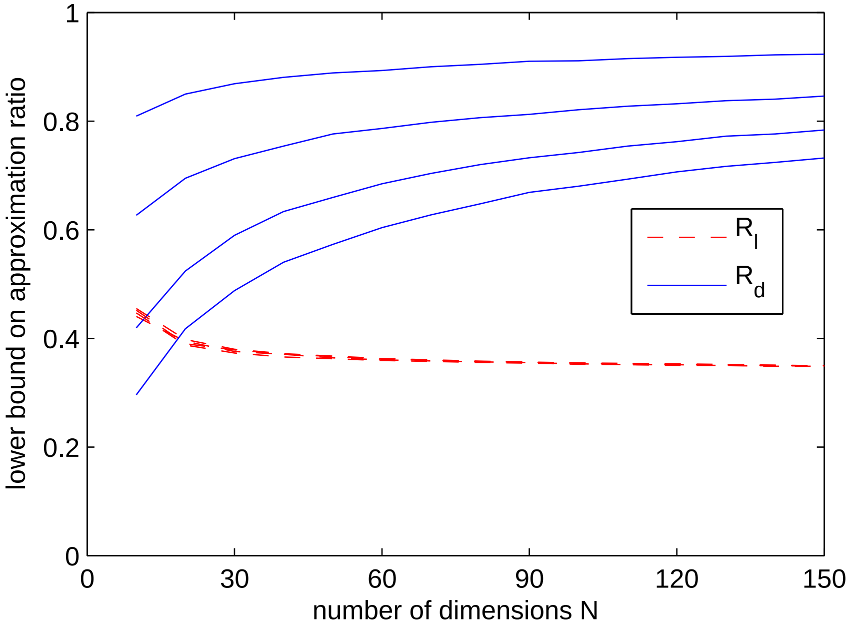


Figure 3-7: Average values of $R_\ell$ and $R_d$ for a $1/\lambda$ eigenvalue distribution. Within each set of curves, $\kappa(\mathbf{Q}) = \sqrt{N}, N, 10N, 100N$ from top to bottom.

To understand the spreads around the mean values plotted in Fig. 3-7, in Fig. 3-8 we show histograms of the optimal values of the linear and diagonal relaxations and the objective values of the feasible solutions for the 1000 instances with $N = \kappa(\mathbf{Q}) = 100$. There is a wide separation between the histograms corresponding to the linear and diagonal relaxations. The spread in the histograms around the average values is similar for other values of $N$ and $\kappa(\mathbf{Q})$.

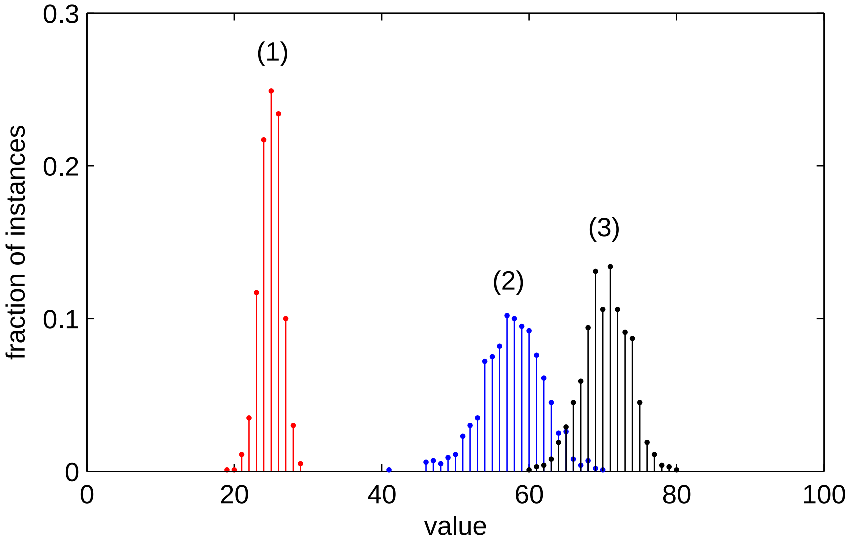


Figure 3-8: Histograms of the optimal values of linear relaxations (1), the optimal values of diagonal relaxations (2), and the objective values of feasible solutions (3) for a $1/\lambda$ eigenvalue distribution and $N = \kappa(\mathbf{Q}) = 100$.

In the second method, the eigenvalues of $\mathbf{Q}$ are drawn from a uniform distribution and then scaled according to the condition number as before. Fig. 3-9 shows the average values of $R_\ell$ and $R_d$ as a function of $N$ and $\kappa(\mathbf{Q})$ for the second method. The behavior of $R_\ell$ is largely unchanged. Each $R_d$ curve however is lower than its counterpart in Fig. 3-7 and the decrease in $R_d$ with increasing condition number is more pronounced. The linear relaxation is now preferable to the diagonal relaxation when $\kappa(\mathbf{Q})$ is significantly greater than $N$. The difference between Figs. 3-7 and 3-9 can be explained by referring to the discussion in Section 3.4.4 following Theorem 2. There it was argued that the diagonal relaxation tends to yield a better approximation when most of the eigenvalues are small and of comparable size. This situation is better represented by the distribution $f_1(\lambda) \propto 1/\lambda$ than by a uniform distribution, given the same condition number in both cases. Figs. 3-7 and 3-9 demonstrate numerically that eigenvalue distributions that are more heavily weighted toward small values are preferred. There does not appear to be a similar preference in the case of linear relaxations.

Histograms corresponding to those in Fig. 3-8 are shown for the uniform eigenvalue distribution in Fig. 3-10. It is seen that the spreads in the histograms are similar.

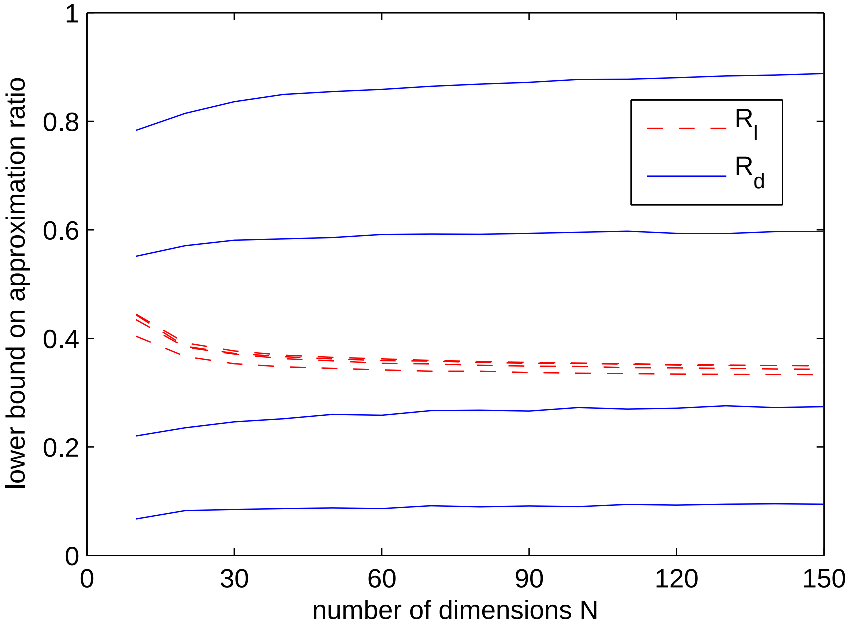


Figure 3-9: Average values of $R_\ell$ and $R_d$ for a uniform eigenvalue distribution. Within each set of curves, $\kappa(\mathbf{Q}) = \sqrt{N}, N, 10N, 100N$ from top to bottom.

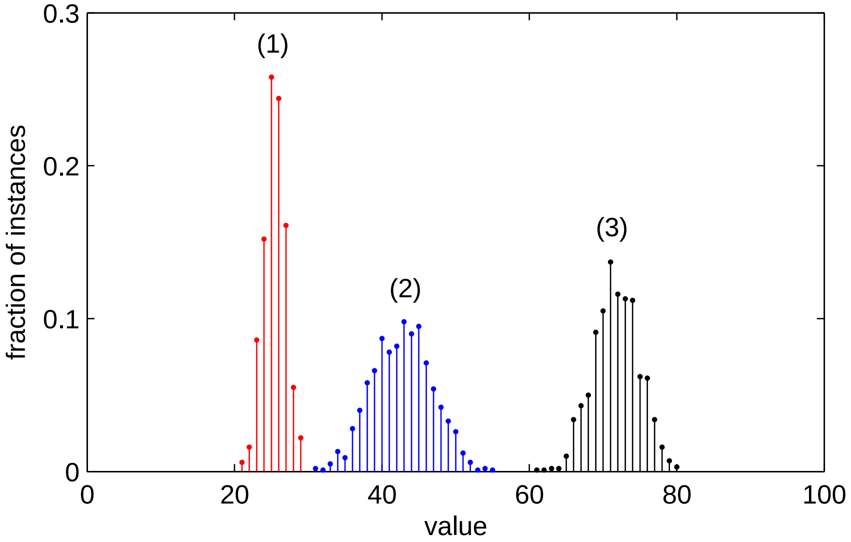


Figure 3-10: Histograms of the optimal values of linear relaxations (1), the optimal values of diagonal relaxations (2), and the objective values of feasible solutions (3) for a uniform eigenvalue distribution and $N = \kappa(\mathbf{Q}) = 100$.

In the third method, the eigenvalues of $\mathbf{Q}^{-1}$ are drawn from a uniform distribution. It is straightforward to show that the eigenvalues of $\mathbf{Q}$, which are the reciprocals of the eigenvalues of $\mathbf{Q}^{-1}$, are distributed according to $f_2(\lambda) \propto 1/\lambda^2$. Fig. 3-11 plots the average values of $R_\ell$ and $R_d$ resulting from the third method. The curves for $R_d$ are higher than in either Fig. 3-7 or 3-9 and the dependence on $\kappa(\mathbf{Q})$ is reduced. This is not surprising in light of the previous discussion since a $1/\lambda^2$ distribution is more concentrated toward the lower end of the eigenvalue spectrum than either a $1/\lambda$ or uniform distribution. On the other hand, the dependence of $R_\ell$ on the eigenvalue distribution is hardly discernible.

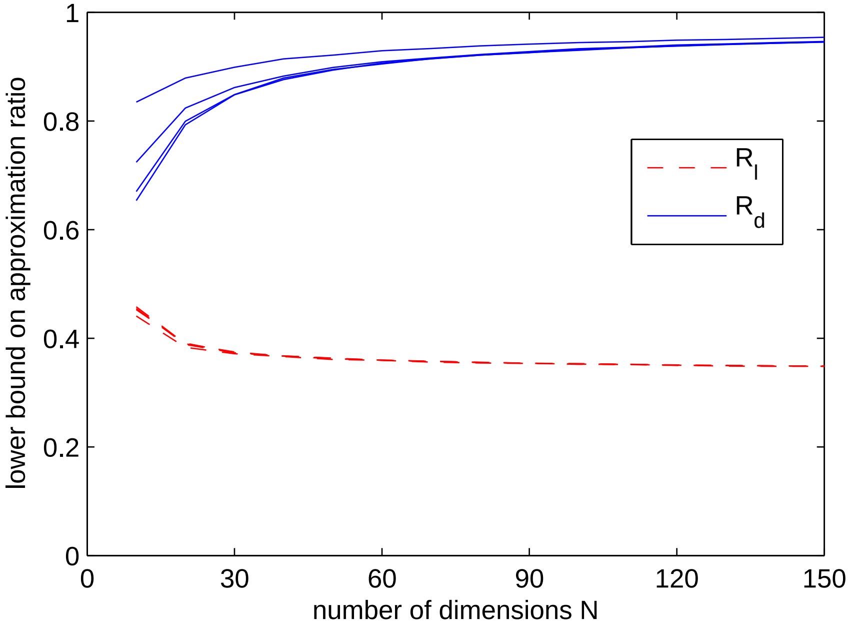


Figure 3-11: Average values of $R_\ell$ and $R_d$ for a $1/\lambda^2$ eigenvalue distribution. Within each set of curves, $\kappa(\mathbf{Q}) = \sqrt{N}, N, 10N, 100N$ from top to bottom.

In the fourth method, $\mathbf{Q}$ is chosen to correspond to an exponentially decaying autocorrelation function. Specifically,

$$Q_{mn} = \rho^{|m-n|}, \tag{3.6.1}$$

where the decay ratio $\rho$ is varied between 0.05 and 0.99. The vector $\mathbf{c}$ is generated as before based on the diagonal entries of $\mathbf{Q}^{-1}$. It is sufficient to consider only positive values of $\rho$. Changing $\rho$ to $-\rho$ can be shown to be equivalent to multiplying $\mathbf{b} - \mathbf{c}$ component-wise by the vector $\begin{bmatrix} +1 & -1 & +1 & -1 & \dots \end{bmatrix}$. The zero-norm $\|\mathbf{b}\|_0$ remains the same under sign changes and the distribution for $\mathbf{c}$, which is symmetric about the origin, is also unaffected.

For $\rho \leq 1/3$, the matrix $\mathbf{Q}$ specified by (3.6.1) is diagonally dominant in the sense assumed in Theorem 3. To verify this, we use (3.6.1) to rewrite the left-hand side of the criterion for diagonal dominance as follows:

$$\max_{m} \sum_{n \neq m} \frac{|Q_{mn}|}{\sqrt{Q_{mm} Q_{nn}}} = \max_{m} \sum_{n \neq m} \rho^{|m-n|}.$$

The maximizing index $m$ corresponds to the central row if $N$ is odd, i.e., $m = (N+1)/2$, or to either of the two central rows if $N$ is even, i.e., $m = N/2$ or $m = N/2 + 1$. In all three cases, the maximum sum is slightly less than $2\rho/(1-\rho)$ (more precisely, the sum approaches $2\rho/(1-\rho)$ exponentially from below as a function of $N$). For $\rho \leq 1/3$, $2\rho/(1-\rho) \leq 1$, and hence the assumption of Theorem 3 is satisfied. On the other hand, $\mathbf{Q}$ is not diagonally dominant for large values of $\rho$.

For each pair of $N$ and $\rho$, 1000 instances are generated and evaluated as before. Fig. 3-12 shows the dependence of the average values of $R_\ell$ and $R_d$ on $N$ for selected values of $\rho$. As with the condition number $\kappa(\mathbf{Q})$ in Figs. 3-7, 3-9, and 3-11, the parameter $\rho$ does not appear to have much effect on $R_\ell$. Furthermore, while it is expected from the analysis in Section 3.4.5 that the diagonal relaxation yields a close approximation for $\rho = 0.1$, it is somewhat surprising that the performance does not degrade by much even for $\rho$ close to 1.

The results in this section indicate that diagonal relaxations yield better lower bounds than linear relaxations in many instances. This can be true even when the condition number $\kappa(\mathbf{Q})$ or the decay ratio $\rho$ is high, whereas the analysis in Sections 3.4.4–3.4.6 tends to be more pessimistic. The experiments also confirm the dependence of the diagonal relaxation on the conditioning and diagonal dominance of $\mathbf{Q}$, and reveal an additional dependence on the eigenvalue distribution. As noted in Section 3.4.4, the preference for distributions in which most eigenvalues are small has a geometric basis, but a more rigorous explanation is currently lacking.

## 3.7 Description of branch-and-bound algorithm

In Sections 3.2–3.6, we focused on developing lower bounds on the optimal value of problem (2.0.1) and its subproblems. These bounds are now incorporated in a branch-and-bound algorithm for solving (2.0.1) exactly. This section describes our algorithm in greater detail,

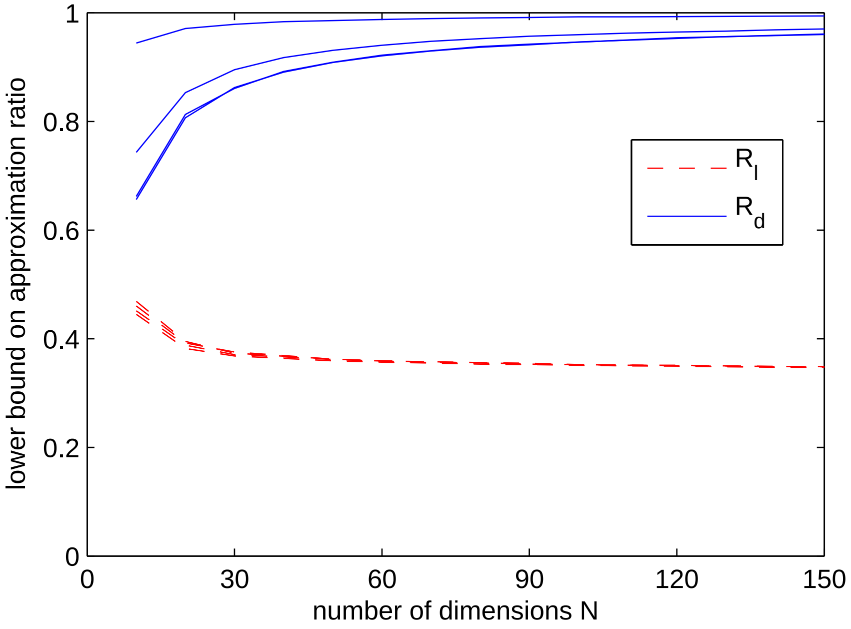


Figure 3-12: Average values of $R_\ell$ and $R_d$ for exponentially decaying $\mathbf{Q}$ matrices. Within each set of curves, $\rho = 0.1, 0.5, 0.9, 0.99$ from top to bottom.

building upon the overview given in Section 3.1.

The algorithm processes a series of subproblems beginning with the root problem (2.0.1). The subproblems are organized into a tree as depicted in Fig. 3-1. For each subproblem, a lower bound on its optimal value, denoted by $LB$ in this section, is inherited from its parent (or initialized to zero in the case of the root problem) and then updated during processing. The algorithm maintains a list of open subproblems whose lower bounds indicate that they have the potential to improve upon the incumbent solution. Subproblems are added to the list by the branching process and are removed as they are visited or pruned. The algorithm terminates when the list is empty.

A summary of the algorithm is given under Algorithm 2. The processing steps for each subproblem are numbered and described in more detail below. The indicator variable $i_{\text{last}}$ refers to the last indicator variable that was fixed in creating a subproblem from its parent.

1. *Select subproblem from list:* We choose the open subproblem for which the lower bound inherited from its parent is the smallest. This choice is motivated by the desire to increase as quickly as possible the global lower bound, which is the minimum of the lower bounds for currently open subproblems. Thus if the algorithm is terminated early after a fixed number of iterations, the bound on the deviation of the incum-

**Algorithm 2** Branch-and-bound for problem (2.0.1)

```
Input: Parameters $\mathbf{Q}$, $\mathbf{c}$, $\gamma$
Output: Optimal solution $\mathbf{b}_I$ to (2.0.1)
  Initialize: Place root problem in list with $LB = 0$ and $i_{\text{last}} = 0$. Incumbent solution
  $\mathbf{b}_I = \mathbf{c}$ with cost $\|\mathbf{b}_I\|_0 = \|\mathbf{c}\|_0$.
  while list not empty do
    1) Select subproblem with smallest $LB$ and remove from list.
    2) Subproblem parameters $\mathbf{Q}_{\text{eff}}$, $\mathbf{c}_{\text{eff}}$, $\gamma_{\text{eff}}$, $\mathbf{f}_{\text{eff}}$ given by (2.3.2), (A.3.6).
    if $i_{\text{last}} = 0$ then
      3) Identify coefficients in $\mathcal{F}$ for which a zero value is no longer feasible using (2.3.3).
      Update $\mathcal{U}$, $\mathcal{F}$, $\mathbf{Q}_{\text{eff}}$, $\mathbf{c}_{\text{eff}}$, $\mathbf{f}_{\text{eff}}$ if necessary.
      if $|\mathcal{U}| \geq \|\mathbf{b}_I\|_0$ then
        Go to step 1.
    if $LB < |\mathcal{U}| + 2$ then
      4) Check for solutions with $\|\mathbf{b}_{\mathcal{F}}\|_0 = 0$, $\|\mathbf{b}_{\mathcal{F}}\|_0 = 1$ (see Section 3.2).
      if subproblem solved and $|\mathcal{U}| + \|\mathbf{b}_{\mathcal{F}}\|_0 < \|\mathbf{b}_I\|_0$ then
        Update $\mathbf{b}_I$ and prune list. Go to step 1.
      else
        $LB \leftarrow |\mathcal{U}| + 2$.
        if $LB \geq \|\mathbf{b}_I\|_0$ then
          Go to step 1.
    5) Generate feasible solution $\mathbf{b}_{\mathcal{F}}$ using successive thinning (see Section 2.3) or with
    $\|\mathbf{b}_{\mathcal{F}}\|_0 = |\mathcal{F}| - 1$.
    if $|\mathcal{U}| + \|\mathbf{b}_{\mathcal{F}}\|_0 < \|\mathbf{b}_I\|_0$ then
      Update $\mathbf{b}_I$ and prune list (possibly including current subproblem).
    if $i_{\text{last}} = 0$ and $|\mathcal{F}| \geq N_{\text{min}}$ then
      6) Solve linear or diagonal relaxation (see Section 3.5) and update $LB$.
      if $LB \geq \|\mathbf{b}_I\|_0$ then
        Go to step 1.
    7) Determine $m$ from (3.7.1). Create two new subproblems by fixing $i_m = 0$ and $i_m = 1$
    and add to list. Go to step 1.
```

bent solution from optimality is as tight as possible. Furthermore, this selection rule increases efficiency by deferring on subproblems that are more likely to be pruned without being visited. Whenever the incumbent solution is improved, the subproblems with the highest $LB$ values are pruned first, so the algorithm should concentrate on processing the subproblems with the lowest bounds.

2. *Subproblem parameters:* As discussed in Section 3.1, each subproblem is defined by index sets $(\mathcal{Z}, \mathcal{U}, \mathcal{F})$ and is equivalent to an $|\mathcal{F}|$-dimensional instance of the root problem (2.0.1) with parameters given by (2.3.2) and (A.3.6). The parameter values are computed only when needed as it is sometimes possible to avoid the computation entirely.

3. *Identify coefficients for which a zero value is no longer feasible:* The algorithm checks whether a zero value is still feasible for all coefficients in $\mathcal{F}$ as described in Section 3.2. This step is not necessary for a subproblem with $i_{\text{last}} = 1$ since the set of feasible $\mathbf{b}$ is unchanged relative to the parent subproblem. Indicator variables and subproblem parameters are updated based on the results of these tests. If the new value of $|\mathcal{U}|$ equals or exceeds the cost of the incumbent solution, the current subproblem can be pruned. Otherwise, after this step it is known that a feasible solution with $|\mathcal{U}|+|\mathcal{F}|-1$ non-zero coefficients exists because a zero value is feasible for every coefficient in $\mathcal{F}$.

4. *Check for solutions with $\|\mathbf{b}_{\mathcal{F}}\|_0 = 0$, $\|\mathbf{b}_{\mathcal{F}}\|_0 = 1$:* We determine whether there are any feasible solutions with $\|\mathbf{b}_{\mathcal{F}}\|_0 = 0$ (i.e., $\mathbf{b}_{\mathcal{F}} = \mathbf{0}$) or $\|\mathbf{b}_{\mathcal{F}}\|_0 = 1$ as described in Section 3.2. This step can result in an optimal solution to the current subproblem and thus avoid further branching, particularly if $|\mathcal{F}|$ is small. If $\mathcal{F}$ consists of a single index, then $b_{\mathcal{F}} = 0$ is feasible by the definition of $\mathcal{F}$. If $|\mathcal{F}| = 2$, there exist solutions with $\|\mathbf{b}_{\mathcal{F}}\|_0 = 1$ and it suffices to check whether $\mathbf{b}_{\mathcal{F}} = \mathbf{0}$ is also feasible. Similarly if $|\mathcal{F}| = 3$, solutions with $\|\mathbf{b}_{\mathcal{F}}\|_0 = 2$ must exist and we search for solutions with fewer than two non-zero components. To decide which one of the three components in $\mathcal{F}$ to make non-zero (in the case $\|\mathbf{b}_{\mathcal{F}}\|_0 = 1$) or zero (in the case $\|\mathbf{b}_{\mathcal{F}}\|_0 = 2$), we look to maximize the margin in the overall quadratic constraint (2.1.1), which corresponds to maximizing the margin in (3.2.1) in the first case and (2.3.3) in the second.

   If the current subproblem is solved in this step and the solution has a lower cost than the incumbent solution, we update the incumbent solution and prune any open subproblems with a lower bound greater than or equal to the new incumbent cost. If the current subproblem is not solved, we can conclude that its optimal value is no less than $|\mathcal{U}|+2$. This may represent an improvement upon the the lower bound inherited from the parent subproblem and result in pruning of the current subproblem. On the other hand, if the inherited lower bound already exceeds $|\mathcal{U}|+2$, the current step can be skipped entirely. Therefore the execution of this step is conditioned on the value of $LB$. In either case, at the end of this step we have a lower bound for the current subproblem that is at least $|\mathcal{U}|+2$.

5. *Generate a feasible solution:* In this step, we obtain a feasible solution to the current subproblem, which may result in an update to the incumbent solution and the pruning of subproblems as described in Step 4. We consider two variants of the algorithm. In the first variant, we use the successive thinning algorithm of Section 2.3 to obtain a feasible solution only for the root problem. For all other subproblems, we rely only on the knowledge that a solution with $|\mathcal{U}|+|\mathcal{F}|-1$ non-zero coefficients exists by the definition of $\mathcal{F}$. The single zero-valued coefficient in $\mathcal{F}$ in this solution can be chosen to maximize the margin in (2.3.3) as in Step 4. In the second variant, successive thinning is used to generate feasible solutions for all subproblems. The incumbent solution tends to improve more quickly with the second variant at the cost of increased computation in every iteration. In Section 4.1, we explore the performance of both variants on randomly generated instances of (2.0.1).

6. *Solve relaxation:* To improve upon the value of $LB$, we solve either a diagonal or a linear relaxation of the current subproblem. If the new lower bound is equal to or exceeds the cost of the incumbent solution, the current subproblem is pruned. For diagonal relaxations, we use an efficient custom solver described in Section 3.5. For linear relaxations, the MATLAB function `fmincon` is used to solve the dual (3.3.9). Although the comparisons in Section 3.6 suggest that diagonal relaxation yields better lower bounds than linear relaxation in most cases, we use and compare both types of relaxations in the numerical experiments in Section 4.1.

   Solving relaxations is by far the most computationally intensive step in the algorithm. The increased computation is justified if a sufficiently large number of subproblems can be eliminated as a result of stronger lower bounds. We have found that it is not worthwhile to solve a relaxation for every subproblem. In particular, relaxations of subproblems for which $i_{\text{last}} = 1$ rarely lead to pruning, so we skip the relaxation step for these subproblems. In addition, small subproblems can often be solved more efficiently by relying only on the low-complexity lower bounds of Section 3.2 and the branch-and-bound process. For this reason, we only solve relaxations of subproblems for which the dimension $|\mathcal{F}|$ is greater than or equal to a parameter $N_{\text{min}}$. In Section 4.1, we investigate how the efficiency of the algorithm depends on the value of $N_{\text{min}}$.

7. *Create new subproblems:* At this point, the current subproblem has not been solved, nor has the value of $LB$ increased enough to eliminate it from consideration. The final step is to create two new subproblems by fixing an indicator variable in the subset $\mathcal{F}$ to 0 or 1. We choose the index $m$ in $\mathcal{F}$ that minimizes the margin in (2.3.3), i.e.,

$$m = \arg\min_{n \in \mathcal{F}} \quad \gamma - \frac{c_n^2}{\left(\mathbf{Q}^{-1}\right)_{nn}}. \tag{3.7.1}$$

We have observed that this choice of index for branching tends to reduce the number of subproblems that are visited. Equation (3.7.1) implies that when the coefficient $b_m$ is constrained to a zero value, the resulting subproblem, while still feasible, tends to be tightly constrained. Therefore the subtree created under the current subproblem is unbalanced with many more nodes under the $i_m = 1$ branch than under the $i_m = 0$ branch. Generally speaking, the higher that these asymmetric branchings occur in the tree, the greater the reduction in the number of subproblems. As an extreme example, if for the root problem we choose an index for branching such that there are very few feasible subproblems under one of the branches, then the number of subproblems is almost halved. This intuition supports the branching rule in (3.7.1).

# Chapter 4

# Sparse filter design under a quadratic constraint: Numerical experiments and design examples

In this chapter, the successive thinning algorithm of Section 2.3 and the branch-and-bound algorithm of Chapter 3 are applied to a range of examples. In Section 4.1, we investigate the properties of the two algorithms using randomly generated problem instances. For the successive thinning algorithm, the experiments show that the solutions it produces are usually close to the true optimum. For the branch-and-bound algorithm, the decrease in complexity due to relaxations, in particular diagonal relaxations, is verified and quantified in terms of running time. The experiments also illustrate the dependence on properties of the matrix $\mathbf{Q}$, validating earlier results in Sections 3.4 and 3.6, and on parameters introduced in Section 3.7 for the branch-and-bound algorithm.

Section 4.2 presents several filter and array design examples. In Sections 4.2.1 and 4.2.2, the algorithms are used to design sparse equalizers for representative wireless communication channels. The high levels of sparsity observed by other researchers using heuristic algorithms is verified using our branch-and-bound algorithm. In Section 4.2.3, non-uniformly spaced MVDR beamformers are designed to detect signals propagating from target directions in the presence of noise and discrete interferers. The SNR is observed to increase relative to a uniformly spaced beamformer, especially in the vicinity of the interferers.

## 4.1 Numerical experiments

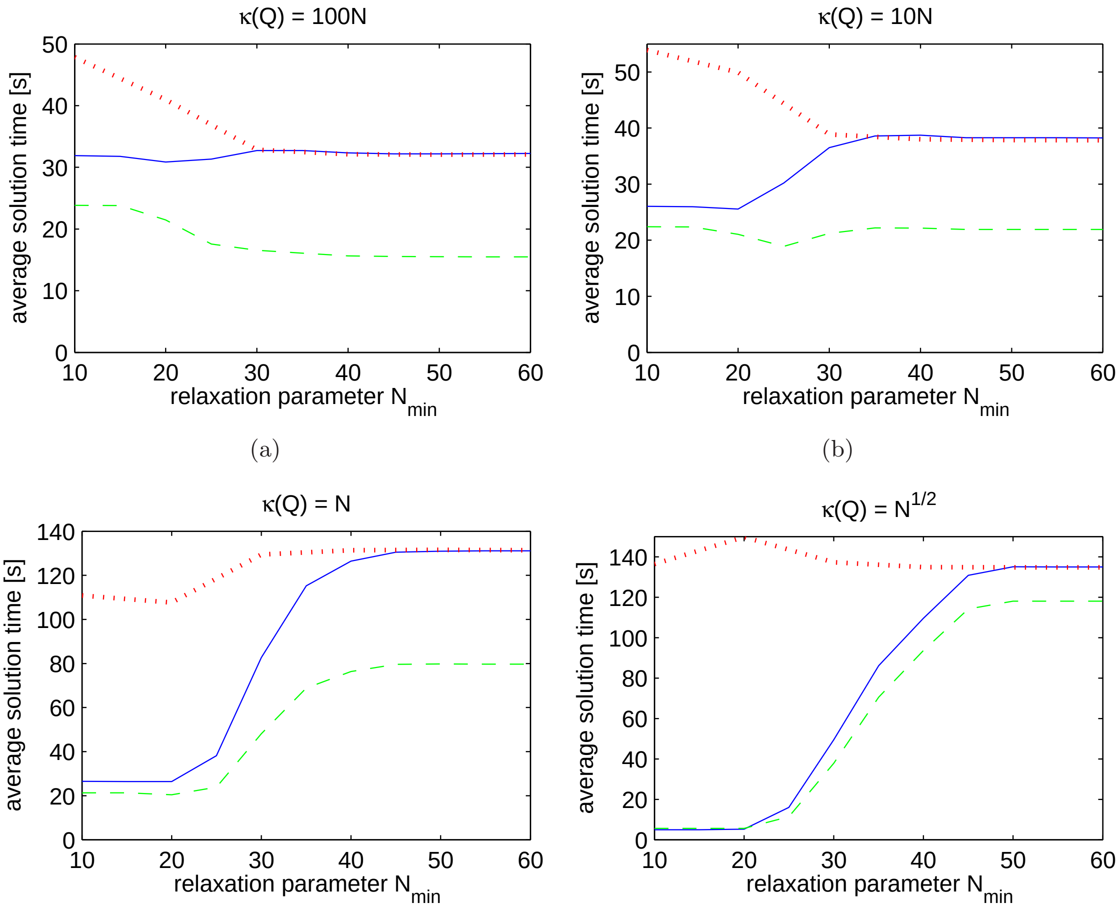
In this section, we discuss numerical experiments in which the successive thinning and branch-and-bound algorithms are applied to randomly generated instances of problem (2.0.1). The results support the claims made at the beginning of the chapter regarding the properties of the algorithms.

We use the same random generation methods in these experiments as in Section 3.6. The parameters $\mathbf{Q}$ and $\mathbf{c}$ are chosen randomly with $\gamma = 1$ and both the condition number $\kappa(\mathbf{Q})$ and the eigenvalue distribution of $\mathbf{Q}$ are varied. More details can be found in Section 3.6. The number of dimensions $N$ is fixed at 60 and $\kappa(\mathbf{Q})$ is set to $\sqrt{N}$, $N$, $10N$, and $100N$ as before. For each value of $\kappa(\mathbf{Q})$, one hundred (100) instances are created.

We also examine the effect of different choices in the branch-and-bound algorithm, namely the use of successive thinning to obtain feasible solutions to all subproblems, the type of relaxation used (linear or diagonal), and the value of the parameter $N_{\min}$ that controls when relaxations are solved (see Step 6 in Section 3.7). Each instance is solved repeatedly with different algorithm choices. The parameter $N_{\min}$ is varied between 10 and 60 to explore the trade-off inherent in solving relaxations. For $N_{\min} = 60$, no relaxations are solved except for the root problem. As $N_{\min}$ decreases, more relaxations are solved, leading to better lower bounds and more subproblems being pruned, but the amount of computation also increases. Whether relaxations improve the overall complexity depends on the quality of the lower bounds and the amount of pruning that result. As will be seen, in some cases the solution time decreases substantially as $N_{\min}$ decreases and more relaxations are solved, while in other cases the solution time can actually increase.

The algorithms are implemented in MATLAB running on a Fedora Linux computer with a 2.4 GHz quad-core processor and 3.9 GB of memory (only one core tends to be used at a time however). We use the average solution time as the measure of complexity for the branch-and-bound algorithm; the absolute times are less important than the relative times, which indicate gains or losses in efficiency.

In Fig. 4-1, we plot the average solution time of the branch-and-bound algorithm against the relaxation parameter $N_{\min}$ for the first method of generation in which the eigenvalue distribution $f_1(\lambda)$ is proportional to $1/\lambda$. The different line types refer to the following algorithm variants: diagonal relaxations with successive thinning only for the root problem, diagonal relaxations with successive thinning for all subproblems, and linear relaxations with successive thinning only for the root problem. For $\kappa(\mathbf{Q}) = 100N$, the best strategy appears to be to use successive thinning on all subproblems while solving as few relaxations as possible. Successive thinning roughly halves the average solution time, suggesting that the more rapid improvement of the incumbent solution outweighs the increase in computation. On the other hand, solving relaxations actually increases the solution time, which is perhaps expected given the relative weakness of the lower bounds shown in Fig. 3-7 for $\kappa(\mathbf{Q}) = 100N$.



(a)

(b)

(c)

(d)

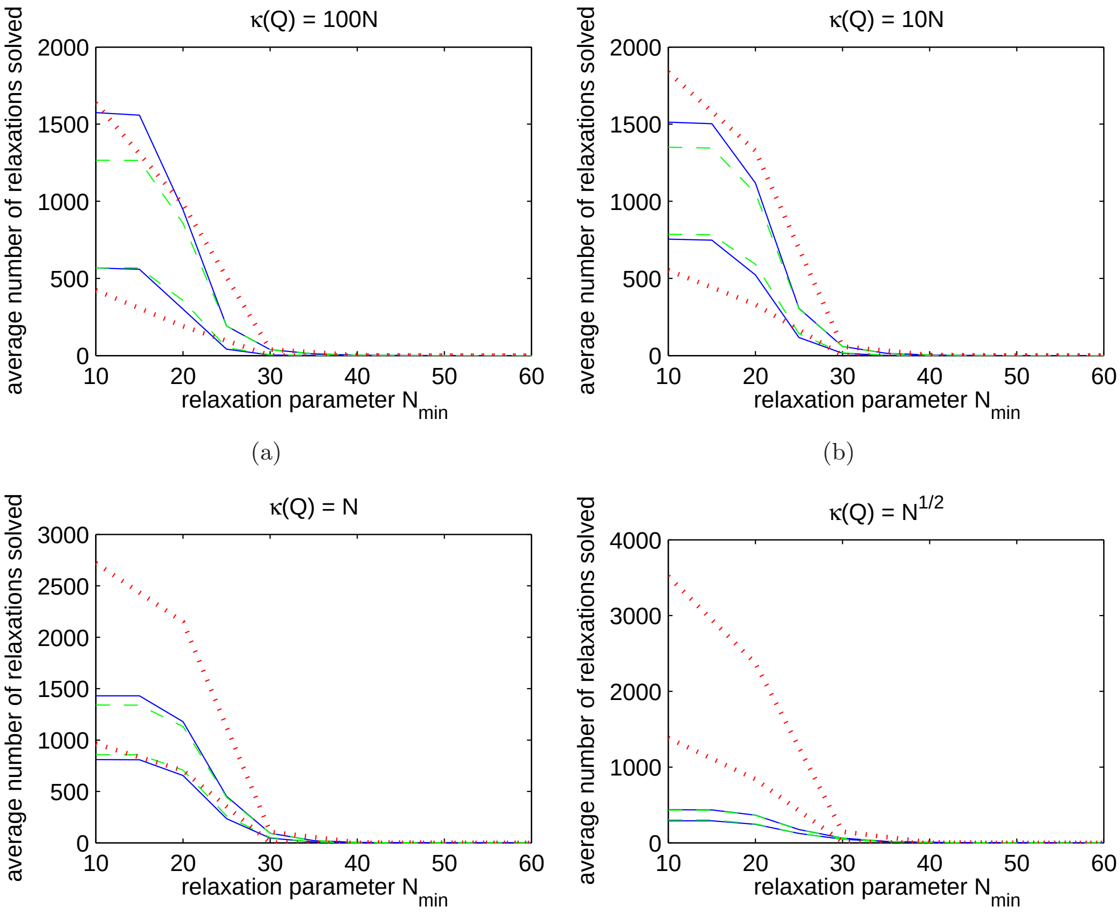
Figure 4-1: Average solution time as a function of the relaxation parameter $N_{\text{min}}$ for a $1/\lambda$ eigenvalue distribution. Solid blue line: diagonal relaxations, no successive thinning for subproblems; dashed green line: diagonal relaxations, successive thinning for subproblems; dotted red line: linear relaxations, no successive thinning for subproblems.

In contrast, for $\kappa(\mathbf{Q}) = 10N$ in Fig. 4-1(b), solving diagonal relaxations does decrease the average solution time as the resulting lower bounds improve. For the solid blue curve corresponding to diagonal relaxations and no successive thinning for subproblems, the minimum value at $N_{\min} = 20$ is approximately one third lower than the maximum. Adding successive thinning yields a further decrease in solution time. For $\kappa(\mathbf{Q}) = N$ and $\kappa(\mathbf{Q}) = \sqrt{N}$, the reduction in complexity due to diagonal relaxations becomes very pronounced. There is a 27-fold difference between the minimum and maximum values of the solid blue curve for $\kappa(\mathbf{Q}) = \sqrt{N}$. At the same time, the additional gain due to successive thinning decreases. The varying results of using successive thinning on subproblems is explained later using Table 4.1. As for linear relaxations, they are seen to offer no benefit at low $\kappa(\mathbf{Q})$ and to worsen solution times at high $\kappa(\mathbf{Q})$.

It is interesting to note that the intrinsic difficulty of problem instances seems to increase as the condition number decreases. This trend is suggested by the solution times at $N_{\min} = 60$ for the solid blue and dotted red curves, i.e., in the absence of relaxations and successive thinning, leaving only the simple lower bounds of Section 3.2. The solution times increase from 32 s for $\kappa(\mathbf{Q}) = 100N$ to 135 s for $\kappa(\mathbf{Q}) = \sqrt{N}$. Thus diagonal relaxations appear to be most beneficial for those instances that are inherently more difficult.

To gain more insight into the behavior seen in Fig. 4-1, we plot in Fig. 4-2 the average number of relaxations solved per instance as a function of $N_{\min}$. As an indication of effectiveness, we also plot the average number of relaxations that result in the pruning of the current subproblem. We refer to these relaxations as successful since they eliminate the need for further branching. Note that a successful relaxation of a larger subproblem eliminates more branches than a successful relaxation of a smaller subproblem. For $N_{\min} > 30$, the number of relaxations solved is small. The number of relaxations increases quickly as $N_{\min}$ decreases from 30 before levelling off around $N_{\min} = 15$. In the absence of successive thinning, the fraction of diagonal relaxations that are successful increases from roughly one third for $\kappa(\mathbf{Q}) = 100N$ to over two thirds for $\kappa(\mathbf{Q}) = \sqrt{N}$, as might be expected from Fig. 3-7. The total number of diagonal relaxations trends downward as the condition number decreases, a consequence of increased pruning. With the addition of successive thinning, the total number of diagonal relaxations decreases slightly while the number of successful diagonal relaxations increases slightly, both as a result of better incumbent solutions. In contrast, the number of linear relaxations increases as $\kappa(\mathbf{Q})$ decreases, reflecting the corresponding increase in solution times seen in Fig. 4-1.

Fig. 4-3 shows the average solution times for the second method of generation in which the eigenvalues of $\mathbf{Q}$ follow a uniform distribution. As expected from comparing Figs. 3-7

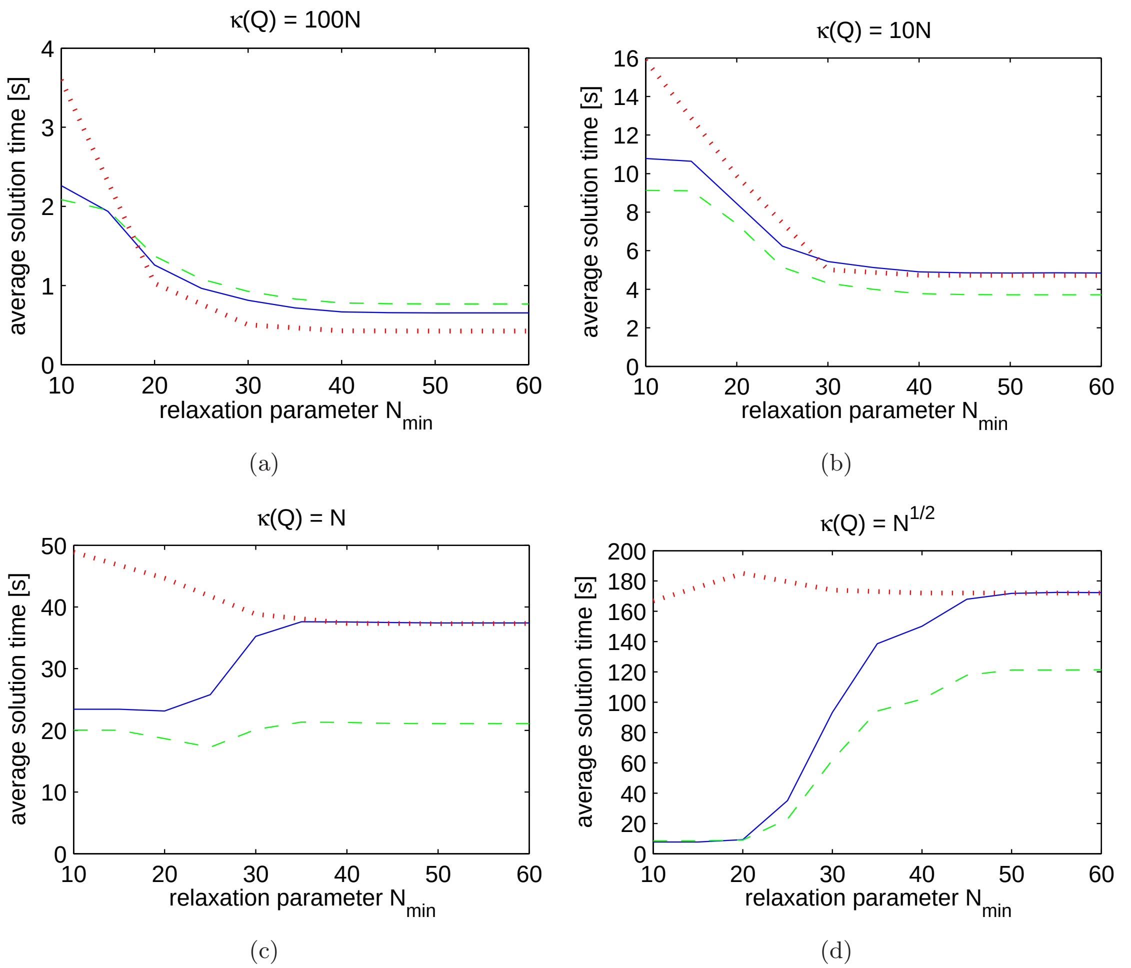


(a) (b) (c) (d)

Figure 4-2: Average number of relaxations solved as a function of the relaxation parameter $N_{\min}$ for a $1/\lambda$ eigenvalue distribution. Solid blue line: diagonal relaxations, no successive thinning for subproblems; dashed green line: diagonal relaxations, successive thinning for subproblems; dotted red line: linear relaxations, no successive thinning for subproblems. For each line type, the upper curve represents the total number of relaxations while the lower curve represents the number of successful relaxations.

and 3-9, the positive effect of solving diagonal relaxations is reduced and lower values of $\kappa(\mathbf{Q})$ are required to realize the same efficiency gains as in Fig. 4-1. For $\kappa(\mathbf{Q}) = 100N$, linear relaxations are in fact preferable to diagonal relaxations. The effect of using successive thinning on all subproblems is similar to that observed in Fig. 4-1. In the absence of any relaxations or successive thinning, the decrease in solution times with increasing condition number is even more apparent than for the $1/\lambda$ eigenvalue distribution. Indeed, for $\kappa(\mathbf{Q}) = 100N$, the solution times fall below 1 s and even the relatively low computational cost of applying successive thinning to subproblems is not justified.

Fig. 4-4 shows the average solution times for a $1/\lambda^2$ eigenvalue distribution. As can



(a)

(b)

(c)

(d)

Figure 4-3: Average solution time as a function of the relaxation parameter $N_{\min}$ for a uniform eigenvalue distribution. Solid blue line: diagonal relaxations, no successive thinning for subproblems; dashed green line: diagonal relaxations, successive thinning for subproblems; dotted red line: linear relaxations, no successive thinning for subproblems.

be predicted from Fig. 3-11, solving diagonal relaxations now improves efficiency by a substantial factor, even at $\kappa(\mathbf{Q}) = 100N$. For $\kappa(\mathbf{Q}) = \sqrt{N}$, the efficiency gain is over two orders of magnitude. Compared to diagonal relaxations, neither solving linear relaxations nor applying successive thinning to subproblems offers significant benefits.

The order of growth of the branch-and-bound algorithm with respect to the dimension $N$ is investigated in Fig. 4-5. For each $N$ and $\kappa(\mathbf{Q})$, one hundred instances are generated from a $1/\lambda$ eigenvalue distribution. Four algorithm variants are compared: diagonal relaxations with $N_{\min} = 20$, with and without successive thinning for subproblems, and no relaxations, also with and without successive thinning. It is clear that for $\kappa(\mathbf{Q}) = \sqrt{N}$, the rate of growth is substantially slower when diagonal relaxations are used. Even for $\kappa(\mathbf{Q}) = 100N$, the

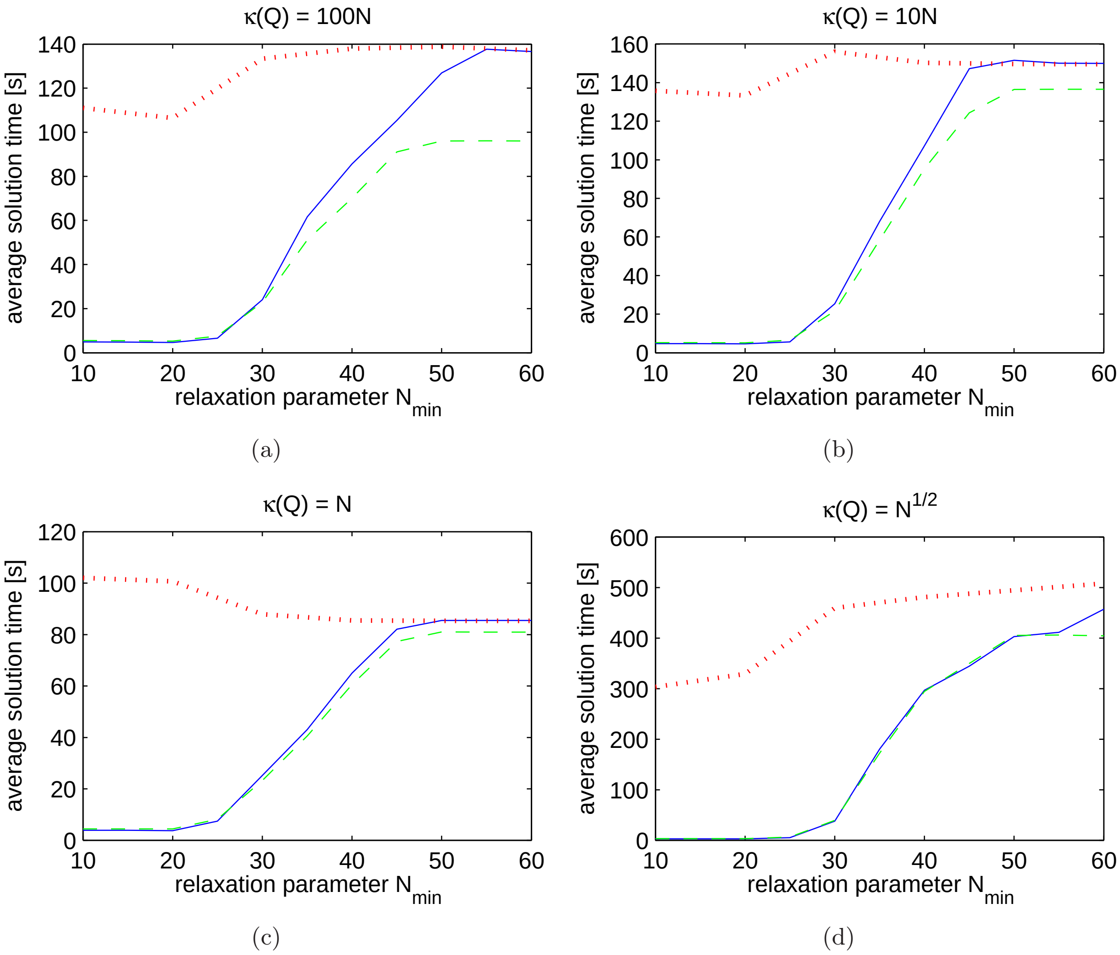
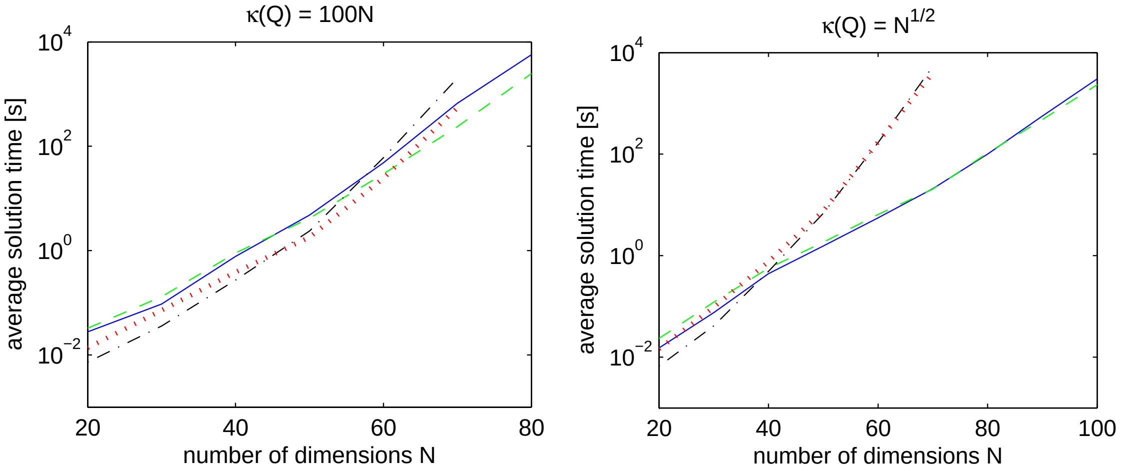


Figure 4-4: Average solution time as a function of the relaxation parameter $N_{\text{min}}$ for a $1/\lambda^2$ eigenvalue distribution. Solid blue line: diagonal relaxations, no successive thinning for subproblems; dashed green line: diagonal relaxations, successive thinning for subproblems; dotted red line: linear relaxations, no successive thinning for subproblems.

trend suggests that diagonal relaxations become more beneficial as $N$ increases, resulting in a slight gain in efficiency at $N = 70$ over the variants that do not use relaxations. Successive thinning is more effective at the higher condition number as seen earlier. Further experiments could be done to extend these results to higher dimensions.

In Fig. 4-6, we examine the behavior of the branch-and-bound algorithm acting on individual instances rather than in an average sense. The figure shows the progress of lower and upper bounds on the optimal cost as the number of iterations increases for two contrasting instances. For each algorithm variant, the lower bound plotted in Fig. 4-6 is the smallest of the lower bounds for currently open subproblems, while the upper bound is provided by the incumbent solution. In both instances, the lower bounds increase most rapidly



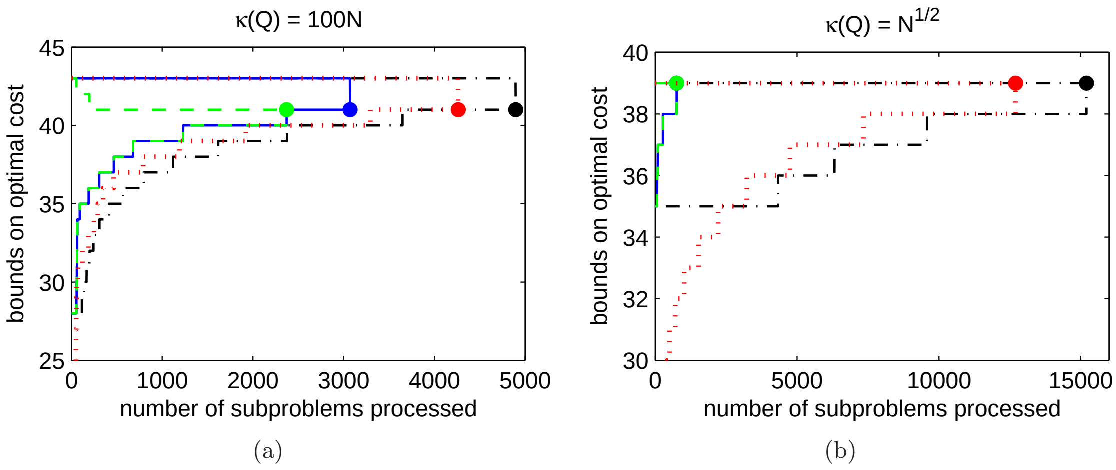
(a)

(b)

Figure 4-5: Growth of the average solution time with the number of dimensions $N$ for a $1/\lambda$ eigenvalue distribution. Solid blue line: diagonal relaxations, no successive thinning for subproblems, $N_{\min} = 20$; dashed green line: diagonal relaxations, successive thinning for subproblems, $N_{\min} = 20$; dash-dot black line: no relaxations, no successive thinning for subproblems; dotted red line: no relaxations, successive thinning for subproblems.

in the beginning and more slowly as they approach the true optimal value. The fastest improvement is achieved using diagonal relaxations, followed by linear relaxations and no relaxations. For the $\kappa(\mathbf{Q}) = 100N$ instance in Fig. 4-6(a), the slightly faster convergence of the lower bound when relaxations are solved is not enough to offset the added computation. As a consequence, the algorithm variant in which no relaxations are solved is the fastest in terms of solution time. On the other hand, for the instance in Fig. 4-6(b), the variants that use diagonal relaxations also take the least time. With respect to upper bounds, it is seen in Fig. 4-6(a) that the use of successive thinning for all subproblems results in much faster convergence to an optimal solution. For the other three variants, an optimal solution is found only after the lower bound has converged. In Fig. 4-6(b), the initial solution is already optimal.

We note that the solution times reported in this section depend on the specific algorithms used to obtain feasible solutions and lower bounds, and on how the algorithms are implemented. This is particularly true of solving relaxations, which is a computationally intensive step that is performed in a significant fraction of all iterations. Given an inefficient algorithm or implementation, there may be no benefit to solving diagonal relaxations even when the resulting lower bounds are strong. Likewise, the situation for linear relaxations



(a)

(b)

Figure 4-6: Lower and upper bounds on the optimal cost as functions of the number of subproblems processed for two contrasting instances generated using a $1/\lambda$ eigenvalue distribution. Solid blue line: diagonal relaxations, no successive thinning for subproblems, $N_{\min} = 20$; dashed green line: diagonal relaxations, successive thinning for subproblems, $N_{\min} = 20$; dotted red line: linear relaxations, no successive thinning for subproblems, $N_{\min} = 20$; dash-dot black line: diagonal relaxation only for the root problem, no successive thinning for subproblems, $N_{\min} = 60$. The filled circles indicate algorithm termination.

may improve somewhat with a more efficient algorithm or implementation.

Thus far we have concentrated on the behavior of the branch-and-bound algorithm. We are also interested in the extent to which the initial solutions provided by the successive thinning algorithm deviate from optimality. Since the branch-and-bound algorithm guarantees that the final solution is optimal, it is possible to measure this deviation exactly. Table 4.1 lists the average ratios between the cost of the initial successive thinning solution and the final optimal solution for different condition number values and eigenvalue distributions. In all cases, the successive thinning algorithm produces solutions that are close to optimal, within 5% or less on average. The quality of approximation follows the same pattern as for diagonal relaxations, i.e., worse for higher condition numbers and more uniform eigenvalue distributions. This dependence on condition number and eigenvalue distribution explains the varying results of applying successive thinning to subproblems that is seen in Figs. 4-1, 4-3, and 4-4. When the initial solution is already optimal or very close to it, there is little to no benefit in using successive thinning for subproblems because the incumbent solution barely improves while the amount of computation increases. This is the case at low condition number and for $\lambda(\mathbf{Q}) \sim 1/\lambda^2$ in Fig. 4-4. When the initial solution is farther from optimal at higher condition numbers, spending additional effort to improve the incumbent solution is justified.

Table 4.1: Average approximation ratios for the successive thinning algorithm.

| | $\kappa(\mathbf{Q}) = 100N$ | $\kappa(\mathbf{Q}) = 10N$ | $\kappa(\mathbf{Q}) = N$ | $\kappa(\mathbf{Q}) = \sqrt{N}$ |
|---|---|---|---|---|
| $\lambda(\mathbf{Q}) \sim 1/\lambda$ | 1.024 | 1.018 | 1.009 | 1.002 |
| $\lambda(\mathbf{Q}) \sim$ uniform | 1.052 | 1.038 | 1.013 | 1.003 |
| $\lambda(\mathbf{Q}) \sim 1/\lambda^2$ | 1.001 | 1.001 | 1.001 | 1.001 |

We summarize the results of this section. For the branch-and-bound algorithm, the use of diagonal relaxations was shown to significantly reduce complexity in many instances as measured by the average solution time. Moreover, diagonal relaxations tend to be the most beneficial for instances in which it is a good approximation to the original problem, i.e., when the matrix $\mathbf{Q}$ is well-conditioned or when most of the eigenvalues of $\mathbf{Q}$ are small as seen earlier in Section 3.6. For instances in which diagonal relaxations enhance efficiency and with $N = 60$, setting the relaxation parameter $N_{\text{min}} \sim 20$ appears to be a good choice. Linear relaxations on the other hand were observed to provide no gain. In addition, we saw that the successive thinning algorithm often produces nearly optimal designs. When the initial solution is very close to optimal, there is usually no benefit to applying successive thinning to all subproblems in the branch-and-bound algorithm.

## 4.2 Design examples

In this section, we present several equalizer and beamformer design examples that illustrate potential applications of the algorithms developed in Chapters 2 and 3.

### 4.2.1 Equalizers for an idealized multipath communication channel

In this subsection and in Section 4.2.2, we discuss the design of sparse equalizers for multipath communication channels. The approximate sparsity of multipath channel responses has been exploited by many researchers to reduce the number of non-zero equalizer coefficients [33]– [42]. In particular, it has been observed that the trade-off between sparsity and MSE is quite favorable in the sense that the number of non-zero coefficients can be reduced substantially with only a small increase in MSE. Nearly all of the references cited above use heuristic design algorithms. In this section, we use the branch-and-bound algorithm to establish the best possible trade-off and thereby verify the previous observation. We also explore the effect of other parameters such as the equalizer length and the input SNR.

The application of our framework to sparse equalizer design was introduced in Section 2.1.2. Equation (2.1.11) specifies the channel model and (2.1.10) and (2.1.12) together specify the values of the parameters $\mathbf{Q}$, $\mathbf{f}$, and $\beta$ in constraint (2.1.3) in terms of the channel parameters. In this subsection, the channel response $h[n]$ is chosen to represent an ideal multipath channel with a direct path and two delayed paths that are aligned with the sampling grid. More precisely,

$$h[n] = \delta[n] + a_1\delta[n - N_1] + a_2\delta[n - N_2],$$

where the delays $N_1$ and $N_2$ are positive integers and the amplitudes $a_1$ and $a_2$ are sampled randomly from the interval $[-1, 1]$. A more realistic channel response with more delayed paths that are not aligned with the sampling grid is considered in Section 4.2.2. In both cases, we assume that the transmitted sequence $x[n]$ and the noise $\eta[n]$ are white so that $\phi_{xx}[m] = \sigma_x^2\delta[m]$ and $\phi_{\eta\eta}[m] = \sigma_\eta^2\delta[m]$. Either $\sigma_x$ or $\sigma_\eta$ can be normalized to 1 since the key parameter is the signal-to-noise ratio $\mathrm{SNR}_0 = \sigma_x^2/\sigma_\eta^2$. We also assume that $x[n]$ is estimated with a delay $\Delta$, i.e., $x[n]$ in (2.1.9) is changed to $x[n - \Delta]$, to accommodate the causality of both the channel and the equalizer. Equations (2.1.10b) and (2.1.12b) are modified as a result, yielding

$$f_n = \phi_{xy}[n - \Delta] = \sigma_x^2 h[\Delta - n].$$

In this subsection, $\Delta$ is chosen to be equal to the largest channel delay, i.e., $\Delta = N_2$.

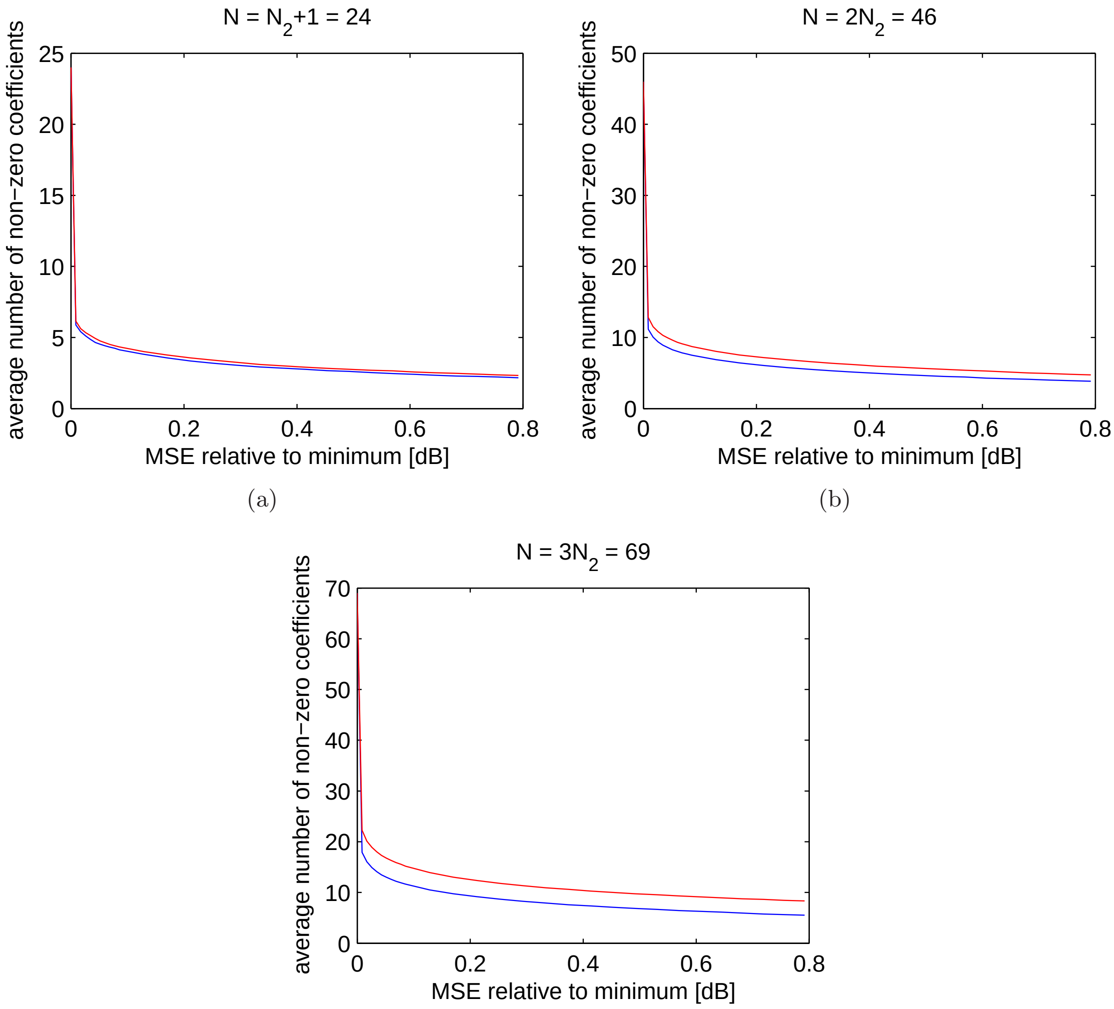
In the design experiments, we consider equalizer lengths $N$ ranging from the length of the channel, $N_2 + 1$, to three times the channel length. For a fixed length $N$, the MMSE equalizer (i.e., the causal Wiener filter of length $N$) is given by $\mathbf{c} = \breve{\mathbf{Q}}^{-1}\breve{\mathbf{f}}$, where $\breve{\mathbf{Q}} = \sigma_x^{-2}\mathbf{Q}$ and $\breve{\mathbf{f}} = \sigma_x^{-2}\mathbf{f}$ are normalized parameters that depend only on $h[n]$ and $\mathrm{SNR}_0$. The corresponding MMSE is

$$\delta_{\min} = \sigma_x^2\left(1 - \breve{\mathbf{f}}^T\breve{\mathbf{Q}}^{-1}\breve{\mathbf{f}}\right).$$

To design sparse equalizers of the same length $N$, we set the MSE tolerance $\delta$ in (2.1.9) to be slightly higher than $\delta_{\min}$ so that solutions other than $\mathbf{b} = \mathbf{c}$ become feasible. The parameter $\gamma$ is then given by $\delta - \delta_{\text{min}}$ as discussed in Section 2.1.2. The ratio $\delta/\delta_{\text{min}}$ is a normalized measure of performance degradation relative to the MMSE equalizer. It will be seen throughout this subsection and Section 4.2.2 that significant sparsity can be attained even for $\delta/\delta_{\text{min}}$ close to 1. With $\mathbf{Q}$, $\mathbf{c}$, and $\gamma$ determined as described above, problem (2.0.1) may be solved using the branch-and-bound algorithm to obtain the sparsest equalizer with an MSE of at most $\delta$. Typically, we set the relaxation parameter $N_{\text{min}}$ in Algorithm 2 equal to 20 and do not use successive thinning for subproblems.

In Fig. 4-7, we show the number of non-zero coefficients, averaged over 1600 amplitude pairs $(a_1, a_2)$, as a function of the MSE ratio $\delta/\delta_{\text{min}}$ for $N_1 = 7$, $N_2 = 23$, equalizer lengths $N = N_2+1$, $2N_2$, $3N_2$, and $\text{SNR}_0 = 10, 25$ dB. In each panel, the left-most point corresponds to the MMSE equalizer, which generally does not have any zero values. However, for this idealized example, the MMSE equalizer is close to being exactly sparse. Hence there is an abrupt decrease in the number of non-zero coefficients as soon as $\delta/\delta_{\text{min}}$ exceeds 1, followed by a rapid approach toward an asymptote. These curves are consistent, albeit in an exaggerated fashion, with results in the literature and later in Section 4.2.2. The gain in sparsity is slightly smaller for the higher SNR value and the behavior is very similar for all values of $N$.

Next we examine in Fig. 4-8 the effect of the length $N$ on both the MMSE and the number of non-zero coefficients in a sparse equalizer. For this experiment, the MSE ratio $\delta/\delta_{\text{min}}$ is fixed at 1.05, $\text{SNR}_0 = 10$ dB, and each data point again represents the average of 1600 $(a_1, a_2)$ pairs. The staircase patterns can be explained by reference to the infinite-length MMSE equalizer (infinite-length causal Wiener filter). For this idealized channel, the infinite-length equalizer tends to have non-zero values only at integer combinations of $N_1$ and $N_2$. The coefficients in these integer combinations do not have to be strictly non-negative because the channel is not necessarily minimum-phase. Furthermore, the coefficients of finite-length MMSE equalizers are approximately given by truncated versions of the infinite-length coefficients. Thus as $N$ increases, significant non-zero values are incorporated in the finite-length approximations only at certain values of $N$. As a result, the MMSE decreases the most at these points. For $N_1 = 7$ and $N_2 = 23$ in Fig. 4-8(a), the largest decreases occur at $N - 1 = 30 = N_2 + N_1$, $39 = 2N_2 - N_1$, and $46 = 2N_2$, followed by smaller decreases at other integer combinations of $N_1$ and $N_2$. Similarly, the number of non-zero coefficients in sparse equalizers increases at these special values of $N$. The same phenomenon is seen in



(a)

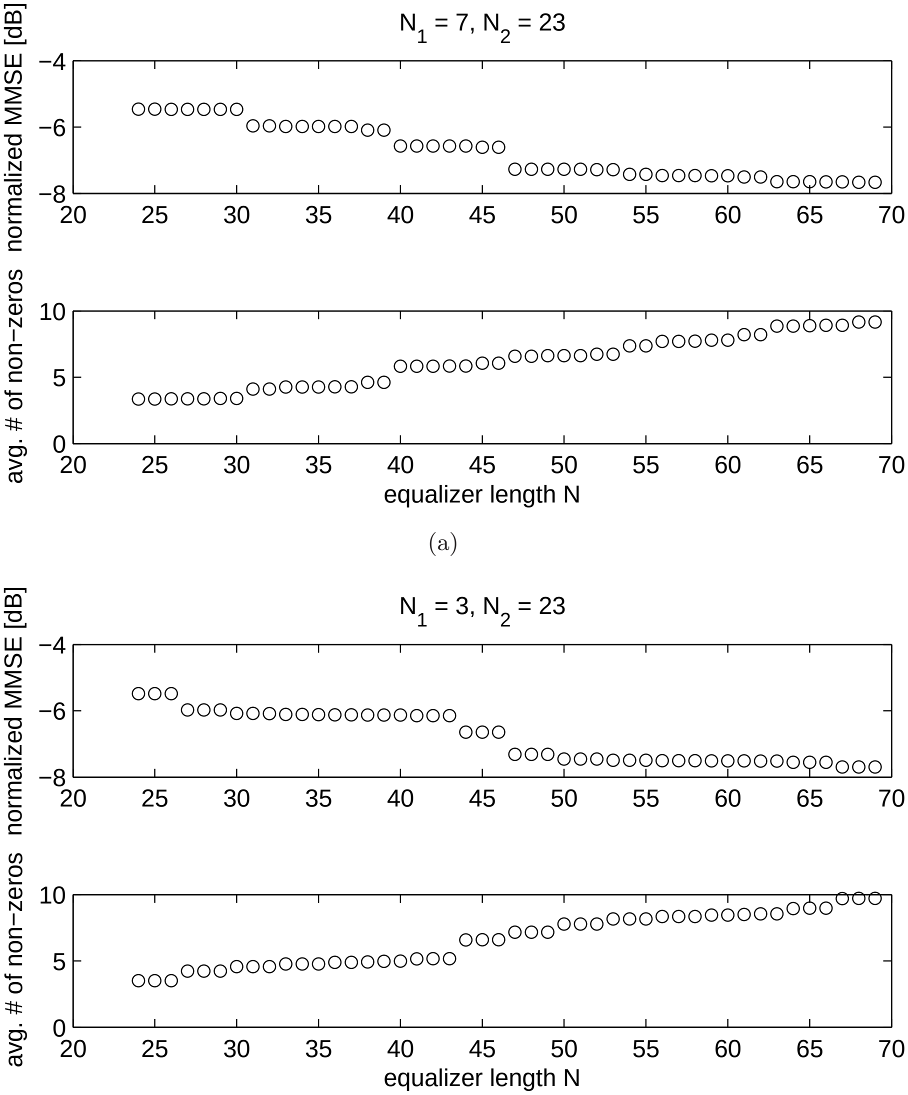
(b)

(c)

Figure 4-7: Average number of non-zero coefficients as a function of the MSE ratio $\delta/\delta_{\min}$. The lower blue curves correspond to $\text{SNR}_0 = 10$ dB and the upper red curves to $\text{SNR}_0 = 25$ dB.

Fig. 4-8(b) for $N_1 = 3$ and $N_2 = 23$ with the largest changes at $N - 1 = 26 = N_2 + N_1$, $43 = 2N_2 - N_1$, and $46 = 2N_2$.

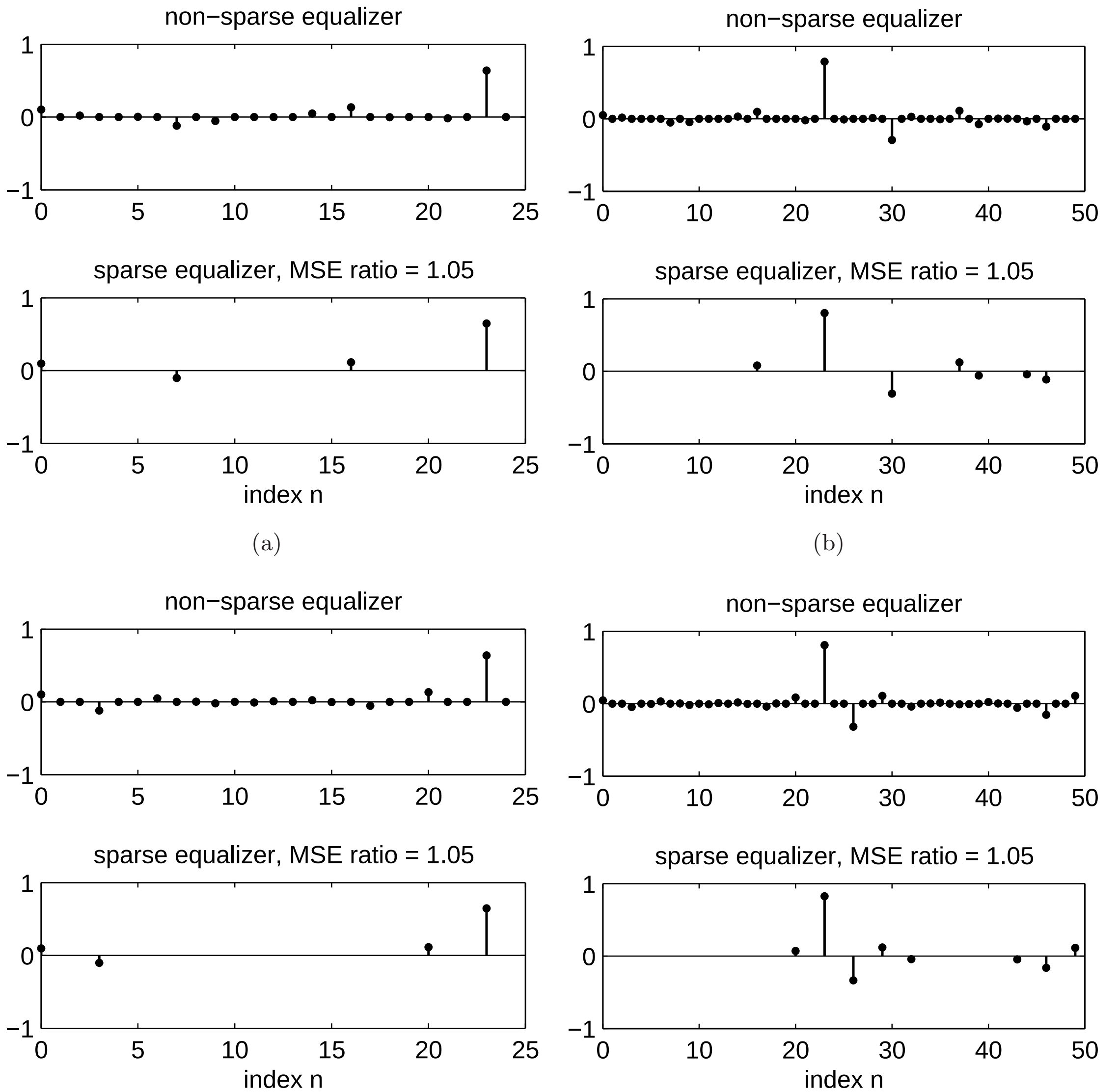
To reinforce the results in Fig. 4-8, we show in Fig. 4-9 the coefficient values of MMSE and sparse equalizers for two different values of $N$ and the same two $N_1$ values used in Fig. 4-8. As before, $N_2 = 23$, the MSE ratio $\delta/\delta_{\min} = 1.05$ and $\text{SNR}_0 = 10$ dB. It is seen that the large values in the MMSE equalizers occur at integer combinations of $N_1$ and $N_2$, and that the sparse equalizers tend to retain the largest of these coefficients.



(a)

(b)

Figure 4-8: MMSE normalized by $\sigma_x^2$ and average number of non-zero coefficients for sparse equalizers as functions of the equalizer length $N$. The MSE for the sparse equalizers is 5% higher than the corresponding MMSE.



(a)

(b)

(c)

(d)

Figure 4-9: Coefficient values of MMSE and sparse equalizers for (a) $N = 25$, $N_1 = 7$, $N_2 = 23$, (b) N $= 50, N_1 = 7$, $N_2 = 23$, (c) $N = 25$, $N_1 = 3$, $N_2 = 23$, (d) N $= 50, N_1 = 3$, $N_2 = 23$. Zero values are omitted.

Fig. 4-10 plots the average number of non-zero coefficients against the parameter $\mathrm{SNR}_0$ for $N = 30$, $N_1 = 7$, $N_2 = 23$, and $\delta/\delta_{\min} = 1.02$. The number of non-zero coefficients increases monotonically with $\mathrm{SNR}_0$. This dependence on $\mathrm{SNR}_0$ has also been noted in [41,42] and can be understood by considering the limits as $\mathrm{SNR}_0 \to \infty$ and $\mathrm{SNR}_0 \to 0$. In the first case, the MMSE equalizer converges to the channel inverse, whereas in the second case, it tends toward a matched filter for the channel response. The former is less sparse than the latter, which accounts for the trend in Fig. 4-10.

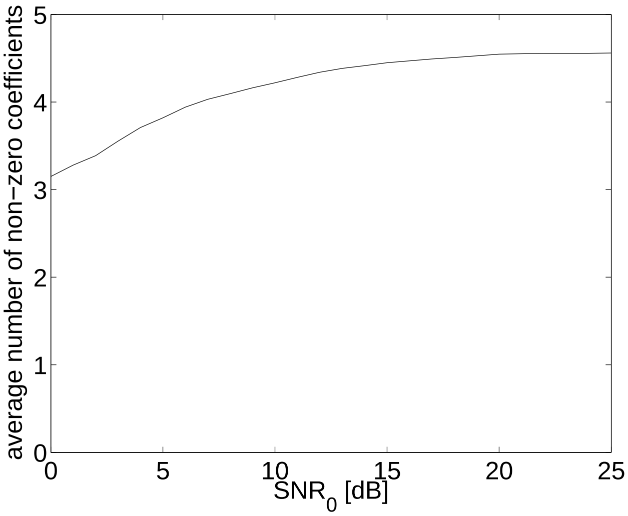


Figure 4-10: Average number of non-zero coefficients as a function of $\mathrm{SNR}_0$ for $N = 30$, $N_1 = 7$, $N_2 = 23$, and $\delta/\delta_{\min} = 1.02$.

In the experiments of this subsection, the combination of the diagonal relaxation and the methods of Section 3.2 often yields initial lower bounds for the branch-and-bound algorithm that match or nearly match the initial cost value. Hence the number of iterations is small, even zero, in a large majority of instances. The tightness of the initial lower bound may be due to a number of factors present in this idealized example, including the high level of sparsity, the relative unambiguity regarding which coefficients should be non-zero, and the diagonally dominant structure of the matrix $\mathbf{Q}$. In addition, we have observed in these experiments that it is rare for the initial successive thinning solution not to be optimal.

### 4.2.2 Equalizers for a realistic wireless communication channel

The experiments in Section 4.2.1 are extended to a more realistic wireless channel, specifically a test channel used to evaluate terrestrial broadcast systems for high-definition television. This example was also considered in [37, 38]. To make the design problem more tractable for the branch-and-bound algorithm, the channel is simplified by halving all of the multipath delays and by converting complex amplitudes to real values with the same magnitude. The modified multipath parameters are shown in Table 4.2. The effective discrete-time channel response is given by

$$h[n] = \sum_{i=0}^{5} a_i p(n - \tau_i),$$

where the pulse $p(t)$ is the convolution of the transmit and receive filter responses and the sampling period has been normalized to unity. Following [37, 38], we assume that the transmit and receive filters are square-root raised-cosine filters with excess bandwidth parameter $\beta = 0.115$. The resulting DT channel response is plotted in Fig. 4-11. The remainder of the experimental setup is the same as in Section 4.2.1. The estimation delay $\Delta$ is set to $0.8L + 0.2N$, where $L = 54$ is the largest delay in the channel (rounded up).

Table 4.2: Multipath parameters for the HDTV broadcast example.

| $i$ | $\tau_i$ | $a_i$ |
|---|---|---|
| 0 | 0 | 0.5012 |
| 1 | 4.84 | $-1$ |
| 2 | 5.25 | 0.1 |
| 3 | 9.68 | 0.1259 |
| 4 | 20.18 | $-0.1995$ |
| 5 | 53.26 | $-0.3162$ |

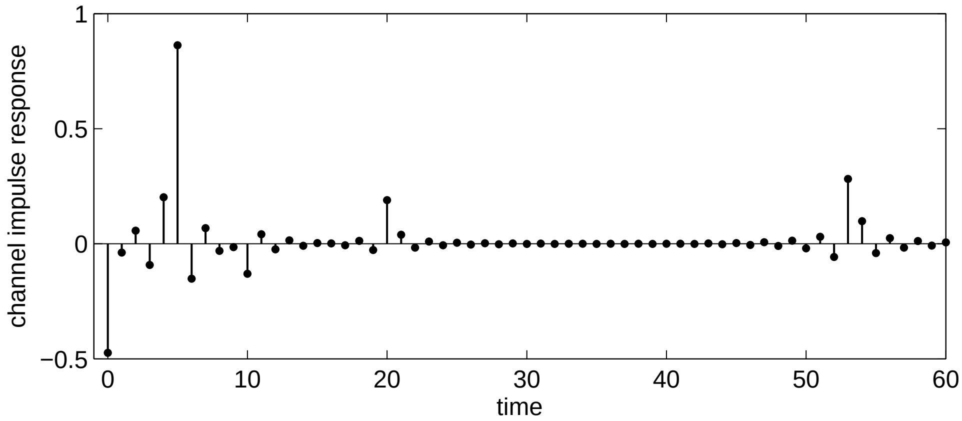


Figure 4-11: Effective discrete-time channel response for the HDTV broadcast example.

In Fig. 4-12, we plot the minimum number of non-zero equalizer coefficients against the MSE ratio $\delta/\delta_{\min}$ for an equalizer length of $N = L + 1 = 55$ and $\mathrm{SNR}_0 = 10, 25$ dB. The MMSE equalizers achieve MSE values (normalized by the signal power $\sigma_x^2$) of $-5.74$ and $-7.37$ dB respectively for $\mathrm{SNR}_0 = 10, 25$ dB. At low MSE ratios, the decrease in the number of non-zero coefficients is still fairly steep despite the channel response not being exactly sparse in this example. For $\mathrm{SNR}_0 = 10$ dB in particular, the number is nearly halved with only a 0.1 dB increase in MSE. The curves then level out beyond 1 dB. This behavior has been observed previously using heuristic algorithms (e.g. in [42]) and is now confirmed by the branch-and-bound algorithm.

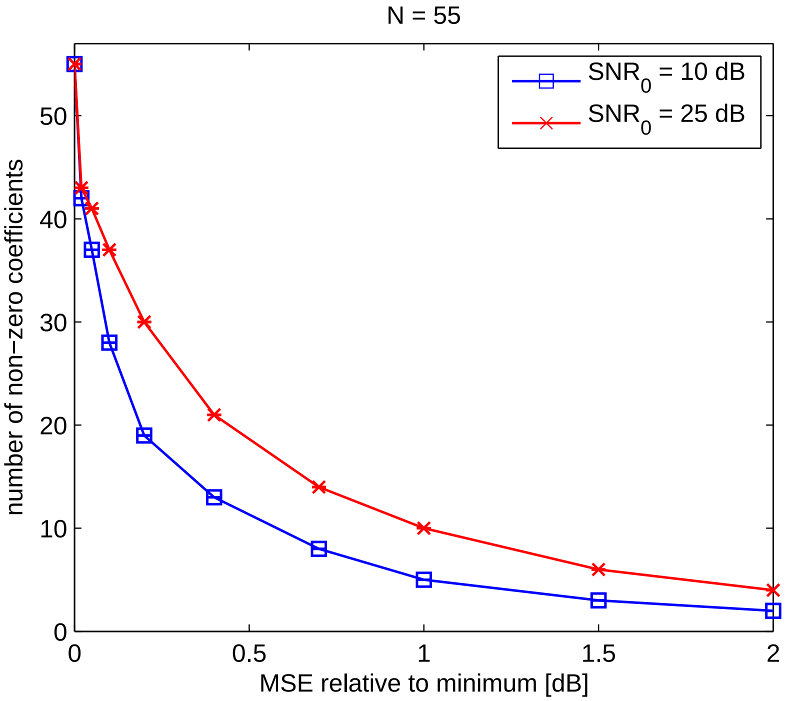


Figure 4-12: Number of non-zero equalizer coefficients as a function of the MSE ratio $\delta/\delta_{\min}$ for an equalizer length of $N = 55$.

Fig. 4-13 shows the coefficient values for the length 55 MMSE equalizer for $\mathrm{SNR}_0 = 10$ dB and a sparse equalizer with an MSE that is 0.2 dB higher. The sparse equalizer has about one third as many non-zero coefficients as the MMSE equalizer. The larger coefficients in the MMSE equalizer tend to be retained in the sparse equalizer, including a cluster surrounding the largest coefficient that corresponds to the strongest path in the channel.

Figs. 4-14 and 4-15 depict the same sparsity-MSE trade-off as in Fig. 4-12 for equalizer lengths of $N = 82$ and $N = 109$, which are respectively 1.5 and 2 times the channel length,

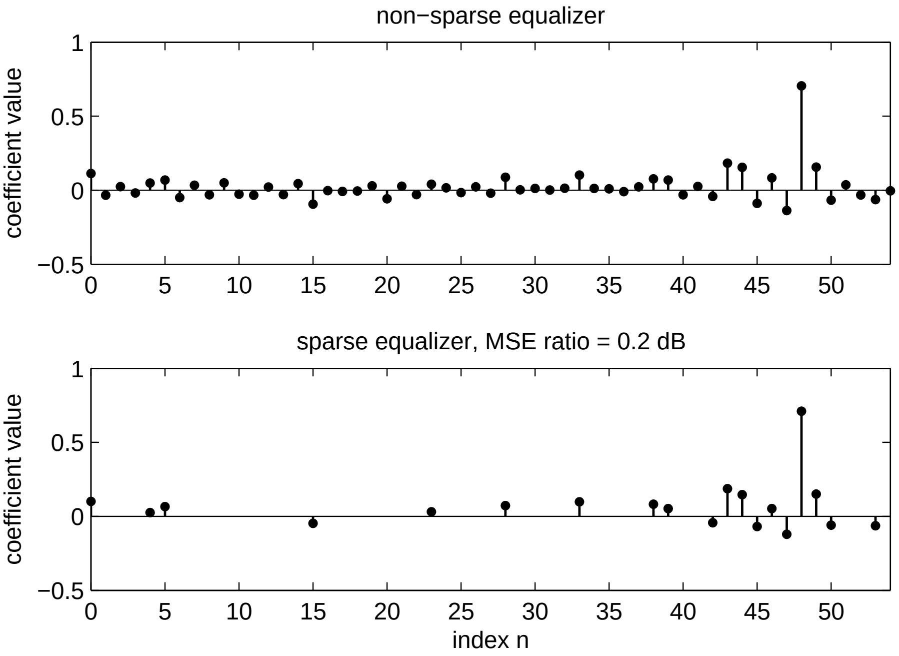


Figure 4-13: Coefficient values for the length 55 MMSE equalizer for $\text{SNR}_0 = 10$ dB and a corresponding sparse equalizer with MSE ratio $\delta/\delta_{\min} = 0.2$ dB. Zero values are omitted.

and $\text{SNR}_0 = 10, 25$ dB. The normalized MMSE values are now $-6.29$ and $-8.28$ dB for $N = 82$ and $\text{SNR}_0 = 10, 25$ dB, and $-7.18$ and $-11.06$ dB for $N = 109$ and the same SNR values. Similar trade-offs are observed although the relative decreases in the number of nonzero coefficients are smaller than before, especially for $\text{SNR}_0 = 25$ dB. With the significant increase in dimension, some of the problem instances become quite computationally complex for the branch-and-bound algorithm despite the efficiency improvements made in this thesis. To keep the computational load manageable, we have limited the solution time to one hour per instance. If the algorithm does not converge within one hour, it produces both a feasible solution and a lower bound on the optimal cost. The lower bound returned is the minimum of the lower bounds for open subproblems at the time of termination, and is indicated by an error bar in the figures. Recall that the subproblem selection rule (Step 1 in Section 3.7) is designed to improve this lower bound as quickly as possible.

Because of early termination, for some instances we cannot conclude that the final solution is optimal. However, the branch-and-bound algorithm does provide strong upper and lower bounds on the optimal cost, in contrast to a heuristic algorithm which only gives an upper bound. Furthermore, in most of the instances for which the branch-and-bound algorithm does converge, the initial solution provided by the successive thinning algorithm

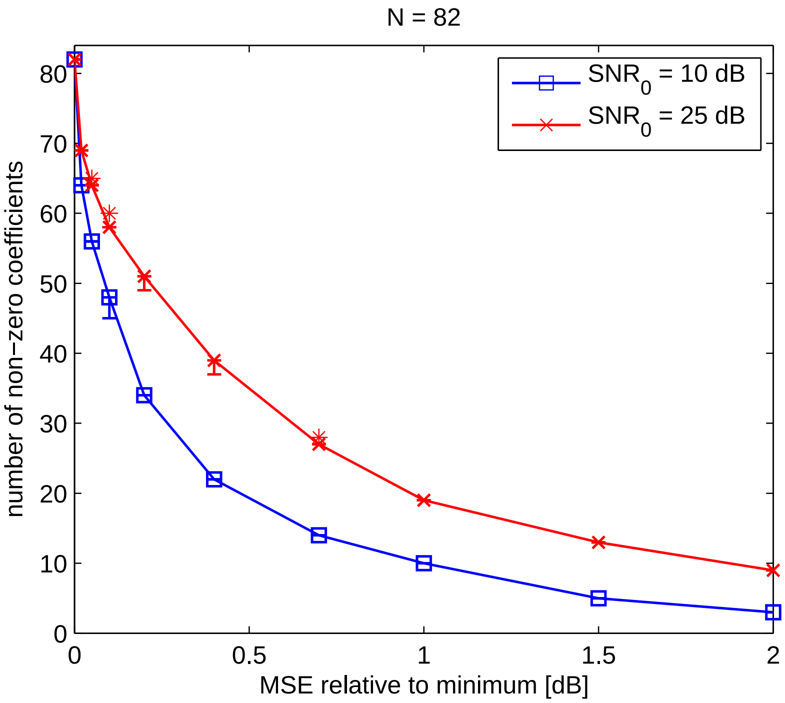


Figure 4-14: Number of non-zero equalizer coefficients as a function of the MSE ratio $\delta/\delta_{\min}$ for an equalizer length of $N = 82$. Points with error bars represent instances for which the branch-and-bound algorithm did not converge; the square marks the cost of the final solution while the error bar represents the final lower bound. Asterisks just above the upper curve indicate the cost of the initial successive thinning solution when it differs from the final cost.

turns out to be optimal. The few exceptions are indicated by asterisks in Figs. 4-14 and 4-15 marking the cost of the initial successive thinning solution. Our experience suggests that for the instances that did not converge, the final solutions are also optimal or very close to optimal, and further iterations will only cause the lower bound to increase until convergence.

We note that the instances in Figs. 4-14 and 4-15 that fail to converge tend to have optimal values near $N/2$. From a naive estimate of the problem complexity, we would expect these instances to be the most difficult since the number of ways of selecting $K$ zero-valued coefficients out of $N$, $\binom{N}{K}$, is strongly peaked around $K = N/2$. An optimal value around $N/2$ would necessitate searching through a very large number of combinations.

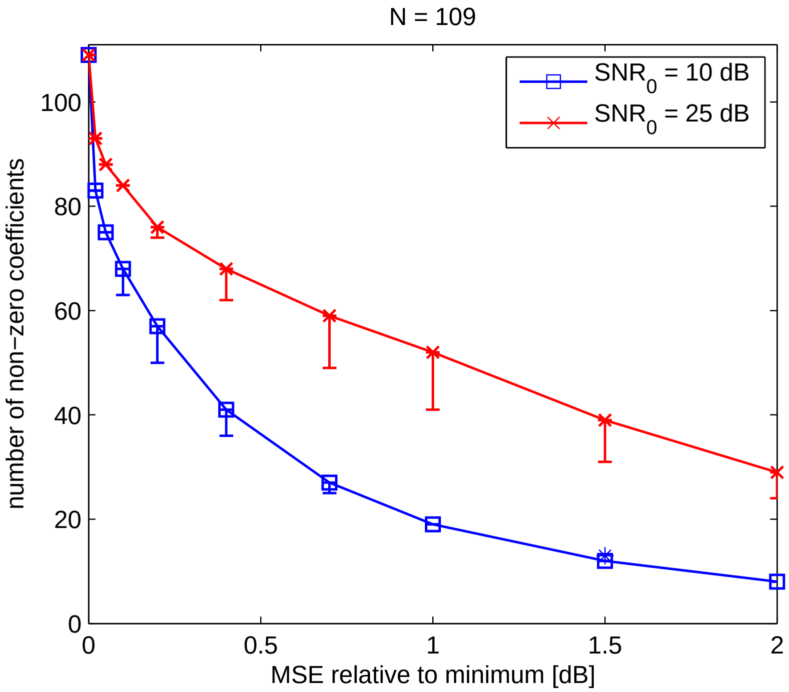


Figure 4-15: Number of non-zero equalizer coefficients as a function of the MSE ratio $\delta/\delta_{\text{min}}$ for an equalizer length of $N = 109$. As in Fig. 4-14, error bars represent final lower bounds for instances that did not converge and asterisks indicate the initial cost value when it differs from the final cost.

### 4.2.3 MVDR beamformers

The framework developed in Chapters 2 and 3 can also be used to design sparse MVDR beamformers, specifically as an example of the basic detection problem discussed in Section 2.1.3. As mentioned in that section, in the beamforming context the target signal $\mathbf{s}$ corresponds to a propagation direction of interest and $\mathbf{R}$ represents the covariance of the array input due to noise and interference.

In this subsection, we consider non-uniformly spaced beamformers. More specifically, the positions of the elements are constrained to an underlying uniform grid, but the number of available positions is larger than the number of active elements so that the array is sparse. Compared to a uniformly spaced array with the same number of elements, the longer length improves the rejection of interference close to the desired direction. The increased freedom in the placement of active elements also improves the SNR overall.

To apply the algorithms developed in the thesis, we focus on a real-valued formulation of the beamforming problem as opposed to the more conventional complex-valued formulation. Although the reduction of the sparse detection problem to (2.0.1) in Section 2.1.3 can be generalized to the complex case with minor modifications, some of the subsequent algorithms assume real values. The complex-valued generalization of these algorithms is an area for future study. In a real-valued formulation, we assume that narrowband signals $A\cos(\omega(t - n\Delta t) + \phi)$ are received by the array sensors, where $n$ is the element index and $\Delta t = d\cos\theta/c$ is the relative time delay corresponding to an inter-element spacing of $d$ and a propagation angle of $\theta$ measured from the array axis. Assuming that $d$ is equal to half the wavelength, a straightforward calculation shows that the array input is a linear combination of $\cos(n\pi\cos\theta)$ and $\sin(n\pi\cos\theta)$ and these are the only spatially-varying and angle-dependent quantities. Hence the propagation direction is encoded by two vectors, referred to as the array manifold vectors, with components given by $\cos(n\pi\cos\theta)$ and $\sin(n\pi\cos\theta)$, and detecting signals in the direction $\theta$ reduces to determining the components of the input along the array manifold vectors. Furthermore, because of the orthogonality of $\cos(n\pi\cos\theta)$ and $\sin(n\pi\cos\theta)$, the sine components can be neglected when targeting a cosine component. Therefore we restrict attention in the sequel to cosine array manifold vectors.

In the example that we consider, a desired signal at angle $\theta_0$ is to be detected in the presence of discrete interferers at $\theta_1$ and $\theta_2$ and isotropic (white) noise $\boldsymbol{\eta}$. The array input is given by

$$y_n = A_0\cos(n\pi\cos\theta_0) + \sum_{i=1}^{2} A_i\cos(n\pi\cos\theta_i) + \eta_n,$$
$$n = -\frac{N-1}{2}, -\frac{N-1}{2}+1, \ldots, \frac{N-1}{2}-1, \frac{N-1}{2},$$

where the element indices $n$ have been chosen symmetrically for convenience. The desired amplitude $A_0$ is regarded as deterministic while the interferer amplitudes $A_1$ and $A_2$ are regarded as zero-mean random variables with variances $\sigma_1^2$ and $\sigma_2^2$. For normalization purposes we set $A_0 = 1$. With $\mathbf{s}_i$ denoting the array manifold vector with components $\cos(n\pi\cos\theta_i)$, the covariance of the array output is

$$\mathbf{R} = \sigma_\eta^2\mathbf{I} + \sum_{i=1}^{2}\sigma_i^2\mathbf{s}_i\mathbf{s}_i^T, \tag{4.2.1}$$

where $\sigma_\eta^2$ is the white noise variance. In this example, the interferer powers are fixed at 10 and 25 dB respectively relative to the white noise power, and the interferer angles are fixed at $\cos\theta_1 = 0.18$ and $\cos\theta_2 = 0.73$ while the target angle is swept from $\cos\theta_0 = 0$ to $\cos\theta_0 = 1$.

In the design experiments, the number of elements $M$ is fixed at 30 and four different lengths $N = 30, 40, 50, 60$ are considered. For each $N$ and target angle $\theta_0$, the output SNR, defined as the ratio of the mean of the array output to the standard deviation, is to be maximized. For $N = 30$, the SNR is maximized by the non-sparse MVDR solution, i.e., $\mathbf{b} \propto \mathbf{R}^{-1}\mathbf{s}_0$. For $N > 30$, the branch-and-bound algorithm of Chapter 3 is used to maximize the SNR given $M = 30$ active elements. This is done through a search over SNR values, i.e., values of $\rho$ in (2.1.13). The search is initialized at the maximum SNR for $N = 30$, which is always achievable when $N > 30$, and proceeds in 0.05 dB increments. For each value of $\rho$, the branch-and-bound algorithm is run to determine whether a feasible solution to problem (2.1.13) exists. The translation of parameters to the canonical formulation (2.0.1) is as described in Section 2.1.3. The branch-and-bound algorithm can be terminated as soon as a feasible solution with $M$ or fewer non-zero weights is found, or when all of the subproblem lower bounds exceed $M$, implying that no such solution exists. The relaxation parameter $N_{\text{min}}$ is set to 20 and successive thinning is applied only to the root problem.

The branch-and-bound algorithm converges quickly (in under a second to a few seconds on a MacBook Pro with a 2.4 GHz dual-core processor and 4 GB of memory) for nearly all values of $N$ and $\theta_0$. However, for a few angles $\theta_0$, the problem instances are much more difficult to solve. Many of these difficult angles correspond to array manifold vectors $\mathbf{s}_0$ that have equal or nearly equal components, e.g. for $\cos\theta_0 = 0$ or $\cos\theta = 1/2$, and are also nearly orthogonal to the interference vectors $\mathbf{s}_1$ and $\mathbf{s}_2$. In these cases, it can be shown that the quadratic form on the left-hand side of (2.1.16) has a similar value for all subsets $\mathcal{Y}$ of a given size. The large number of similar subsets is a potential explanation for the high complexity that is observed. To avoid excessively long searches, the solution time is restricted to one hour as in Section 4.2.2. As a consequence, for certain values of $\theta_0$ and SNR we are unable to conclude whether or not a solution with $M$ non-zero coefficients exists. In the following figures, SNR values plotted with a solid line represent the highest SNR for which a feasible solution with $M$ non-zero weights was found. For cases in which the branch-and-bound algorithm did not converge, crosses indicate the highest possible SNR consistent with the final lower bound, i.e., the highest value for which the lower bound did not exceed $M$.

In Fig. 4-16, we plot the SNR as a function of the target angle $\theta_0$ for all four array lengths. For clarity, crosses indicating non-convergence are suppressed. For each $\theta_0$, the SNR values are normalized so that a value of 0 dB represents the maximum SNR achievable with $N = M = 30$ and white noise alone, i.e., in the absence of the interference terms in (4.2.1). With the addition of interference, it follows that the SNR curve for $N = 30$ must remain below 0 dB at all angles. The most severe losses occur when the target angle coincides with an interferer angle, i.e., at $\cos\theta_0 = \cos\theta_1 = 0.18$ and $\cos\theta_0 = \cos\theta_2 = 0.73$, so that it is not possible to distinguish the two. Therefore the SNR curves exhibit notches at these angles. For $N > 30$ however, the width of these notches decreases significantly compared to $N = 30$. The improvement can be attributed to the greater angular resolution offered by the longer sparse arrays. In addition, the SNR is also increased by a few dB at angles far from the interferers. This is due to the much larger number of configurations in which the $M$ active elements may be placed. As will be seen in Fig. 4-19, it is often better to choose the non-zero weights to correspond to the largest components in the target array manifold vector rather than a contiguous arrangement.

In Figs. 4-17 and 4-18, we compare the SNR curves for the sparse beamformers of length 40 and 60 against the SNR curves for the non-sparse MVDR beamformers of the same lengths. The sparse beamformers achieve nearly all of the improvement in interference rejection around $\cos\theta_0 = 0.18$ and $\cos\theta_2 = 0.73$ even though they have only three-quarters or one-half as many active elements. This observation supports the hypothesis that the increased interference rejection is due mainly to the increase in length. Fig. 4-17 also shows that the sparse beamformer nearly matches the SNR performance of the non-sparse beamformer at almost all other angles. The same is true to a lesser extent for the length 60 beamformers. The angles at which there is a significant gap are close to $\cos\theta_0 = 0$ and $\cos\theta_0 = 1/2$. As mentioned earlier, the array manifold vectors for these angles have components of equal or nearly equal magnitude, and hence a beamformer with more active elements can collect appreciably more energy from the target direction. In contrast, when the array manifold vector has highly unequal components, minimal loss is incurred by omitting some elements.

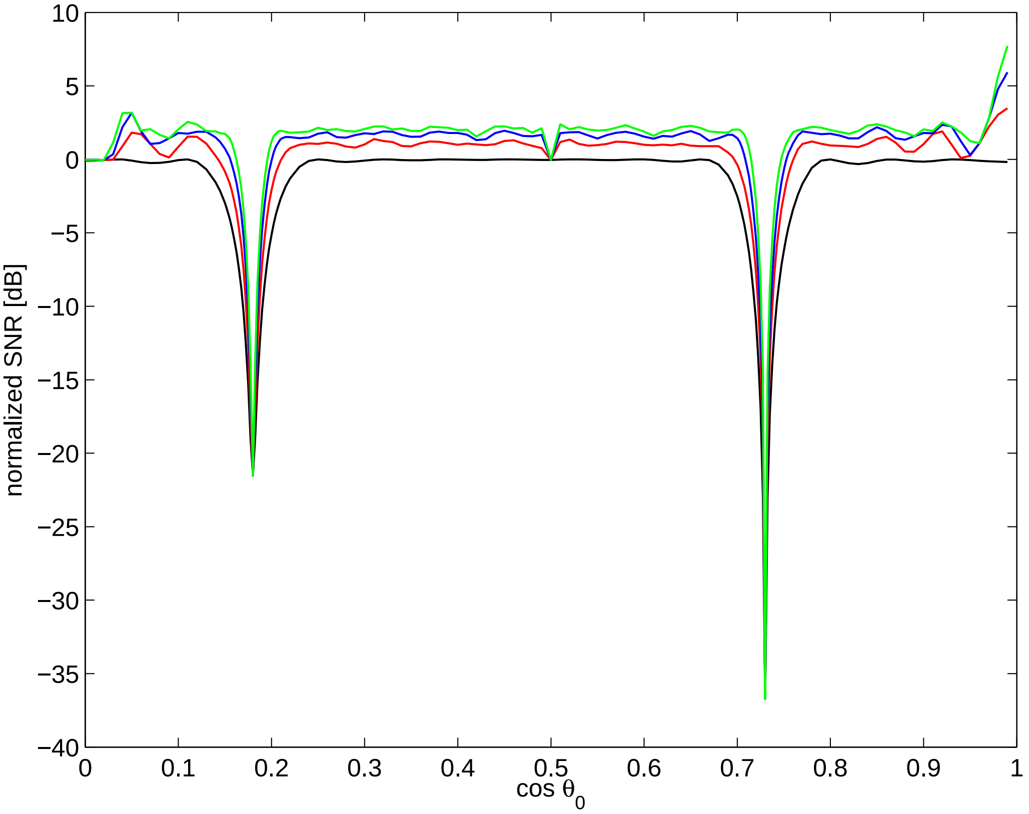


Figure 4-16: Normalized SNR as a function of the target angle $\theta_0$ for beamformers of length $N = 30, 40, 50, 60$ (bottom to top) and $M = 30$ active elements.

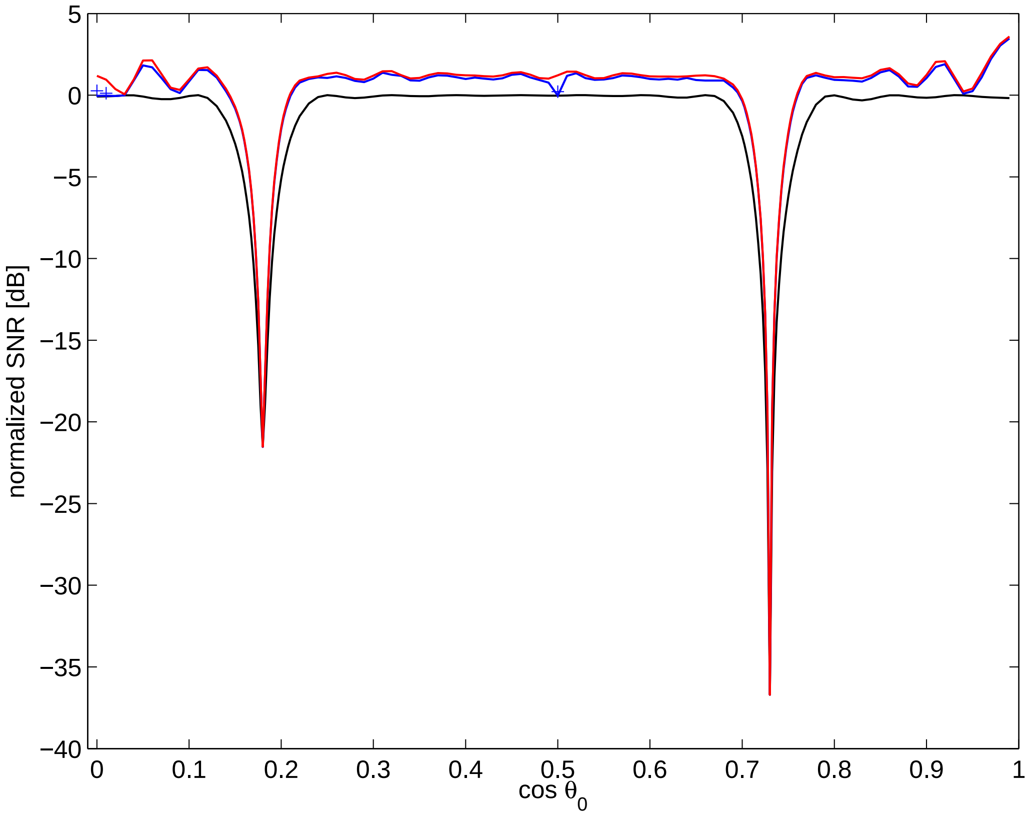


Figure 4-17: Normalized SNR as a function of the target angle $\theta_0$ for non-sparse beamformers of length 30 (bottom, black), sparse beamformers of length 40 (middle, blue), and non-sparse beamformers of length 40 (top, red). For the middle curve, crosses indicate upper bounds on the maximum SNR in cases where the branch-and-bound algorithm did not converge.

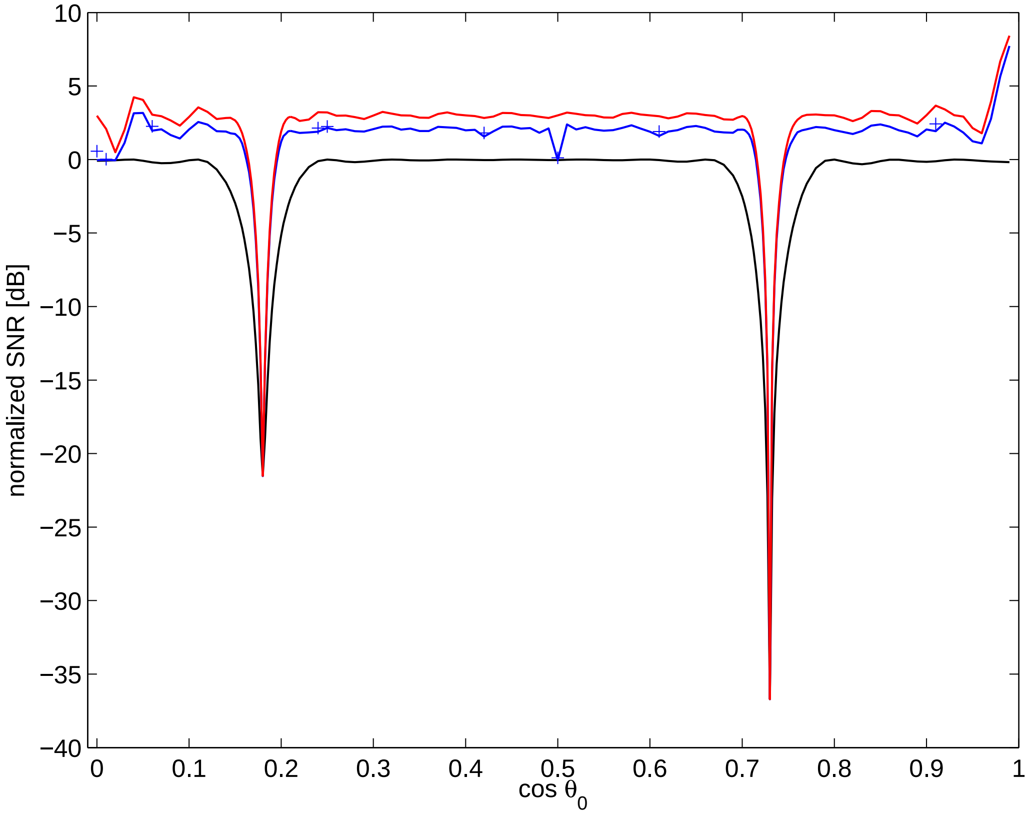


Figure 4-18: Normalized SNR as a function of the target angle $\theta_0$ for non-sparse beamformers of length 30 (bottom, black), sparse beamformers of length 60 (middle, blue), and non-sparse beamformers of length 60 (top, red). For the middle curve, crosses indicate upper bounds on the maximum SNR in cases where the branch-and-bound algorithm did not converge.

Fig. 4-19 compares the beamformer weights of the non-sparse beamformer of length 30, the sparse beamformer of length 50, and the non-sparse beamformer of length 50 at $\cos\theta_0 = 0.195$. Most of the larger weights in the non-sparse length-50 beamformer are located beyond the span of the length-30 beamformer, and these tend to be the positions where the sparse beamformer has non-zero weights. The improvement in SNR for the sparse beamformer over the non-sparse length-30 beamformer is 5.35 dB at this angle.

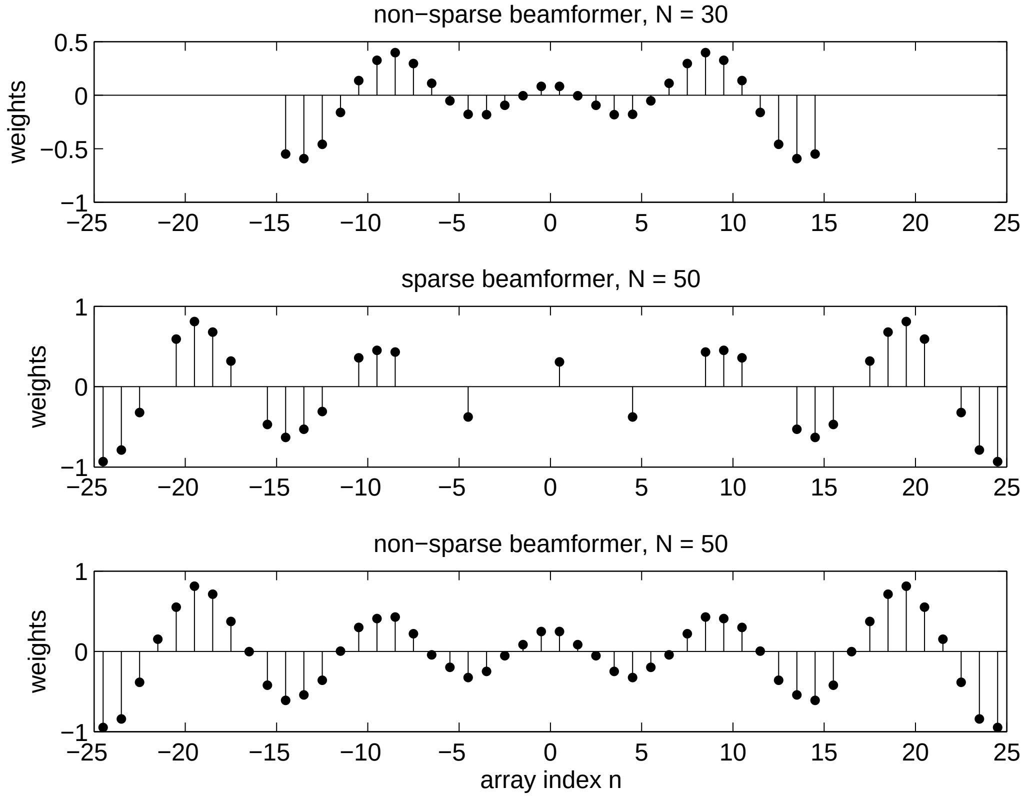


Figure 4-19: Beamformer weights for the non-sparse beamformer of length 30, the sparse beamformer of length 50, and the non-sparse beamformer of length 50 at a target angle of $\cos\theta_0 = 0.195$. Zero-valued weights are omitted.

We note that the covariance structure in (4.2.1) is favorable to the use of diagonal relaxations. The matrix $\mathbf{R}$ has two large eigenvalues contributed by the interference terms while the remaining eigenvalues are small and equal. As observed in Section 3.6, the diagonal relaxation tends to yield a good approximation for this type of eigenvalue distribution. This hypothesis finds further support in the present context. Specifically, it was observed in the beamformer experiments that the ratio between the initial lower bound and the final cost value in the branch-and-bound algorithm typically ranges between 0.9 and 1.

# Chapter 5

# Bit-efficient filter design under a quadratic constraint: Problem formulations and low-complexity algorithms

In Chapters 5–7, the development and results of Chapters 2–4 are extended to measures of complexity involving the number of bits in finite-precision representations of the filter coefficients. As in the first part of the thesis, three filter design problems are considered simultaneously: weighted least-squares frequency response approximation, signal estimation, including prediction and channel equalization, and signal detection. By reducing the performance constraint in each case to the quadratic constraint in (2.1.1), we again obtain a unified framework for solving these design problems.

Two different measures of complexity are considered in Chapters 5–7. In both cases, the total number of coefficients $N$ and the maximum number of bits per coefficient $P$ (i.e., the maximum wordlength) are fixed. The first measure of complexity assumes a conventional sign-magnitude binary representation. Motivated by the fact that arithmetic operations involving larger coefficients tend to be more expensive, we measure the cost of a coefficient by the number of bits excluding leading zeros, i.e., the number of non-leading-zero (NLZ) bits, denoted as $C_{\mathrm{NLZ}}(b_n)$. Additional discussion of the NLZ cost measure can be found in Section 1.1. For the second measure of complexity, a canonic signed digit

(CSD) representation for the coefficients is assumed. The CSD representation is based on signed powers-of-two (SPTs) and is commonly used in multiplier-less implementations in which coefficient multiplications are decomposed into additions, subtractions, and bit shifts. Each non-zero digit corresponds to an addition (+1) or a subtraction (−1), and the CSD representation is distinguished by requiring the fewest non-zero digits among all SPT number representations and by the additional property that no two non-zero digits are adjacent. Accordingly, we measure the complexity by the total number of non-zero digits in all coefficients, which we denote as $C_{\text{SPT}}(\mathbf{b})$. Further discussion of SPT representations can also be found in Section 1.1.

The structure of this chapter follows closely that of Chapter 2. In Section 5.1, we formulate the problems in greater detail, showing in particular how the signal detection problem can be reduced to multiple instances of a problem involving constraint (2.1.1). The choice of quantization step size and the issue of overall scaling are also discussed. As in Chapter 2, we restrict ourselves to low-complexity algorithms in the current chapter. Section 5.2 focuses on special cases in which the matrix $\mathbf{Q}$ is diagonal or block-diagonal, which again permit efficient and exact solution methods. For the general case, a low-complexity heuristic algorithm is proposed in Section 5.3. An exact branch-and-bound algorithm is developed later in Chapter 6.

## 5.1 Problem formulations and reductions

In this section, the filter design problems considered in Chapters 5–7 are formulated in more detail. The problems of least-squares frequency response approximation, estimation, prediction, and channel equalization are discussed together in Section 5.1.1 since they are very similar to those formulated in Sections 2.1.1 and 2.1.2. The main difference is the possible inclusion of an overall scale factor as an additional degree of freedom in the design. The problem of designing filters for signal detection is treated separately in Section 5.1.2, where it is shown to be reducible to multiple instances of a problem with (2.1.1) as the only constraint.

### 5.1.1 Weighted least-squares filter design, estimation, prediction, and equalization

In Section 2.1.1, we formulated the problem of designing a sparse filter to approximate a desired frequency response within an error of $\delta$ as specified in (2.1.5). In the current chapter, we impose the same performance constraint (2.1.5) and change only the cost function to be minimized from the number of non-zero coefficients to either the number of NLZ bits or the number of SPTs. As shown in Section 2.1.1, constraint (2.1.5) can be rewritten in the form of (2.1.1). Similarly, for the estimation, prediction, and channel equalization problems, only the cost function is changed and the constraint on the MSE is reduced to (2.1.1) as in Section 2.1.2. We will focus therefore on the following two problems:

$$\min_{\mathbf{b}} \quad C_{\text{NLZ}}(\mathbf{b}) \quad \text{s.t.} \quad (\mathbf{b}-\mathbf{c})^T\mathbf{Q}(\mathbf{b}-\mathbf{c}) \le \gamma, \quad \mathbf{b} \in \mathbb{Z}^N, \tag{5.1.1}$$

$$\min_{\mathbf{b}} \quad C_{\text{SPT}}(\mathbf{b}) \quad \text{s.t.} \quad (\mathbf{b}-\mathbf{c})^T\mathbf{Q}(\mathbf{b}-\mathbf{c}) \le \gamma, \quad \mathbf{b} \in \mathbb{Z}^N. \tag{5.1.2}$$

It is shown later in this subsection that $\mathbf{b}$ can be restricted to be integer-valued without loss of generality.

For the problems discussed in this subsection, the parameter $\gamma$ represents the amount by which the error $\delta$ exceeds the minimum error achievable with $N$ continuous-valued coefficients. In the present context, the coefficients must be quantized to a maximum wordlength of $P$. Thus the value of $\gamma$ should depend in part on the value of $P$, with $\gamma$ being larger for smaller $P$. At a minimum, $\gamma$ should be large enough to ensure that there is at least one quantized solution of length $N$ and wordlength $P$ that meets the error constraint. In many cases it may be desirable to increase $\gamma$ beyond this minimum threshold to allow for solutions with lower cost.

The quantization step size for the coefficients is determined by the parameter $\mathbf{c}$, which represents the continuous-valued minimum-error design, and the parameters $\mathbf{Q}$ and $\gamma$, which control the size of the feasible set around $\mathbf{c}$. We assume for the moment that the problem parameters are fixed and that the step size is restricted to be a power of two, i.e., the coefficient values are sums of signed powers of two. The smallest power of two, which corresponds to the step size, is determined based on the maximum absolute coefficient values that are feasible under constraint (2.1.1). From (3.3.3), the largest feasible value for

$|b_n|$ subject to (2.1.1) is $\sqrt{\gamma\big(\mathbf{Q}^{-1}\big)_{nn}} + |c_n|$. The largest feasible value for any coefficient, i.e., the maximum $\infty$-norm, is given by $\max \|\mathbf{b}\|_\infty = \max_n \sqrt{\gamma\big(\mathbf{Q}^{-1}\big)_{nn}} + |c_n|$. We choose powers of two such that $\max \|\mathbf{b}\|_\infty$ is smaller than the largest number representable in our chosen system. Given powers $p_0, p_0 + 1, \ldots, p_0 + P - 1$ in a sign-magnitude binary representation, the largest number in absolute value is obtained by setting all bits equal to 1, yielding

$$\sum_{p=p_0}^{p_0+P-1} 2^p = 2^{p_0}(2^P - 1).$$

It follows that the lowest power $p_0$ should be chosen as

$$p_0 = \left\lceil \log_2 \left( \frac{\max \|\mathbf{b}\|_\infty}{2^P - 1} \right) \right\rceil .$$

The situation is slightly different for the CSD representation because of the non-adjacency of non-zero digits. Given powers $p_0, p_0 + 1, \ldots, p_0 + P - 1$ and $P$ even, the largest number is given by

$$2^{p_0+P-1} + 2^{p_0+P-3} + \cdots + 2^{p_0+1} = \frac{2}{3} \cdot 2^{p_0}(2^P - 1),$$

whereas for $P$ odd we have

$$2^{p_0+P-1} + 2^{p_0+P-3} + \cdots + 2^{p_0} = \frac{2}{3} \cdot 2^{p_0} \left( 2^P - \frac{1}{2} \right).$$

Combining the even and odd cases, $p_0$ should be chosen as

$$p_0 = \left\lceil \log_2 \left( \frac{3 \max \|\mathbf{b}\|_\infty}{2(2^P - (1/2)^{P \bmod 2})} \right) \right\rceil .$$

In some situations, there is additional freedom to scale the feasible set by an arbitrary real number. For example, in frequency response approximation, the relative magnitude of the response at different frequencies is often more important than the absolute magnitude. Accordingly, we may consider scaling the desired frequency response $D(e^{j\omega})$ by an arbitrary factor $s$ while also scaling the allowable error $\delta$ by $s^2$. It can be seen from (2.1.6) and the relations $\mathbf{c} = \mathbf{Q}^{-1}\mathbf{f}$ and $\gamma = \beta + \mathbf{c}^T\mathbf{Q}\mathbf{c}$ that this results in $\mathbf{c}$ being scaled by $s$ and $\gamma$ by $s^2$. Equivalently, (2.1.1) is changed to

$$(s^{-1}\mathbf{b} - \mathbf{c})^T \mathbf{Q} (s^{-1}\mathbf{b} - \mathbf{c}) \leq \gamma. \tag{5.1.3}$$

The scaling can also represent quantization with a step size that is not a power of two. In either case, we may continue to assume that the components of $\mathbf{b}$ are restricted to be sums of signed powers of two.

The scale factor $s$ can affect the geometry of the problem, specifically by changing the position of the feasible set corresponding to (5.1.3) relative to the quantization lattice. (It suffices to consider a single octave of values for $s$ since any positive value $s_1$ is related to a value $s_0$ in the chosen octave by a power-of-two factor, and hence $s_0$ and $s_1$ are equivalent in terms of quantization.) This additional degree of freedom has been exploited by several researchers, notably in [53], to reduce the approximation error in the design of finite-precision filters. In the present context, different values for $s$ can lead to different optimal values when the cost functions $C_{\text{NLZ}}(\mathbf{b})$ and $C_{\text{SPT}}(\mathbf{b})$ are minimized subject to (5.1.3). Hence $s$ should be included as an additional variable in the optimization problems.

In the remainder of this chapter, we will assume that when scaling is permitted, it is done in an outer loop in which $s$ is set to different values. We concentrate therefore on solving the problems for a fixed value of $s$. A particularly convenient choice is $s = 2^{-p_0}$, which allows the coefficients to be integer-valued. This results in the substitutions $\mathbf{c} \leftarrow 2^{-p_0}\mathbf{c}$ and $\gamma \leftarrow 2^{-2p_0}\gamma$[1] and effectively resets $p_0$ to 0. In the sequel we will assume that this normalization has been done and that $b_n$ is integer. Under this assumption, the number of NLZ bits can be expressed as

$$C_{\text{NLZ}}(\mathbf{b}) = \sum_{n=1}^{N} \lceil \log_2(1 + |b_n|) \rceil , \qquad (5.1.4)$$

i.e., setting $b_n = 0$ has a cost of 0, $b_n = 1$ has a cost of 1, $b_n = 2, 3$ has a cost of 2, and so on. Thus the cost of a given coefficient value is approximately equal to the base 2 logarithm of the absolute value.

### 5.1.2 Signal detection

We now consider the design of filters for signal detection with the property that either the number of NLZ bits or the number of SPTs is minimized. We assume as in Section 2.1.3 that the output SNR is required to be no smaller than a given threshold $\rho$. Previously in Section 2.1.3, it was shown that when the cost measure is the number of non-zero co-

[1] Equivalently, $\mathbf{Q}$ may be rescaled by $2^{2p_0}$.

efficients, the problem is exactly equivalent to problem (2.0.1). Our proof of equivalence relied however on the assumption that the coefficients are continuous-valued, leading to closed-form solutions to certain optimizations. The previous approach cannot be applied to the present situation in which the coefficients are discrete-valued. We take instead a geometrically-inspired approach to transform the detection problem into multiple instances of the problems (5.1.1) or (5.1.2) formulated in the previous subsection.

We begin by noting that the SNR constraint in (2.1.13) specifies a cone, i.e., a set that contains the points $s\mathbf{b}$ for all $s \geq 0$ whenever $\mathbf{b}$ belongs to the set. Under certain conditions, this cone has sections that are $(N-1)$-dimensional ellipsoids. Specifically, we make the mild assumption that there is an index $n$ such that $b_n$ is positive everywhere inside the cone except at $\mathbf{b} = \mathbf{0}$. In other words, there is at least one coefficient that has a positive value in all designs that satisfy the SNR constraint. As will be shown, it follows from our assumption that the intersections of the cone with the hyperplanes $b_n = v$ for $v > 0$ are finite $(N-1)$-dimensional ellipsoids. The same result holds if $b_n$ is negative inside the cone except at $\mathbf{b} = \mathbf{0}$ and $v < 0$. This phenomenon is illustrated in Fig. 5-1 for the case $N = 3$.

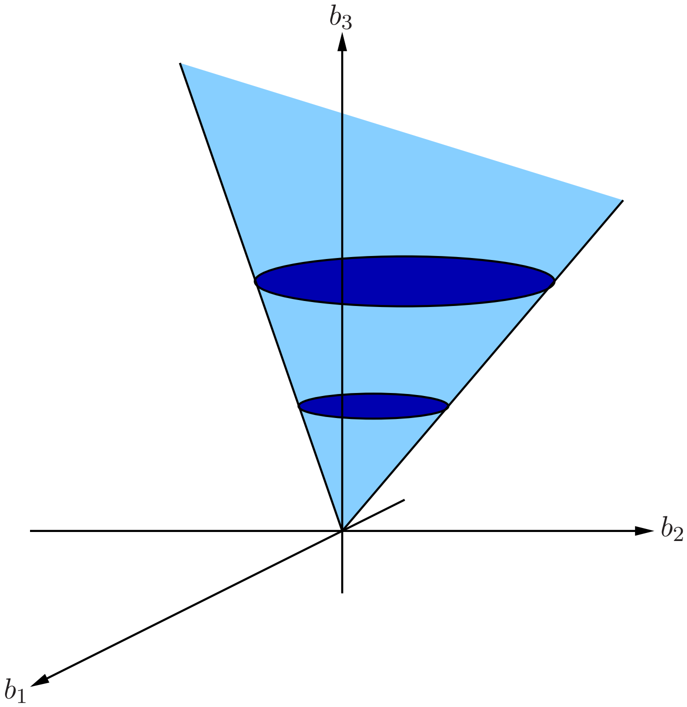


Figure 5-1: A cone of the type corresponding to (2.1.13) for $N = 3$. The component $b_3$ is positive at all points in the cone except the origin. The intersections of the cone with the hyperplanes $b_3 = v$ for $v > 0$ are finite 2-dimensional ellipsoids.

Our approach is to minimize the cost functions $C_{\mathrm{NLZ}}(\mathbf{b})$ and $C_{\mathrm{SPT}}(\mathbf{b})$ over the cone by performing separate minimizations over ellipsoidal sections of the cone. Since the coefficient $b_n$ must be quantized, its possible values form a discrete set. We assume in addition that this set of values is finite. Hence the problem is reduced to solving one $(N-1)$-dimensional instance of (5.1.1) or (5.1.2) for each quantization level for $b_n$.

To prove that the conic sections in question are ellipsoidal, we assume for concreteness that the index $n = N$ is such that $b_N > 0$ for every $\mathbf{b} \neq \mathbf{0}$ satisfying the constraint in (2.1.13). First we derive an inequality describing the conic section obtained by fixing $b_N$ to a positive value. By partitioning the vectors $\mathbf{b}$ and $\mathbf{s}$ and the matrix $\mathbf{R}$ as follows with $\mathcal{F} = \{1, \ldots, N-1\}$:

$$\mathbf{b} = \begin{bmatrix} \mathbf{b}_{\mathcal{F}} \\ b_N \end{bmatrix}, \qquad \mathbf{s} = \begin{bmatrix} \mathbf{s}_{\mathcal{F}} \\ s_N \end{bmatrix}, \qquad \mathbf{R} = \begin{bmatrix} \mathbf{R}_{\mathcal{F}\mathcal{F}} & \mathbf{R}_{\mathcal{F}N} \\ \mathbf{R}_{N\mathcal{F}} & R_{NN} \end{bmatrix},$$

the constraint in (2.1.13) may be rewritten as

$$\rho \left( \begin{bmatrix} \mathbf{b}_{\mathcal{F}}^T & b_N \end{bmatrix} \begin{bmatrix} \mathbf{R}_{\mathcal{F}\mathcal{F}} & \mathbf{R}_{\mathcal{F}N} \\ \mathbf{R}_{N\mathcal{F}} & R_{NN} \end{bmatrix} \begin{bmatrix} \mathbf{b}_{\mathcal{F}} \\ b_N \end{bmatrix} \right)^{1/2} \leq \mathbf{s}_{\mathcal{F}}^T \mathbf{b}_{\mathcal{F}} + s_N b_N. \tag{5.1.5}$$

Since both sides are non-negative, we may square them without changing the inequality. After rearranging terms, we obtain

$$\mathbf{b}_{\mathcal{F}}^T (\rho^2 \mathbf{R}_{\mathcal{F}\mathcal{F}} - \mathbf{s}_{\mathcal{F}} \mathbf{s}_{\mathcal{F}}^T) \mathbf{b}_{\mathcal{F}} - 2 b_N (s_N \mathbf{s}_{\mathcal{F}} - \rho^2 \mathbf{R}_{\mathcal{F}N})^T \mathbf{b}_{\mathcal{F}} \leq b_N^2 (s_N^2 - \rho^2 R_{NN}), \tag{5.1.6}$$

which is of the same form as (2.1.3), the alternative form of (2.1.1), when we regard $b_N$ as being fixed. To show that (5.1.6) corresponds to a finite $(N-1)$-dimensional ellipsoid, it suffices to show that the matrix $\rho^2 \mathbf{R}_{\mathcal{F}\mathcal{F}} - \mathbf{s}_{\mathcal{F}} \mathbf{s}_{\mathcal{F}}^T$ is positive definite. Equation (5.1.6) specifies a non-empty set as long as the original cone is non-empty.

To prove that $\rho^2 \mathbf{R}_{\mathcal{F}\mathcal{F}} - \mathbf{s}_{\mathcal{F}} \mathbf{s}_{\mathcal{F}}^T$ is positive definite, we return to our assumption that $b_N > 0$ whenever $\mathbf{b} = (\mathbf{b}_{\mathcal{F}}, b_N)$ satisfies constraint (5.1.5) and $\mathbf{b}$ is not equal to $\mathbf{0}$. The assumption implies that if $b_N = 0$ and $\mathbf{b}_{\mathcal{F}} \neq \mathbf{0}$, (5.1.5) must be violated, i.e.,

$$\rho \sqrt{\mathbf{b}_{\mathcal{F}}^T \mathbf{R}_{\mathcal{F}\mathcal{F}} \mathbf{b}_{\mathcal{F}}} > \mathbf{s}_{\mathcal{F}}^T \mathbf{b}_{\mathcal{F}}. \tag{5.1.7}$$

We restrict our attention for the moment to those non-zero vectors $\mathbf{b}_{\mathcal{F}}$ satisfying $\mathbf{s}_{\mathcal{F}}^T\mathbf{b}_{\mathcal{F}} \geq 0$ so that the right-hand side of (5.1.7) is non-negative. Squaring both sides of (5.1.7) and rearranging, we have

$$\mathbf{b}_{\mathcal{F}}^T(\rho^2\mathbf{R}_{\mathcal{F}\mathcal{F}} - \mathbf{s}_{\mathcal{F}}\mathbf{s}_{\mathcal{F}}^T)\mathbf{b}_{\mathcal{F}} > 0 \quad \forall\, \mathbf{b}_{\mathcal{F}} \neq \mathbf{0} : \mathbf{s}_{\mathcal{F}}^T\mathbf{b}_{\mathcal{F}} \geq 0, \tag{5.1.8}$$

which shows that $\rho^2\mathbf{R}_{\mathcal{F}\mathcal{F}} - \mathbf{s}_{\mathcal{F}}\mathbf{s}_{\mathcal{F}}^T$ is positive definite over the half-space $\mathbf{s}_{\mathcal{F}}^T\mathbf{b}_{\mathcal{F}} \geq 0$. This is a sufficient condition for $\rho^2\mathbf{R}_{\mathcal{F}\mathcal{F}} - \mathbf{s}_{\mathcal{F}}\mathbf{s}_{\mathcal{F}}^T$ to be positive definite over all of $\mathbb{R}^{N-1}$ since every vector $\mathbf{b}_{\mathcal{F}}$ such that $\mathbf{s}_{\mathcal{F}}^T\mathbf{b}_{\mathcal{F}} < 0$ is the negative of some vector for which $\mathbf{s}_{\mathcal{F}}^T\mathbf{b}_{\mathcal{F}} > 0$, and (5.1.8) is equally true for $-\mathbf{b}_{\mathcal{F}}$ as it is for $\mathbf{b}_{\mathcal{F}}$. Hence the additional qualification $\mathbf{s}_{\mathcal{F}}^T\mathbf{b}_{\mathcal{F}} \geq 0$ in (5.1.8) can be removed.

We conclude that under the assumption made in this subsection, the conic sections specified by (5.1.6) are finite $(N-1)$-dimensional ellipsoids. The equivalent parameters in (2.1.3) are given by

$$\mathbf{Q} = \rho^2\mathbf{R}_{\mathcal{F}\mathcal{F}} - \mathbf{s}_{\mathcal{F}}\mathbf{s}_{\mathcal{F}}^T, \quad \mathbf{f} = b_N(s_N\mathbf{s}_{\mathcal{F}} - \rho^2\mathbf{R}_{\mathcal{F}N}), \quad \beta = b_N^2(s_N^2 - \rho^2 R_{NN}).$$

The values of $\mathbf{f}$ and $\beta$ depend linearly and quadratically on $b_N$. The same dependence was observed in Section 5.1.1 when we considered scaling the desired frequency response $D(e^{j\omega})$ by a factor $s$ (see (2.1.6)). Quantizing $b_N$ to different values is thus analogous to varying the scale factor $s$.

## 5.2 Special cases

In Section 5.1, we showed that the problems addressed in this chapter can all be reduced to problems (5.1.1) and (5.1.2). Generally speaking, (5.1.1) and (5.1.2) are considered to be even more difficult than the sparsity maximization problem (2.0.1) because the variables are required to be discrete-valued. As in Section 2.2 however, we can identify some special cases for which exact algorithms are also efficient. In Sections 5.2.1 and 5.2.2, we discuss the solution of (5.1.1) and (5.1.2) in the case where the matrix $\mathbf{Q}$ is diagonal. We extend the algorithms to the block-diagonal case in Section 5.2.3 and to separable non-quadratic constraint functions in Section 5.2.4.

In analogy with Section 2.2, the methods in the current section solve (5.1.1) or (5.1.2) by determining for each $B = 0, 1, 2, \ldots$ whether a feasible solution requiring $B$ NLZ bits or $B$ SPTs exists. Such a solution exists only if (2.1.1) is satisfied when the left-hand side is minimized over all integer-valued $\mathbf{b}$ of cost $C_{\text{NLZ}}(\mathbf{b}) = B$ or $C_{\text{SPT}}(\mathbf{b}) = B$, since (2.1.1) cannot be satisfied for any $\mathbf{b}$ with the specified cost if it is not satisfied for the minimizer. We define $E_{\text{NLZ}}(B)$ to be the minimum value of the left-hand side of (2.1.1) under a total cost constraint of $B$ NLZ bits, and similarly for $E_{\text{SPT}}(B)$. We refer to $E_{\text{NLZ}}(B)$ and $E_{\text{SPT}}(B)$ as the minimum quantization error since $E_{\text{NLZ}}(B) = E_{\text{SPT}}(B) = 0$ in the limit $B \to \infty$, i.e., with continuous values, and the increase from zero is due to the quantization of $\mathbf{b}$. In the general case, $E_{\text{NLZ}}(B)$ and $E_{\text{SPT}}(B)$ are very difficult to compute. In the diagonal and block-diagonal cases however, efficient algorithms are known. Unlike in Section 2.2, an efficient method for the case of banded $\mathbf{Q}$ has not been found to date.

### 5.2.1 Diagonal Q, NLZ cost function

We start with the solution of (5.1.1) in the case of diagonal $\mathbf{Q}$. First we present a dynamic programming algorithm for computing values of $E_{\text{NLZ}}(B)$, the minimum quantization error given a total of $B$ NLZ bits. Later we indicate how the algorithm can be simplified in the context of solving (5.1.1).

When $\mathbf{Q}$ is diagonal, $E_{\text{NLZ}}(B)$ takes the following form:

$$\begin{aligned} E_{\text{NLZ}}(B) = \min \quad & \sum_{n=1}^{N} Q_{nn}(b_n - c_n)^2 \\ \text{s.t.} \quad & \sum_{n=1}^{N} \lceil \log_2(1 + |b_n|) \rceil = B, \\ & b_n \in \mathbb{Z} \quad \forall\, n, \end{aligned} \tag{5.2.1}$$

using the expression in (5.1.4) for $C_{\text{NLZ}}(\mathbf{b})$. The optimization in (5.2.1) bears some resemblance to the one in (2.2.9) and can be solved using a similar dynamic programming approach. For $m = 1, 2, \ldots, N$, we define $V_m(B)$ to be the minimum quantization error given $B$ bits for the first $m$ coefficients, i.e.,

$$\begin{aligned} V_m(B) = \min \quad & \sum_{n=1}^{m} Q_{nn}(b_n - c_n)^2 \\ \text{s.t.} \quad & \sum_{n=1}^{m} \lceil \log_2(1 + |b_n|) \rceil = B, \\ & b_n \in \mathbb{Z} \quad \forall\, n. \end{aligned} \tag{5.2.2}$$

Hence $E_{\text{NLZ}}(B) = V_N(B)$. We also define $v_n(B)$ for $n = 1, \dots, N$ to be the minimum quantization error given $B$ bits for the $n$th coefficient,

$$\begin{aligned} v_n(B) = \min \quad & Q_{nn}(b_n - c_n)^2 \\ \text{s.t.} \quad & \lceil \log_2(1 + |b_n|) \rceil = B, \\ & b_n \in \mathbb{Z}. \end{aligned} \tag{5.2.3}$$

Comparing (5.2.2) and (5.2.3), we have $V_1(B) = v_1(B)$. For $m$ higher than 1, we may compute $V_m(B)$ through the following recursion, in analogy with (2.2.12):

$$V_m(B) = \min_{B'=0,1,\dots,\min\{B,P\}} \left\{ v_m(B') + V_{m-1}(B - B') \right\}. \tag{5.2.4}$$

The upper bound of $P$ on $B'$ reflects the fact that we cannot allocate more than $P$ bits to a single coefficient.

The minimization in (5.2.3) has a straightforward solution: If $|c_n|$ is greater than $2^B - 1$, the maximum absolute value achievable with $B$ NLZ bits, then the best solution is to set $b_n = \text{sgn}(c_n)(2^B - 1)$. Otherwise, $b_n$ is equal to $c_n$ rounded to the nearest integer, denoted as $[c_n]$. Thus

$$v_n(B) = \begin{cases} Q_{nn}(|c_n| - (2^B - 1))^2, & |c_n| > 2^B - 1, \\ Q_{nn}(c_n - [c_n])^2, & |c_n| \le 2^B - 1. \end{cases} \tag{5.2.5}$$

The minimum quantization error $E_{\text{NLZ}}(B)$ can be computed using (5.2.4) and (5.2.5). It can be seen that to determine $E_{\text{NLZ}}(B)$ for a given value of $B$, it is necessary to perform most of the intermediate computations for determining $E_{\text{NLZ}}(B')$ for $B' < B$ because we require the values of $V_{N-1}(B')$ for $B' < B$ in (5.2.4). However, in the context of solving (5.1.1), we often do not need to carry out the full recursion. The optimal value of (5.1.1) is equal to the smallest value of $B$ such that $E_{\text{NLZ}}(B) \leq \gamma$. Since we are only interested in determining whether $E_{\text{NLZ}}(B)$ exceeds a threshold, we may terminate the recursion as soon as it is known that $E_{\text{NLZ}}(B) > \gamma$ and proceed to evaluate $E_{\text{NLZ}}(B+1)$. More specifically, if $V_m(B) > \gamma$ for some $m < N$, we may conclude that $V_{m'}(B) > \gamma$ for all $m' > m$, and in particular, $V_N(B) = E_{\text{NLZ}}(B) > \gamma$. To see this, first note that the condition $V_m(B) > \gamma$ implies that $V_m(B') > \gamma$ for all $B' < B$ since the quantization error cannot decrease with fewer bits. From (5.2.4), it follows that $V_{m+1}(B) \geq V_m(B')$ for all $B' \leq B$, and hence $V_{m+1}(B) > \gamma$. Applying induction yields the same result for higher values of $m' > m$.

In light of the previous observation, we now describe a simplified method for solving (5.1.1). We start with $B = 0$ and evaluate $V_m(0)$ for increasing $m$ using (5.2.4) and (5.2.5) until either $V_N(0) = E_{\text{NLZ}}(0) \leq \gamma$, at which point we are done, or $V_m(0) > \gamma$ for some $m < N$. In the latter case, we proceed with evaluating $V_m(B)$ for $B = 1$. Continuing in this manner and assuming that at least one feasible solution exists for the given wordlength $P$, we eventually terminate with the condition $V_N(B) = E_{\text{NLZ}}(B) \leq \gamma$ for some $B$ and conclude that $B$ is the optimal value of (5.1.1). It is also possible to further simplify the minimization in (5.2.4). Since the recursion terminates as soon as $V_m(B) > \gamma$, it suffices to minimize over those values of $B'$ for which $V_{m-1}(B - B') \leq \gamma$. Therefore (5.2.4) is changed to

$$V_m(B) = \min_{\substack{B'=0,1,\ldots,\min\{B,P\} \\ V_{m-1}(B-B') \leq \gamma}} \left\{ v_m(B') + V_{m-1}(B - B') \right\}. \tag{5.2.6}$$

If a solution $\mathbf{b}$ that achieves the optimal value of (5.1.1) is also desired, it can be obtained through a backtracking procedure. The algorithm proceeds as before, except that each time (5.2.6) is evaluated, the number of bits $B'$ that minimizes the right-hand side is recorded. When the algorithm terminates with the condition $V_N(B) \leq \gamma$, the minimizer $B'$ in (5.2.6) corresponding to $V_N(B)$ becomes the number of bits allocated to the $N$th coefficient. Then the minimizer $B''$ corresponding to $V_{N-1}(B - B')$ is the number of bits allocated to the $(N-1)$th coefficient. We continue in this way until bits have been allocated to all coefficients. The values of $b_n$ can then be determined from these allocations as discussed in the paragraph preceding (5.2.5).

A worst-case estimate of the computational complexity is as follows: The recursion in (5.2.6) may need to be carried out for $m = 2, \ldots, N$ and $B = 0, 1, \ldots, NP$, where $NP$ represents the maximum number of bits available. Each evaluation of (5.2.6) requires at most $P+1$ additions and comparisons. Therefore the total number of operations is proportional to $(N-1)(NP+1)(P+1) \sim \mathcal{O}((NP)^2)$, which is only quadratic in $NP$ with a leading coefficient of unity. In addition, $v_n(B)$ may need to be computed for $n = 1, \ldots, N$ and $B = 0, \ldots, P$ in the worst case, a total of $N(P+1)$ evaluations of (5.2.5).

### 5.2.2 Diagonal Q, SPT cost function

We now continue with the assumption that $\mathbf{Q}$ is diagonal and discuss the solution of problem (5.1.2). While it is possible to use the same algorithm as in Section 5.2.1, there is an even more efficient alternative. In [70], Llorens et al. propose a provably optimal greedy algorithm for determining $E_{\text{SPT}}(B)$, the minimum quantization error given $B$ SPTs, in the case $\mathbf{Q} = \mathbf{I}$. For completeness, we review the algorithm of [70], which applies equally well to the general diagonal case, and we present an alternative, more intuitive explanation for the optimality of the algorithm.

First we introduce the concept of incremental error decreases that is fundamental to the algorithm. We define $u_n(B)$ to be the minimum quantization error given $B$ SPTs for the $n$th coefficient, i.e.,

$$\begin{aligned} u_n(B) = \min \quad & Q_{nn}(b_n - c_n)^2 \\ \text{s.t.} \quad & C_{\text{SPT}}(b_n) = B, \\ & b_n \in \mathbb{Z}. \end{aligned} \tag{5.2.7}$$

The number of SPTs can range from 0 to $\lceil P/2 \rceil$, the maximum number given a wordlength of $P$ and the non-adjacency constraint. The solution to the quantization problem in (5.2.7) is described in Appendix C.1. The error decrease due to the allocation of the $B$th SPT to the $n$th coefficient is defined as $\Delta u_n(B) = u_n(B-1) - u_n(B)$. For $B > \lceil P/2 \rceil$, we set $\Delta u_n(B) = 0$ since no further allocations are possible. Llorens et al. prove in [70] that the incremental error decreases for CSD quantization are monotonic,

$$\Delta u_n(B) \geq \Delta u_n(B+1), \quad B = 1, 2, \ldots. \tag{5.2.8}$$

In contrast, this property does not hold for the cost function $C_{\text{NLZ}}(b_n)$. Consider an example where $Q_{nn} = 1$ and $c_n = 4$. From (5.2.5), we have $v_n(0) = 16$, $v_n(1) = 9$, $v_n(2) = 1$, and $v_n(3) = 0$, yielding $\Delta v_n(1) = 7$, $\Delta v_n(2) = 8$, and $\Delta v_n(3) = 1$.

We now explain why the monotonicity property in (5.2.8) allows $E_{\text{SPT}}(B)$ to be computed using a greedy method. The optimization problem to be solved is

$$\begin{aligned} E_{\text{SPT}}(B) = \min \quad & \sum_{n=1}^{N} Q_{nn}(b_n - c_n)^2 \\ \text{s.t.} \quad & \sum_{n=1}^{N} C_{\text{SPT}}(b_n) = B, \\ & b_n \in \mathbb{Z} \quad \forall\, n. \end{aligned} \tag{5.2.9}$$

For $B = 0$, the only possible value for $\mathbf{b}$ is $\mathbf{0}$ and hence $E_{\text{SPT}}(0) = \sum_n Q_{nn}c_n^2$. For $B = 1$, the quantization error is guaranteed to be minimized if the SPT corresponding to the largest error decrease $\Delta u_n(B)$ is added. Because of property (5.2.8), the largest $\Delta u_n(B)$ can be found among $\Delta u_1(1), \Delta u_2(1), \ldots, \Delta u_N(1)$. Note that if (5.2.8) does not hold, the largest $\Delta u_n(B)$ may not occur with $B = 1$ and consequently the largest error decrease may not be available at the current stage. Suppose now that $\Delta u_1(1)$ is the largest. We allocate the first SPT to the first coefficient and subtract $\Delta u_1(1)$ from $E_{\text{SPT}}(0)$ to obtain $E_{\text{SPT}}(1)$. For $B = 2$, the quantization error is again minimized if the SPT corresponding to the second largest $\Delta u_n(B)$ is added. This can be done by selecting the largest among $\Delta u_1(2), \Delta u_2(1), \Delta u_3(1), \ldots, \Delta u_N(1)$. Thus it is possible at each step to add the SPT that results in the next largest error decrease, ensuring continued optimality. In general, given a current allocation of $a_1, a_2, \ldots, a_N$ SPTs to the coefficients, the coefficient $b_m$ to which the next SPT should be allocated is determined according to

$$m = \arg\max_n \Delta u_n(a_n + 1). \tag{5.2.10}$$

We then increment $a_m$ by 1 and subtract $\Delta u_m(a_m+1)$ from the current error $E_{\text{SPT}}(\sum_n a_n)$ to determine the next error $E_{\text{SPT}}(1 + \sum_n a_n)$.

The greedy algorithm above computes $E_{\text{SPT}}(B)$ for increasing values of $B$. The optimal value of (5.1.2) is equal to the first value of $B$ for which $E_{\text{SPT}}(B) \leq \gamma$. Once the optimal value is determined, the optimal allocation of bits to coefficients is also known. From the optimal allocation and using the method of Appendix C.1, we can determine values for $b_n$ that achieve the optimal value.

The computational complexity of the greedy algorithm is lower than that of the dynamic programming algorithm in Section 5.2.1. There are at most $N\lceil P/2\rceil$ values of $u_n(B)$ to be computed since $n$ ranges from 1 to $N$ and $B$ from 1 to $\lceil P/2\rceil$. As discussed in Appendix C.1, each evaluation of $u_n(B)$ requires $\mathcal{O}(\lceil P/2\rceil)$ operations. In the greedy algorithm itself, at most $N\lceil P/2\rceil$ SPTs can be added, and approximately $N$ comparisons are needed to determine the coefficient to which each new SPT should be assigned. Hence the total number of operations is $\mathcal{O}(N\lceil P/2\rceil(N+\lceil P/2\rceil))$ compared to $\mathcal{O}((NP)^2)$ for the dynamic programming algorithm.

### 5.2.3 Block-diagonal Q

In this subsection, we indicate briefly how the dynamic programming algorithm of Section 5.2.1 can be extended to the case of block-diagonal $\mathbf{Q}$. This extension can be equivalently regarded as a generalization of the algorithm of Section 2.2.2.

We assume that $\mathbf{Q}$ has the block-diagonal structure shown in (2.2.8). It follows that the minimum quantization error $E_{\text{NLZ}}(B)$ given a total of $B$ NLZ bits can be expressed as

$$\begin{aligned} E_{\text{NLZ}}(B) = \min \quad & \sum_{b=1}^{L} (\mathbf{b}_b - \mathbf{c}_b)^T \mathbf{Q}_b (\mathbf{b}_b - \mathbf{c}_b) \\ \text{s.t.} \quad & \sum_{b=1}^{L} C_{\text{NLZ}}(\mathbf{b}_b) = B, \\ & \mathbf{b}_b \in \mathbb{Z}^{N_b} \quad \forall\, b, \end{aligned}$$

where the vectors $\mathbf{b}_b$ and $\mathbf{c}_b$ correspond to the $b$th block of $\mathbf{b}$ and $\mathbf{c}$. A similar expression holds for the cost measure $C_{\text{SPT}}$, and the following discussion applies to both metrics. In analogy with Section 2.2.2, we define $V_g(B)$ to be the minimum quantization error over the first $g$ blocks,

$$\begin{aligned} V_g(B) = \min \quad & \sum_{b=1}^{g} (\mathbf{b}_b - \mathbf{c}_b)^T \mathbf{Q}_b (\mathbf{b}_b - \mathbf{c}_b) \\ \text{s.t.} \quad & \sum_{b=1}^{g} C_{\text{NLZ}}(\mathbf{b}_b) = B, \\ & \mathbf{b}_b \in \mathbb{Z}^{N_b} \quad \forall\, b, \end{aligned}$$

and $v_b(B)$ to be the minimum quantization error for the $b$th block alone,

$$\begin{aligned} v_b(B) = \min \quad & (\mathbf{b}_b - \mathbf{c}_b)^T \mathbf{Q}_b (\mathbf{b}_b - \mathbf{c}_b) \\ \text{s.t.} \quad & C_{\text{NLZ}}(\mathbf{b}_b) = B, \\ & \mathbf{b}_b \in \mathbb{Z}^{N_b}. \end{aligned} \tag{5.2.11}$$

Then $V_1(B) = v_1(B)$ and for $g = 2, \ldots, L$, we have the recursion

$$V_g(B) = \min_{B'=0,1,\ldots,\min\{B, N_g P\}} \left\{ v_g(B') + V_{g-1}(B - B') \right\}, \tag{5.2.12}$$

where $N_g P$ is the largest number of bits that can be allocated to the $g$th block. It is also possible to incorporate the simplifications discussed in Section 5.2.1 since the property that $V_{g'}(B) > \gamma$ for all $g' > g$ whenever $V_g(B) > \gamma$ is satisfied in the present case as well. Accordingly we may start from $B = 0$ and compute $V_g(B)$ for increasing values of $g$ until either $V_L(B) \leq \gamma$ or $V_g(B) > \gamma$ for $g < L$, continuing in the second case with the next value of $B$. The first value of $B$ for which $V_L(B) \leq \gamma$ is the optimal value of (5.1.1). Furthermore, the minimization in (5.2.12) may be restricted to those $B'$ for which $V_{g-1}(B - B') \leq \gamma$.

The key difference compared to the diagonal case is the increased complexity of the intra-block minimizations in (5.2.11). Assuming that the submatrices $\mathbf{Q}_b$ do not have further structure, (5.2.11) is a lower-dimensional instance of the original problem of computing $E_{\text{NLZ}}(B)$ in the general case of unstructured $\mathbf{Q}$. Therefore the efficiency of the dynamic programming method is limited by the complexity of solving (5.2.11). In the ideal case, the block dimensions $N_b$ should all be small integers.

### 5.2.4 Generalization to separable non-quadratic constraint functions

The methods in Sections 5.2.1 and 5.2.3 can be generalized to separable non-quadratic constraint functions, similar to the situation in Section 2.2. The dynamic programming algorithm applies equally well to a constraint function that can be separated into a sum of univariate functions $F_n(b_n)$. The case of diagonal $\mathbf{Q}$ corresponds to $F_n(b_n) = Q_{nn}(b_n - c_n)^2$. Similarly, the block-diagonal case can be generalized to block-separable constraint functions.

## 5.3 Low-complexity algorithm for the general case

In the remainder of this chapter and in Chapter 6, we focus on the general case in which $\mathbf{Q}$ is not diagonal or block-diagonal. Given the difficulty of solving problems (5.1.1) and (5.1.2) exactly, we discuss first an algorithm that performs approximate minimization of the two cost functions and requires relatively little computation. The solutions resulting from this algorithm can either be used directly or as a starting point toward the determination of an optimal solution as discussed in Chapter 6.

The algorithm of this section produces a sequence of feasible solutions with gradually decreasing cost (number of NLZ bits or SPTs), using a simplified procedure described in Section 5.3.1 to search among cost allocations with a total cost one unit lower than that of the previous solution. In this respect, the algorithm is similar to the successive thinning algorithm of Section 2.3. The difference is that in sparse filter design with continuous-valued coefficients, once the subset of zero-valued coefficients has been fixed, the values of the non-zero coefficients that minimize the error (i.e., the left-hand side of (2.1.1)) can be determined analytically. In contrast, when the coefficients are discrete-valued, minimizing the error is a difficult integer optimization problem even in the analogous situation of a fixed cost allocation. Hence a simplified algorithm is also required to perform error minimization for a fixed allocation. We present such an algorithm in Section 5.3.2.

### 5.3.1 Cost reduction strategy

We focus in this section on problem (5.1.1); the algorithm for (5.1.2) is similar. We begin by discussing a strategy for iteratively reducing the cost of a feasible solution. Recall from Section 5.2 that (5.1.1) may be solved by determining the smallest value of $B$ such that $E_{NLZ}(B) \leq \gamma$, where $E_{\mathrm{NLZ}}(B)$ is the minimum value of the left-hand side of (2.1.1), i.e., the minimum quantization error, given a total cost of $B$ bits. Evaluating $E_{\mathrm{NLZ}}(B)$ exactly is difficult in part because of the large number of ways in which a total of $B$ bits can be allocated to $N$ coefficients. To be more precise, the number of allocations is given by $\binom{B+N-1}{B}$, which grows rapidly as a function of $B$ and $N$. The method presented in this subsection restricts the search space dramatically. Given a current allocation $(a_1, \ldots, a_N)$ of bits to coefficients, to reduce the cost by one bit we search over only those allocations that differ from the current allocation in one position, e.g. $(a_1 - 1, a_2, \ldots, a_N)$. The allocation in this restricted set for which the quantization error is the lowest is chosen for the next iteration. The problem of minimizing the quantization error for a fixed allocation is treated in Section 5.3.2. We use $\widehat{E}_{\text{NLZ}}(B)$ to denote the quantization error corresponding to the chosen allocation for $B$ bits, which is an approximation to the true minimum $E_{\text{NLZ}}(B)$. The cost reduction procedure terminates when $\widehat{E}_{\text{NLZ}}(B+1) > \gamma$ for some $B$, at which point the last feasible allocation with $B$ bits is taken to be the final allocation.

The search strategy just described greatly reduces the number of allocations considered since at most $N$ different allocations are explored in every iteration. The number of iterations can be no more than $NP$ ($N\lceil P/2\rceil$ for the CSD case), the maximum number of bits available. One special case in which the simplified search does not result in a loss of optimality is the case of CSD quantization under diagonal $\mathbf{Q}$ that was discussed in Section 5.2.2. As shown in that section, the optimal allocations corresponding to $E_{\text{SPT}}(B)$ and $E_{\text{SPT}}(B-1)$ do have the property of differing in only one position, and hence the simplified search can determine one of the allocations from the other. In other cases however, we do not expect the search to guarantee optimality.

In more detail, the algorithm begins by obtaining a feasible solution $\mathbf{b}$ given a maximal bit allocation, i.e., $(P, P, \ldots, P)$ or $(\lceil P/2\rceil, \ldots, \lceil P/2\rceil)$ depending on the cost measure. We use Algorithm 4 to be described in Section 5.3.2 to approximately minimize the quantization error for this initial allocation. Only the actual numbers of bits $(a_1, \ldots, a_N)$ required by the initial solution are retained for the next iteration. We set $\widehat{E}_{\text{NLZ}}\left(\sum_n a_n\right)$ to the initial error value, which is assumed to be below $\gamma$. We also define $\mathcal{C}$ to be the list of indices $n$ that are candidates for further reductions in the number of bits $a_n$. Initially, $\mathcal{C}$ includes all $n$ such that $a_n > 0$.

In a typical iteration, we start with an existing allocation $\mathbf{a}_0 = (a_1, \ldots, a_N)$ and create for each $n \in \mathcal{C}$ a trial allocation $\mathbf{a}_n$ that has one fewer bit than $\mathbf{a}_0$ in the $n$th position. We then approximately minimize the quantization error for each trial allocation using Algorithm 4. The lowest of the errors $\widetilde{E}_n$ returned by Algorithm 4 becomes the value of $\widehat{E}_{\text{NLZ}}\left(\left(\sum_n a_n\right) - 1\right)$. If $\widehat{E}_{\text{NLZ}}\left(\left(\sum_n a_n\right) - 1\right) > \gamma$, the algorithm terminates and returns $\mathbf{a}_0$ as the final allocation. Otherwise, we set $\mathbf{a}_0$ equal to the allocation $\mathbf{a}_n$ that minimizes $\widetilde{E}_n$ and we update the feasible solution $\mathbf{b}$ accordingly.

Before continuing with the next iteration, we remove from $\mathcal{C}$ those indices for which the number of bits $a_n$ has decreased to zero. In addition, indices for which the quantization error

$\widetilde{E}_n > \gamma$ are also eliminated. This second group of indices is removed because Algorithm 4 was unable to find a feasible solution using fewer than $a_n$ bits for the $n$th coefficient. The removal is analogous to the reassignment of indices from $\mathcal{F}$ to $\mathcal{U}$ in Section 2.3 to account for coefficients that can no longer be set to zero and still yield feasible solutions. Note however that since $\widetilde{E}_n$ is only an approximation to the true minimum quantization error, the condition $\widetilde{E}_n > \gamma$ does not guarantee that no feasible solutions exist with fewer than $a_n$ bits for the $n$th coefficient, and the elimination of indices based on this condition is intended only as a further restriction of the search space.

Algorithm 3 summarizes the cost reduction procedure of this subsection.

**Algorithm 3** Heuristic cost reduction algorithm

```
Input: Parameters $\mathbf{Q}$, $\mathbf{c}$, $\gamma$, wordlength $P$.
Output: Feasible solution $\mathbf{b}$ to (5.1.1) or (5.1.2) with cost $B$.
  Initialize: Use Algorithm 4 to approximately minimize quantization error given maximal
  allocation. Set $\mathbf{a}_0 = (a_1, \dots, a_N)$ where $(a_1, \dots, a_N)$ are the numbers of bits required by
  the initial solution, $B = \sum_n a_n$, $\widehat{E}_{\text{NLZ}}(B)$ = initial error value, and $\mathcal{C} = \{n : a_n > 0\}$.
  repeat
    if not first iteration then
      $\mathbf{a}_0 = \mathbf{a}_m$ where $m = \arg\min_{n \in \mathcal{C}} \widetilde{E}_n$.
      $B \leftarrow B - 1$.
      Remove indices $n$ from $\mathcal{C}$ such that $a_n = 0$ or $\widetilde{E}_n > \gamma$.
    $\mathbf{b}$ = solution corresponding to $\widehat{E}_{\text{NLZ}}(B)$.
    for $n \in \mathcal{C}$ do
      Determine trial allocation $\mathbf{a}_n$ from $\mathbf{a}_0$.
      Use Algorithm 4 to approximately minimize quantization error given allocation $\mathbf{a}_n$.
      $\widetilde{E}_n$ = error value returned.
    $\widehat{E}_{\text{NLZ}}(B-1) = \min_{n \in \mathcal{C}} \widetilde{E}_n$.
  until $\widehat{E}_{\text{NLZ}}(B-1) > \gamma$
```

### 5.3.2 Approximate error minimization given a fixed cost allocation

Next we turn to the problem of minimizing the quantization error given a fixed allocation of bits to coefficients. The basic idea of our algorithm is to start from the continuous-valued minimizer of the left-hand side of (2.1.1), $\mathbf{b} = \mathbf{c}$, and quantize the components of $\mathbf{b}$ one by one, each time modifying the remaining unquantized components to partially compensate for the increase in the error. As discussed in [109], the final quantization error depends on the order in which the components are quantized, but determining an optimal ordering is intractable. We address first the more straightforward aspects of the algorithm and propose later a simple rule for selecting an ordering. Since our goal in this section is to develop a low-complexity algorithm, more sophisticated order selection methods are not considered.

We use $\mathcal{K}$ to represent the subset of coefficients that have been quantized and $\mathcal{F}$ to represent the remaining coefficients. As the algorithm progresses, the number of free variables decreases, giving rise to problems of lower dimension that we again refer to as subproblems. As suggested by Fig. 5-2, a subproblem created by fixing coefficients is of the same form as the original problem. This is shown more formally in Appendix C.2, where the parameters for the subproblem are also derived. The equivalence between the original problem and arbitrary subproblems can be exploited to simplify the algorithm following the same approach as in Section 2.3. We begin by selecting the first coefficient to be quantized and modifying the values of the remaining coefficients to compensate. The subproblem that results from quantizing the first coefficient to a value $b_m$ is characterized by the parameters in (C.2.7). This subproblem now takes the place of the original problem and we select the next coefficient to quantize and compensate with the remaining coefficients in the same way as before. The parameters for the second subproblem can be determined using (C.2.7) with the parameters for the first subproblem playing the role of $\mathbf{Q}$, $\mathbf{c}$, and $\gamma$. It suffices therefore to describe a single iteration of the algorithm and to specify the recursion relating the parameters of one subproblem to the next.

In the remainder of this subsection, a superscript $i$ is used to associate quantities with iteration $i$, for example the parameters $\mathbf{Q}^{(i)}$, $\mathbf{c}^{(i)}$ and $\gamma^{(i)}$, and the current coefficient values $\mathbf{b}^{(i)}$. As in Section 2.3, we also define $\mathbf{R}^{(i)}$ to be the inverse of $\mathbf{Q}^{(i)}$. Initially we set $\mathcal{K}^{(0)} = \emptyset$, $\mathcal{F}^{(0)} = \{1, \dots, N\}$, $\mathbf{Q}^{(0)} = \mathbf{Q}$, $\mathbf{R}^{(0)} = \mathbf{Q}^{-1}$, $\mathbf{c}^{(0)} = \mathbf{c}$, $\gamma^{(0)} = \gamma$, and $\mathbf{b}^{(0)} = \mathbf{c}$.

In iteration $i$, a new coefficient corresponding to the index $m \in \mathcal{F}^{(i)}$ is chosen to be quantized. The rule for selecting $m$, which determines the order of quantization, is discussed at the end of this subsection. The current continuous value for the $m$th coefficient, $b_m^{(i)}$, is quantized to $a_m$ bits as described in Section 5.2.1 (or Appendix C.1 in the CSD case), yielding an integer $b_m^{(i+1)}$. The values of already quantized coefficients remain the same, i.e., $b_n^{(i+1)} = b_n^{(i)}$ for $n \in \mathcal{K}^{(i)}$. Next, the index $m$ is removed from $\mathcal{F}^{(i)}$ and added to $\mathcal{K}^{(i)}$ to form $\mathcal{F}^{(i+1)}$ and $\mathcal{K}^{(i+1)}$. To compensate for the quantization of the $m$th coefficient, the values of the remaining unquantized coefficients, $b_n^{(i)}$ for $n \in \mathcal{F}^{(i+1)}$, are modified so as to minimize the error. This corresponds to minimizing (C.2.1) with respect to $\mathbf{b}_{\mathcal{F}}$. From Appendix C.2, the result is

$$\mathbf{b}^{(i+1)}_{\mathcal{F}^{(i+1)}} = \mathbf{c}^{(i+1)} = \mathbf{c}^{(i)}_{\mathcal{F}^{(i+1)}} + \frac{b_m^{(i+1)} - c_m^{(i)}}{R_{mm}^{(i)}} \mathbf{R}^{(i)}_{\mathcal{F}^{(i+1)}m}, \tag{5.3.1}$$

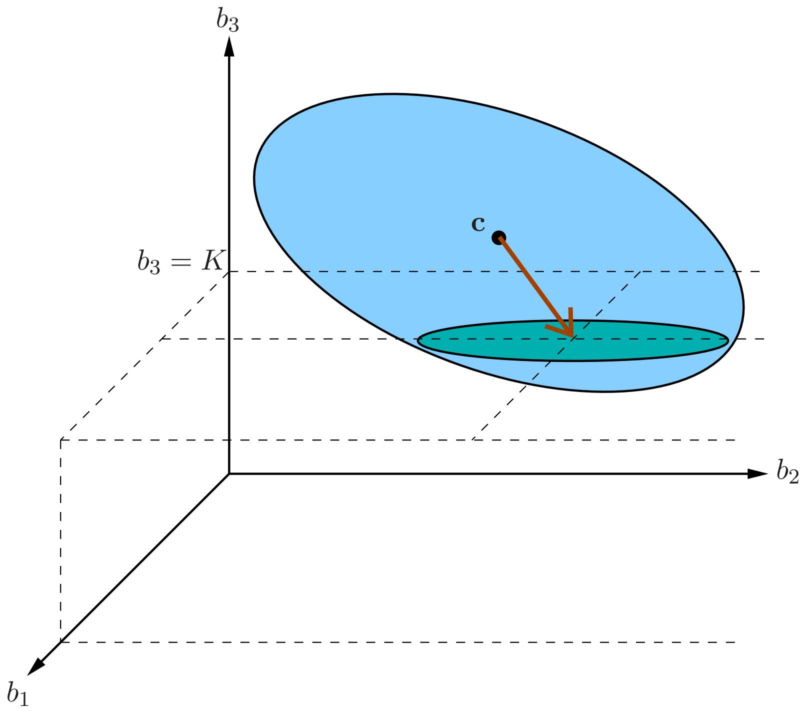


Figure 5-2: Geometric representation of a two-dimensional subproblem formed by quantizing the coefficient $b_3$ to a value $K$. The arrow depicts the quantization of $b_3$ together with the compensating changes in $b_1$ and $b_2$ to re-center the solution.

which can be interpreted geometrically as a re-centering as shown in Fig. 5-2. Equation (5.3.1) also relates the new parameter $\mathbf{c}^{(i+1)}$ to the old parameters. The recursions for the other parameters are given by

$$\mathbf{Q}^{(i+1)} = \mathbf{Q}^{(i)}_{\mathcal{F}(i+1)\mathcal{F}(i+1)}, \tag{5.3.2}$$

$$\mathbf{R}^{(i+1)} = \mathbf{R}^{(i)}_{\mathcal{F}(i+1)\mathcal{F}(i+1)} - \frac{1}{R^{(i)}_{mm}}\mathbf{R}^{(i)}_{\mathcal{F}(i+1)m}\mathbf{R}^{(i)}_{m\mathcal{F}(i+1)}, \tag{5.3.3}$$

$$\gamma^{(i+1)} = \gamma^{(i)} - \frac{\left(b^{(i+1)}_m - c^{(i)}_m\right)^2}{R^{(i)}_{mm}}. \tag{5.3.4}$$

The algorithm can now continue with iteration $i+1$.

After $N$ iterations, all coefficients have been quantized and the solution $\mathbf{b}^{(N)}$ is integer-valued. The final quantization error corresponds to the decrease in the parameter $\gamma$ and is given by $\widehat{E}_{\mathrm{NLZ}}\left(\sum_n a_n\right) = \gamma^{(0)} - \gamma^{(N)}$.

We now present a rule for selecting the coefficient to be quantized in each iteration. The rule is based on an alternative interpretation of the quadratic form in (2.1.1), which we rewrite as

$$(\mathbf{b}-\mathbf{c})^T\mathbf{Q}(\mathbf{b}-\mathbf{c}) = \left\|\mathbf{Q}^{1/2}\mathbf{b}-\mathbf{Q}^{1/2}\mathbf{c}\right\|_2^2. \tag{5.3.5}$$

The right-hand side of (5.3.5) can be regarded as the squared Euclidean distance between the vector $\mathbf{Q}^{1/2}\mathbf{b}$, which is a linear combination of vectors corresponding to the columns of $\mathbf{Q}^{1/2}$, and a fixed vector $\mathbf{Q}^{1/2}\mathbf{c}$. In the beginning when no coefficients have been quantized, $\mathbf{b}=\mathbf{c}$ and the distance is zero. Each time a coefficient is quantized, we modify the values of the remaining unquantized coefficients to minimize the increase in the distance. The quality of the compensation tends to be better when the angles between pairs of vectors are small, as suggested in Fig. 5-3. Moreover, as the number of unquantized components decreases, the ability to compensate for quantization also decreases. Based on these tendencies, we suggest choosing $m$ so that it corresponds to the largest angle between vectors with as yet unquantized coefficients. The aim is to eliminate large angles early in the process when more degrees of freedom are available for compensation, rather than leaving them for later rounds when there are fewer degrees of freedom. Given that the cosine of the angle between two vectors is equal to their normalized inner product, and that the inner product between columns $k$ and $n$ of $\mathbf{Q}^{1/2}$ is equal to $Q_{kn}$, the two indices $m_1$ and $m_2$ corresponding to the largest angle can be determined as follows:

$$(m_1, m_2) = \arg \min_{\substack{k,n\in\mathcal{F}^{(i)}\\ k\neq n}} \frac{|Q_{kn}|}{\sqrt{Q_{kk}Q_{nn}}}. \tag{5.3.6}$$

To decide between $m_1$ and $m_2$, we compare the second-largest angles in which columns $m_1$ and $m_2$ of $\mathbf{Q}^{1/2}$ participate, i.e., we compare

$$\min_{\substack{n\in\mathcal{F}^{(i)}\\ n\neq m_2}} \frac{|Q_{m_1n}|}{\sqrt{Q_{m_1m_1}Q_{nn}}}, \qquad \min_{\substack{n\in\mathcal{F}^{(i)}\\ n\neq m_1}} \frac{|Q_{m_2n}|}{\sqrt{Q_{m_2m_2}Q_{nn}}}. \tag{5.3.7}$$

If the quantity on the left is smaller than the one on the right, we choose $m = m_1$, otherwise $m = m_2$.

A summary of the algorithm of this subsection is given in Algorithm 4.

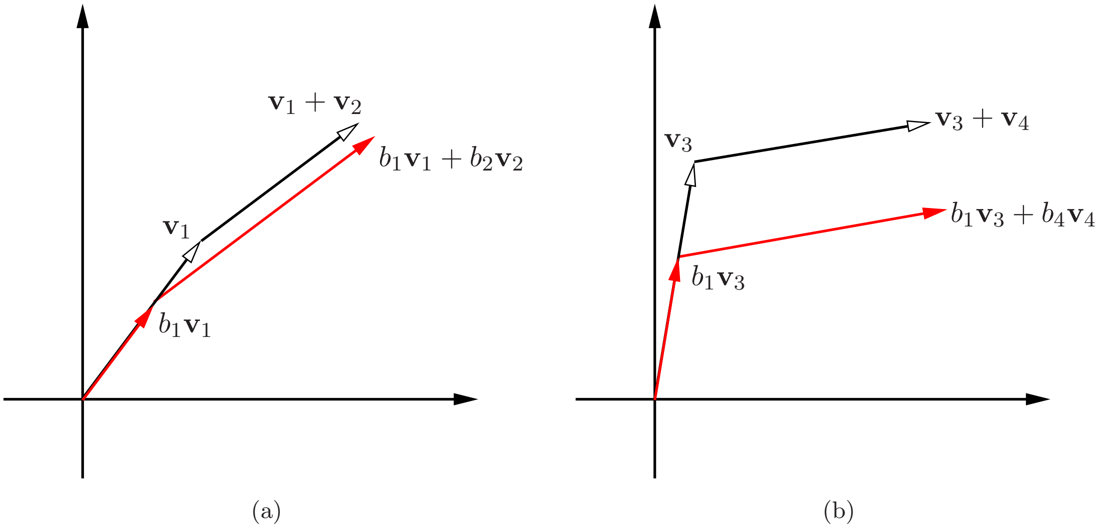

$\mathbf{v}_1 + \mathbf{v}_2$
$b_1\mathbf{v}_1 + b_2\mathbf{v}_2$
$\mathbf{v}_1$
$b_1\mathbf{v}_1$
(a)
$\mathbf{v}_3$
$\mathbf{v}_3 + \mathbf{v}_4$
$b_1\mathbf{v}_3 + b_4\mathbf{v}_4$
$b_1\mathbf{v}_3$
(b)

Figure 5-3: The effect of the angle between two vectors on the quality of compensation. The scale factor $b_1$ is the same in both (a) and (b). The modified linear combination $b_1\mathbf{v}_1 + b_2\mathbf{v}_2$ is much closer to the original $\mathbf{v}_1 + \mathbf{v}_2$ in (a) than $b_1\mathbf{v}_3 + b_4\mathbf{v}_4$ is to $\mathbf{v}_3 + \mathbf{v}_4$ in (b).

**Algorithm 4** Approximate error minimization given fixed cost allocation

```
Input: Parameters $\mathbf{Q}$, $\mathbf{c}$, $\gamma$, wordlength $P$, allocation of bits/SPTs $(a_1, \ldots, a_N)$.
Output: Quantized solution $\mathbf{b}$, quantization error $\widehat{E}_{\text{NLZ}}\left(\sum_n a_n\right)$.
  Initialize: $i = 0$, $\mathcal{K}^{(0)} = \emptyset$, $\mathcal{F}^{(0)} = \{1, \ldots, N\}$, $\mathbf{Q}^{(0)} = \mathbf{Q}$, $\mathbf{R}^{(0)} = \mathbf{Q}^{-1}$, $\mathbf{c}^{(0)} = \mathbf{c}$, $\gamma^{(0)} = \gamma$,
  $\mathbf{b}^{(0)} = \mathbf{c}$.
  for $i = 0, \ldots, N-1$ do
    Determine $m$ from (5.3.6) and (5.3.7).
    Quantize $b_m^{(i)}$ to $a_m$ bits $\longrightarrow b_m^{(i+1)}$.
    $\mathbf{b}_{\mathcal{K}^{(i)}}^{(i+1)} = \mathbf{b}_{\mathcal{K}^{(i)}}^{(i)}$.
    $\mathcal{K}^{(i+1)} = \mathcal{K}^{(i)} \cup \{m\}$, $\mathcal{F}^{(i+1)} = \mathcal{F}^{(i)} \backslash m$.
    Update $\mathbf{b}_{\mathcal{F}^{(i+1)}}$, $\mathbf{Q}$, $\mathbf{R}$, $\mathbf{c}$, $\gamma$ using (5.3.1)–(5.3.4).
    $i \leftarrow i + 1$.
  Return solution: $\mathbf{b} = \mathbf{b}^{(N)}$, $\widehat{E}_{\text{NLZ}}\left(\sum_n a_n\right) = \gamma^{(0)} - \gamma^{(N)}$.
```

# Chapter 6

# Bit-efficient filter design under a quadratic constraint: Optimal algorithm for the general case

This chapter focuses on optimal algorithms for problems (5.1.1) and (5.1.2). For this purpose, we again make use of the method of branch-and-bound introduced in Section 3.1. As in Chapter 3, the emphasis is on reducing the complexity of branch-and-bound by developing strong and efficiently computable lower bounds on the optimal cost.

The organization of this chapter is similar to that of Chapter 3. Section 6.1 discusses at a high level the application of branch-and-bound to problems (5.1.1) and (5.1.2). A branch-and-bound algorithm is presented in further detail in Section 6.7. Sections 6.2–6.4 are devoted to developing lower bounds for use in the branch-and-bound algorithm. In Section 6.2, we give bounds based on the range of possible quantized values for each coefficient under the quadratic constraint (2.1.1). To obtain stronger lower bounds, in Section 6.3 we derive relaxations of both the NLZ and SPT minimization problems in which the discrete-value constraint is relaxed and the cost function is linearized. In Section 6.4, we develop an alternative relaxation that exploits the solution methods of Section 5.2 for the diagonal case and we analyze the approximation properties of this relaxation. An efficient algorithm for obtaining a diagonal relaxation is described in Section 6.5. In Section 6.6, the lower bounds resulting from the methods of Sections 6.2–6.4 are evaluated and compared numerically over a range of problem instances.

## 6.1 Branch-and-bound

In this section, we indicate briefly how branch-and-bound can be applied to solve (5.1.1) and (5.1.2). A detailed description of our branch-and-bound algorithm is provided later in Section 6.7.

As with the sparsity maximization problem (2.0.1), problems (5.1.1) and (5.1.2) can be divided into subproblems by making hard decisions on some of the components of $\mathbf{b}$. In the present case, the decisions involve different quantization levels as opposed to binary choices between zero and non-zero values. To determine the quantization levels that need to be considered for a coefficient $b_n$, we refer to (3.3.3), which specifies the minimum and maximum real values for $b_n$ subject to the quadratic constraint (2.1.1). Given the additional integer constraint on $b_n$, the minimum and maximum values become respectively

$$\underline{B}_n = \left\lceil c_n - \sqrt{\gamma\left(\mathbf{Q}^{-1}\right)_{nn}} \right\rceil, \tag{6.1.1a}$$

$$\overline{B}_n = \left\lfloor c_n + \sqrt{\gamma\left(\mathbf{Q}^{-1}\right)_{nn}} \right\rfloor. \tag{6.1.1b}$$

Each of the quantization levels $\underline{B}_n, \ldots, \overline{B}_n$ for $b_n$ has the property that there exist real values for the other components of $\mathbf{b}$ such that (2.1.1) is satisfied. It is generally difficult however to verify whether a fully integer-valued feasible solution exists given a fixed value for $b_n$. Since it cannot be easily established that the values $\underline{B}_n, \ldots, \overline{B}_n$ are either feasible or infeasible for problems (5.1.1) and (5.1.2), we refer to each as a candidate value and to the collection of values as the candidate range for $b_n$.

In our branch-and-bound algorithm, subproblems are created by selecting a coefficient and fixing it to every integer value in its candidate range. This branching process leads again to a tree of subproblems as depicted in Fig. 6-1. Unlike in Fig. 3-1, the tree is not restricted to be binary. As discussed in Appendix C.2, every subproblem resulting from the fixing of coefficients is equivalent to a lower-dimensional instance of the original problem with parameters given by (C.2.2)–(C.2.6).

Two special cases can be identified. First, if $\underline{B}_n > \overline{B}_n$ for any index $n$, which can happen if the real-valued range for $b_n$ does not contain an integer, the current subproblem is infeasible. Second, if $\underline{B}_n = \overline{B}_n$, there is only one candidate value for $b_n$ and the dimensionality of the current subproblem can be reduced by one.

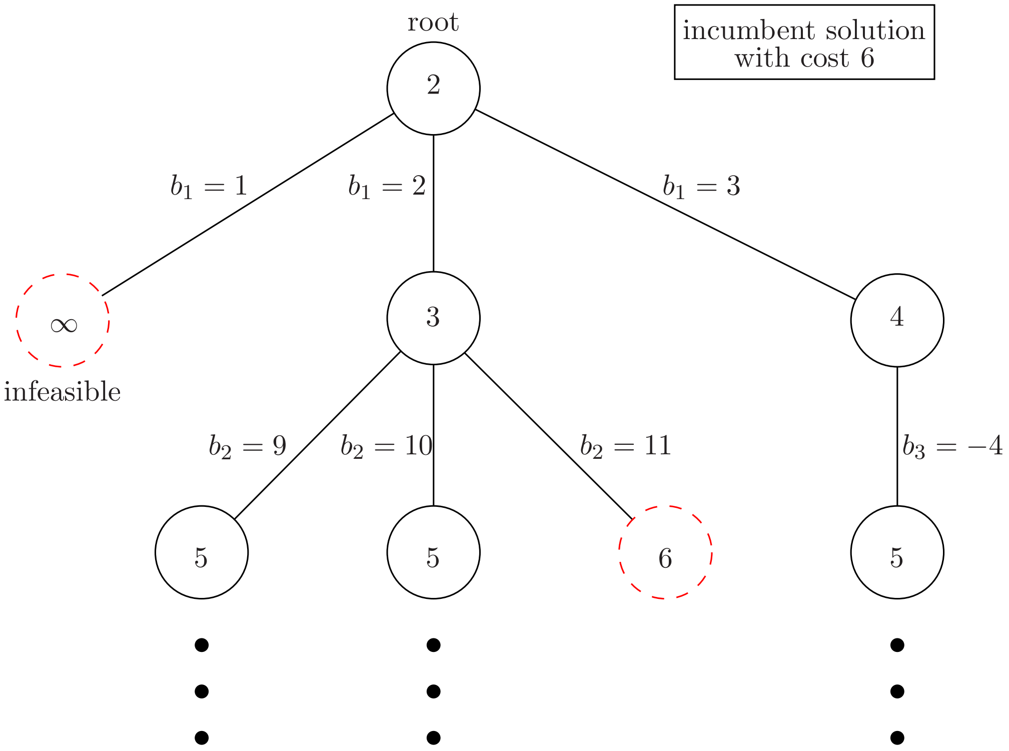


Figure 6-1: Example of a branch-and-bound tree for problem (5.1.2). Each node represents a subproblem and the branch labels indicate the coefficient that is fixed in going from a parent to a child. The top left branch leads to an infeasible subproblem, i.e., one with $\underline{B}_n > \overline{B}_n$ for some $n$, whereas the top right branch leads to a subproblem with $\underline{B}_3 = \overline{B}_3 = -4$ and hence a single child. The number labelling each node is a lower bound on the optimal cost of the corresponding subproblem. Given an incumbent solution with a cost of 6, the subproblems marked by dashed circles need not be considered any further.

As before, the algorithm computes lower bounds on the optimal values of subproblems. We devote Sections 6.2–6.4 to the development of efficiently computable lower bounds on the optimal values of (5.1.1) and (5.1.2). For a general subproblem, one contribution to the lower bound comes from the cost of coefficients that have been fixed while the other contribution comes from the application of the methods in Sections 6.2–6.4 to the free coefficients.

An initial incumbent solution can be obtained using the heuristic algorithm of Section 5.3. Additional feasible solutions are generated whenever the branch-and-bound tree reaches a depth of $N$ levels, implying that all coefficients have been quantized, or potentially by using the heuristic algorithm on subproblems. The selection of coefficients for branching and the ordering of open subproblems are addressed in Section 6.7.

## 6.2 Bounds based on candidate ranges

In this section, we provide lower bounds on the optimal values of (5.1.1) and (5.1.2) that are based on the range of candidate values $\underline{B}_n, \dots, \overline{B}_n$ for each coefficient, where $\underline{B}_n$ and $\overline{B}_n$ are given in (6.1.1). The analogous technique in sparse filter design is to identify coefficients for which a zero value is infeasible, as discussed in Section 3.2. Similar to the lower bounds in Section 3.2, the bounds derived in the current section require the least amount of computation and are accordingly the weakest among the bounds presented in Sections 6.2–6.4.

A lower bound on the optimal value of (5.1.1) may be obtained by minimizing the cost function $C_{\mathrm{NLZ}}(b_n)$ independently for each coefficient over its candidate range, and similarly for (5.1.2). We assume that $\underline{B}_n < \overline{B}_n$ for all $n$ since the cases $\underline{B}_n > \overline{B}_n$ and $\underline{B}_n = \overline{B}_n$ can be eliminated as discussed in Section 6.1. For both cost functions, if the candidate range for a coefficient includes the value zero, then the contribution to the lower bound is equal to zero. If the candidate range does not include zero and the cost is the number of NLZ bits, the contribution to the lower bound is equal to the cost of the smallest candidate value in magnitude ($\underline{B}_n$ if $\underline{B}_n > 0$, $\overline{B}_n$ if $\overline{B}_n < 0$) since $C_{\mathrm{NLZ}}(b_n)$ is monotonic in $|b_n|$. On the other hand, the cost function $C_{\mathrm{SPT}}(b_n)$ is not monotonic and an exhaustive search over the candidate range may be required to determine the contribution to the lower bound. Summing these contributions gives the following lower bound for (5.1.1):

$$\sum_{n:\underline{B}_n>0} \left\lceil \log_2(1+\underline{B}_n) \right\rceil + \sum_{n:\overline{B}_n<0} \left\lceil \log_2\left(1+\left|\overline{B}_n\right|\right) \right\rceil . \tag{6.2.1}$$

The corresponding lower bound for (5.1.2) is

$$\sum_{n:0\notin \underline{B}_n,\dots,\overline{B}_n} \min_{b_n=\underline{B}_n,\dots,\overline{B}_n} C_{\mathrm{SPT}}(b_n). \tag{6.2.2}$$

The minimizations in (6.2.2) can be solved using a lookup table listing integers in CSD form.

The lower bounds in (6.2.1) and (6.2.2) can be interpreted geometrically. As illustrated in Fig. 6-2, the candidate ranges $\underline{B}_n, \dots, \overline{B}_n$ together specify a coordinate-aligned box, denoted as $\mathcal{B}_{\mathbf{Q}}$, that must contain all feasible integer-valued solutions to (5.1.1) or (5.1.2).

Minimizing $C_{\text{NLZ}}(\mathbf{b})$ and $C_{\text{SPT}}(\mathbf{b})$ over $\mathcal{B}_{\mathbf{Q}}$ yields the lower bounds in (6.2.1) and (6.2.2). We can gain some insight into the quality of the box approximation by examining the ratio between the volume of $\mathcal{B}_{\mathbf{Q}}$ and the volume of the original ellipsoid $\mathcal{E}_{\mathbf{Q}}$. The former is bounded by

$$(2\sqrt{\gamma})^N \prod_n \left( \sqrt{\left(\mathbf{Q}^{-1}\right)_{nn}} - 1 \right) < \text{vol}(\mathcal{B}_{\mathbf{Q}}) \leq (2\sqrt{\gamma})^N \sqrt{\prod_n \left(\mathbf{Q}^{-1}\right)_{nn}},$$

using (6.1.1) and the properties of the floor and ceiling functions. The volume of $\mathcal{E}_{\mathbf{Q}}$ is given by

$$\text{vol}(\mathcal{E}_{\mathbf{Q}}) = \frac{(\pi\gamma)^{N/2}}{\Gamma(N/2+1)} \sqrt{\det\left(\mathbf{Q}^{-1}\right)}, \tag{6.2.3}$$

where $\Gamma(\cdot)$ denotes the gamma function. The worst-case volume ratio is

$$\left(\frac{4}{\pi}\right)^{N/2} \Gamma(N/2+1) \sqrt{\frac{\prod \left(\mathbf{Q}^{-1}\right)_{nn}}{\det\left(\mathbf{Q}^{-1}\right)}}, \tag{6.2.4}$$

which grows very rapidly with $N$. Therefore the quality of the approximation, and by extension the strength of the lower bounds (6.2.1)–(6.2.2), degrade significantly as the number of dimensions increases.

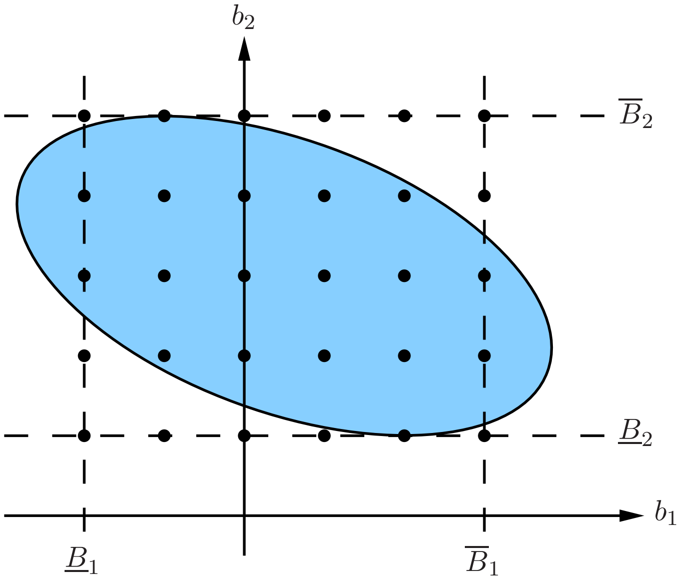


Figure 6-2: Coordinate-aligned box corresponding to candidate ranges. The black dots represent integer-valued solutions.

Unlike in Section 3.2, it is generally difficult to verify whether there exists a feasible solution that achieves the lower bounds in (6.2.1) and (6.2.2). In particular, if for many of the components $b_n$ the candidate value with minimal cost is not unique, there can be exponentially many values of $\mathbf{b}$ that achieve the lower bound and a combinatorial search is required to determine whether any of them are feasible. For the same reason, determining whether a feasible solution exists with a cost one unit higher than the bounds in (6.2.1) and (6.2.2) is also difficult. In contrast, to verify whether the lower bound of $|\mathcal{U}|$ in Section 3.2 is achievable, we need only check whether the solution $\mathbf{b}_{\mathcal{F}} = \mathbf{0}$ is feasible, and to do the same for the next highest value $|\mathcal{U}| + 1$, it is sufficient to evaluate (3.2.1) for all $n \in \mathcal{F}$.

## 6.3 Linear relaxation

To improve upon the lower bounds of the previous section, we consider the use of more sophisticated relaxations. In this section, we develop linear relaxations of problems (5.1.1) and (5.1.2) in which the integer constraint on $\mathbf{b}$ is relaxed and the cost functions are linearized. Different approaches are used for the two cost functions $C_{\mathrm{NLZ}}(\mathbf{b})$ and $C_{\mathrm{SPT}}(\mathbf{b})$ as described in Sections 6.3.1 and 6.3.3 respectively. In both cases, the lower bound resulting from linear relaxation is guaranteed by construction to be stronger than the corresponding bound in Section 6.2. It is also shown in Section 6.3.2 that the optimal value of the linear relaxation of (5.1.1) is bounded from above in a similar manner as that of the linear relaxation (3.3.8) of the sparsity maximization problem.

### 6.3.1 Linear relaxation of the NLZ minimization problem

We begin by deriving a linear relaxation of (5.1.1). As a first step, the integer constraint on $\mathbf{b}$ is relaxed and is replaced with the interval constraints $\underline{B}_n \le b_n \le \overline{B}_n$, where $\underline{B}_n$ and $\overline{B}_n$ are the minimum and maximum candidate values in (6.1.1). The resulting problem is

$$\begin{aligned} \min_{\mathbf{b}} \quad & \sum_{n=1}^{N} \lceil \log_2(1 + |b_n|) \rceil \\ \text{s.t.} \quad & (\mathbf{b} - \mathbf{c})^T \mathbf{Q} (\mathbf{b} - \mathbf{c}) \le \gamma, \\ & \underline{B}_n \le b_n \le \overline{B}_n \quad \forall\, n, \end{aligned} \tag{6.3.1}$$

where we have used (5.1.4). Problem (6.3.1) has a non-convex and discontinuous cost function, making it difficult to solve. To obtain a convex optimization problem, we replace each component $\lceil \log_2(1+|b_n|) \rceil$ of the true cost function by a continuous convex underestimator that is as tight as possible. In addition, we wish to ensure that the lower bound resulting from the minimization of the underestimator of $C_{\mathrm{NLZ}}(\mathbf{b})$ is an improvement over the lower bound in (6.2.1). This can be achieved if for each $n$ the minimum value of the underestimator of $\lceil \log_2(1+|b_n|) \rceil$ is greater than the $n$th term in (6.2.1) (recall that the terms not appearing in (6.2.1) are equal to zero). Given these requirements, the form of the underestimator is different for the three cases $\underline{B}_n < 0 < \overline{B}_n$, $\underline{B}_n \geq 0$, and $\overline{B}_n \leq 0$, which are considered separately below.

For the case $\underline{B}_n < 0 < \overline{B}_n$, the contribution to the lower bound in (6.2.1) is equal to zero, and hence the underestimator must equal zero at $b_n = 0$. As can be seen in Fig. 6-3, the requirement that the underestimator be convex and tight implies that it has a piecewise linear shape with a change in slope at $b_n = 0$. Furthermore, the magnitudes $w_n^+$ and $w_n^-$ of the slopes on either side of $b_n = 0$ should be maximized. For $b_n > 0$, the two points that impose the tightest upper bounds on the slope are $\left(\overline{B}_n, \lceil \log_2(1+\overline{B}_n) \rceil\right)$ and $(2^{p_n^+}-1, p_n^+)$, where $p_n^+ = \left\lfloor \log_2(1+\overline{B}_n) \right\rfloor$. The point $(2^{p_n^+}-1, p_n^+)$ corresponds to the filled circle at $(7,3)$ in Fig. 6-3. It follows that $w_n^+$ should be chosen as

$$w_n^+ = \min\left\{ \frac{\lceil \log_2(1+\overline{B}_n) \rceil}{\overline{B}_n}, \frac{p_n^+}{2^{p_n^+}-1} \right\}. \tag{6.3.2}$$

Similarly, $w_n^-$ is given by

$$w_n^- = \min\left\{ \frac{\lceil \log_2(1+|\underline{B}_n|) \rceil}{|\underline{B}_n|}, \frac{p_n^-}{2^{p_n^-}-1} \right\}, \tag{6.3.3}$$

where $p_n^- = \lfloor \log_2(1+|\underline{B}_n|) \rfloor$ and $(2^{p_n^-}-1, p_n^-)$ corresponds to the filled circle at $(-3,2)$ in Fig. 6-3.

For the case $\underline{B}_n \geq 0$, the contribution to (6.2.1) is equal to $\lceil \log_2(1+\underline{B}_n) \rceil$ and consequently the underestimator must take the value $\lceil \log_2(1+\underline{B}_n) \rceil$ at $b_n = \underline{B}_n$. The tightest convex underestimator is now linear as seen in Fig. 6-4. If $\underline{B}_n = 2^p - 1$ for some integer $p$ as in the left panel of Fig. 6-4, the slope $w_n$ is positive and is constrained by the points $\left(\overline{B}_n, \lceil \log_2(1+\overline{B}_n) \rceil\right)$ and $(2^{p_n^+}-1, p_n^+)$ where $p_n^+$ is defined as before. Otherwise, $w_n$ is

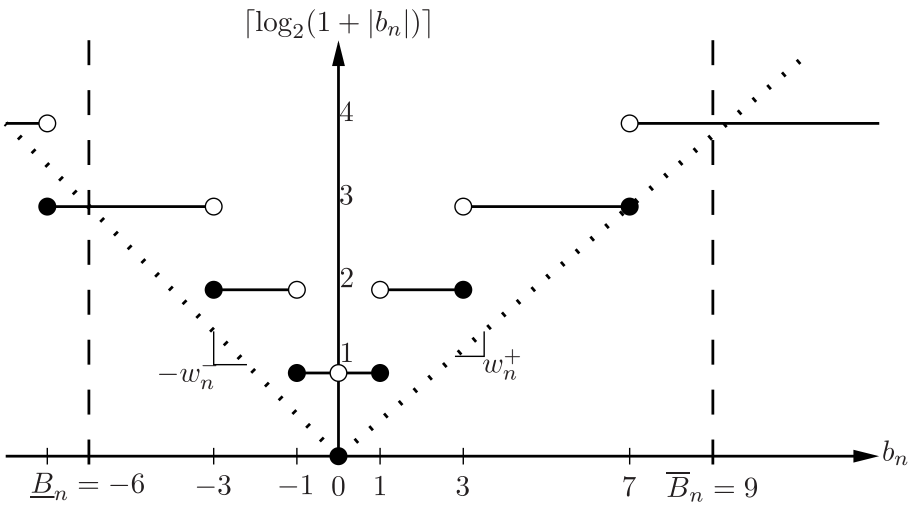


Figure 6-3: The function $\lceil \log_2(1+|b_n|)\rceil$ (solid) and the tightest possible convex underestimator (dotted) over the interval $[\underline{B}_n, \overline{B}_n]$ for the case $\underline{B}_n < 0 < \overline{B}_n$.

forced to be zero as shown in the right panel of Fig. 6-4. Combining these two cases,

$$w_n = \begin{cases} \min\left\{ \dfrac{\lceil \log_2(1+\overline{B}_n)\rceil - \lceil \log_2(1+\underline{B}_n)\rceil}{\overline{B}_n - \underline{B}_n}, \dfrac{p_n^+ - \lceil \log_2(1+\underline{B}_n)\rceil}{2^{p_n^+} - 1 - \underline{B}_n} \right\}, & \underline{B}_n = 2^p - 1, \\ 0 & \text{otherwise.} \end{cases} \tag{6.3.4}$$

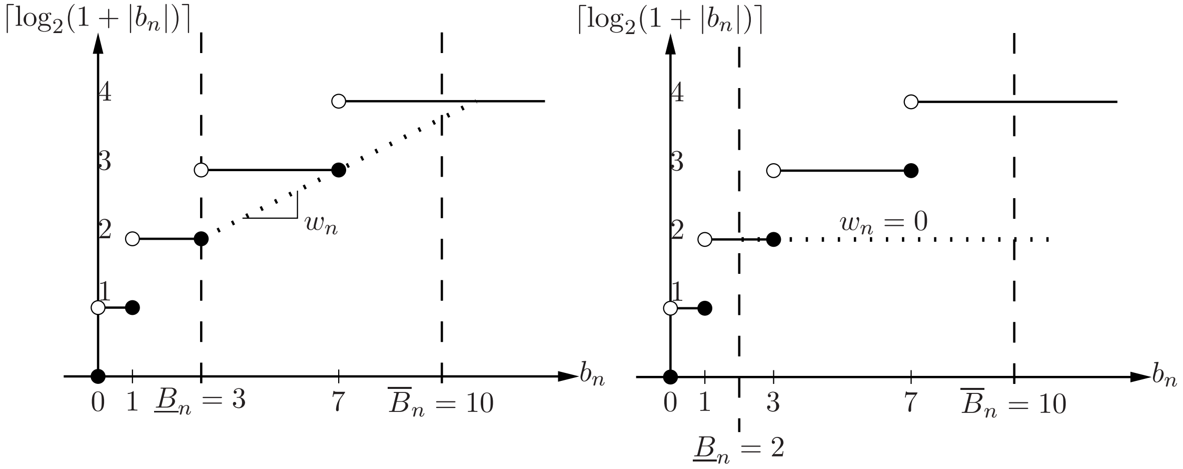


Figure 6-4: The function $\lceil \log_2(1+|b_n|)\rceil$ (solid) and the tightest possible convex underestimator (dotted) over the interval $[\underline{B}_n, \overline{B}_n]$ for the case $\underline{B}_n \geq 0$. In the example on the left, $\underline{B}_n = 2^2 - 1 = 3$ and the slope $w_n$ is positive, whereas on the right, the location of $\underline{B}_n = 2$ to the left of a filled circle forces $w_n$ to be zero.

The case $\overline{B}_n \leq 0$ is similar to the case $\underline{B}_n \geq 0$. The underestimator is again linear and passes through the point $\left(\overline{B}_n, \left\lceil \log_2(1+\left|\overline{B}_n\right|)\right\rceil\right)$ with a negative slope $-w_n$ where

$$w_n = \begin{cases} \min\left\{ \dfrac{\lceil \log_2(1+|\underline{B}_n|)\rceil - \left\lceil \log_2(1+\left|\overline{B}_n\right|)\right\rceil}{|\underline{B}_n| - \left|\overline{B}_n\right|}, \dfrac{p_n^- - \left\lceil \log_2(1+\left|\overline{B}_n\right|)\right\rceil}{2^{p_n^-} - 1 - \left|\overline{B}_n\right|} \right\}, & \left|\overline{B}_n\right| = 2^p - 1, \\ 0 & \text{otherwise.} \end{cases} \tag{6.3.5}$$

We have now replaced the cost function in (6.3.1) with a sum of convex univariate functions of the components $b_n$. The functions corresponding to the case $\underline{B}_n < 0 < \overline{B}_n$ are asymmetrically-weighted absolute value functions while the remaining functions are linear. Following the approach in Section 3.3 of separating positive and negative parts, the absolute value functions can be recast as linear functions. We first define the subsets

$$\mathcal{D} = \{n : \underline{B}_n < 0 < \overline{B}_n\}, \quad \mathcal{P} = \{n : \underline{B}_n \geq 0\}, \quad \mathcal{N} = \{n : \overline{B}_n \leq 0\}.$$

Using the representation in (3.3.5), we express $\mathbf{b}_{\mathcal{D}}$ in terms of its positive and negative parts as $\mathbf{b}_{\mathcal{D}} = \mathbf{b}_{\mathcal{D}}^+ - \mathbf{b}_{\mathcal{D}}^-$ with $\mathbf{b}_{\mathcal{D}}^+, \mathbf{b}_{\mathcal{D}}^- \geq \mathbf{0}$. Each of the interval constraints for $n \in \mathcal{D}$ is transformed into the pair of constraints $0 \leq b_n^+ \leq \overline{B}_n$ and $0 \leq b_n^- \leq |\underline{B}_n|$. Incorporating the underestimators derived for the three cases and also rewriting the quadratic constraint in terms of $\mathbf{b}_{\mathcal{D}}^+$ and $\mathbf{b}_{\mathcal{D}}^-$, the relaxed problem can now be formulated as follows:

$$\begin{aligned} &\sum_{n\in\mathcal{P}} \lceil \log_2(1+\underline{B}_n)\rceil + \sum_{n\in\mathcal{N}} \left\lceil \log_2(1+\left|\overline{B}_n\right|)\right\rceil + \\ &\qquad \min_{\mathbf{b}_{\mathcal{D}}^+, \mathbf{b}_{\mathcal{D}}^-, \mathbf{b}_{\mathcal{P}}, \mathbf{b}_{\mathcal{N}}} \quad \sum_{n\in\mathcal{D}} \left(w_n^+ b_n^+ + w_n^- b_n^-\right) + \sum_{n\in\mathcal{P}} w_n(b_n - \underline{B}_n) - \sum_{n\in\mathcal{N}} w_n(b_n - \overline{B}_n) \\ &\qquad \text{s.t.} \quad \begin{bmatrix} \mathbf{b}_{\mathcal{D}}^+ - \mathbf{b}_{\mathcal{D}}^- - \mathbf{c}_{\mathcal{D}} \\ \mathbf{b}_{\mathcal{P}} - \mathbf{c}_{\mathcal{P}} \\ \mathbf{b}_{\mathcal{N}} - \mathbf{c}_{\mathcal{N}} \end{bmatrix}^T \begin{bmatrix} \mathbf{Q}_{\mathcal{D}\mathcal{D}} & \mathbf{Q}_{\mathcal{D}\mathcal{P}} & \mathbf{Q}_{\mathcal{D}\mathcal{N}} \\ \mathbf{Q}_{\mathcal{P}\mathcal{D}} & \mathbf{Q}_{\mathcal{P}\mathcal{P}} & \mathbf{Q}_{\mathcal{P}\mathcal{N}} \\ \mathbf{Q}_{\mathcal{N}\mathcal{D}} & \mathbf{Q}_{\mathcal{N}\mathcal{P}} & \mathbf{Q}_{\mathcal{N}\mathcal{N}} \end{bmatrix} \begin{bmatrix} \mathbf{b}_{\mathcal{D}}^+ - \mathbf{b}_{\mathcal{D}}^- - \mathbf{c}_{\mathcal{D}} \\ \mathbf{b}_{\mathcal{P}} - \mathbf{c}_{\mathcal{P}} \\ \mathbf{b}_{\mathcal{N}} - \mathbf{c}_{\mathcal{N}} \end{bmatrix} \leq \gamma, \\ &\qquad\qquad 0 \leq b_n^+ \leq \overline{B}_n, \quad n \in \mathcal{D}, \\ &\qquad\qquad 0 \leq b_n^- \leq |\underline{B}_n|, \quad n \in \mathcal{D}, \\ &\qquad\qquad \underline{B}_n \leq b_n \leq \overline{B}_n, \quad n \in \mathcal{P}, \mathcal{N}. \end{aligned} \tag{6.3.6}$$

Problem (6.3.6) is a convex optimization problem with a linear objective function and quadratic and interval constraints. The optimal value of (6.3.6) is a lower bound on that of (5.1.1) since we have relaxed the integer constraint on $\mathbf{b}$ and replaced the cost function by an underestimator. As with the linear relaxation in (3.3.8), we may take the ceiling of the optimal value of (6.3.6) and still maintain a lower bound because the optimal value of (5.1.1) is an integer. We will refer to (6.3.6) as the linear relaxation of (5.1.1).

The two constant terms in the objective function of (6.3.6) are the same as those in (6.2.1), while the remaining terms in the objective function are all non-negative. It follows that the optimal value of the linear relaxation is at least as large as the lower bound in (6.2.1), as desired.

Neglecting the constant terms in the objective, the dual of problem (6.3.6) is given by

$$\begin{aligned}
\max_{\boldsymbol{\pi},\boldsymbol{\nu}} \quad & -\left(\gamma \begin{bmatrix} \boldsymbol{\pi}_{\mathcal{D}} + \boldsymbol{\nu}_{\mathcal{D}}^{+} - \boldsymbol{\nu}_{\mathcal{D}}^{-} \\ \boldsymbol{\pi}_{\mathcal{P}} + \boldsymbol{\nu}_{\mathcal{P}} \\ \boldsymbol{\pi}_{\mathcal{N}} - \boldsymbol{\nu}_{\mathcal{N}} \end{bmatrix}^T \mathbf{Q}^{-1} \begin{bmatrix} \boldsymbol{\pi}_{\mathcal{D}} + \boldsymbol{\nu}_{\mathcal{D}}^{+} - \boldsymbol{\nu}_{\mathcal{D}}^{-} \\ \boldsymbol{\pi}_{\mathcal{P}} + \boldsymbol{\nu}_{\mathcal{P}} \\ \boldsymbol{\pi}_{\mathcal{N}} - \boldsymbol{\nu}_{\mathcal{N}} \end{bmatrix}\right)^{1/2} \\
& + \begin{bmatrix} \mathbf{c}_{\mathcal{D}} \\ \widetilde{\mathbf{c}}_{\mathcal{P}} \\ \widetilde{\mathbf{c}}_{\mathcal{N}} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{\pi}_{\mathcal{D}} + \boldsymbol{\nu}_{\mathcal{D}}^{+} - \boldsymbol{\nu}_{\mathcal{D}}^{-} \\ \boldsymbol{\pi}_{\mathcal{P}} + \boldsymbol{\nu}_{\mathcal{P}} \\ \boldsymbol{\pi}_{\mathcal{N}} - \boldsymbol{\nu}_{\mathcal{N}} \end{bmatrix} - \begin{bmatrix} \mathbf{p}_{\mathcal{D}}^{+} \\ \mathbf{p}_{\mathcal{D}}^{-} \\ \mathbf{p}_{\mathcal{P}} \\ \mathbf{p}_{\mathcal{N}} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{\nu}_{\mathcal{D}}^{+} \\ \boldsymbol{\nu}_{\mathcal{D}}^{-} \\ \boldsymbol{\nu}_{\mathcal{P}} \\ \boldsymbol{\nu}_{\mathcal{N}} \end{bmatrix} \qquad (6.3.7) \\
\text{s.t.} \quad & -\mathbf{w}_{\mathcal{D}}^{-} \le \boldsymbol{\pi}_{\mathcal{D}} \le \mathbf{w}_{\mathcal{D}}^{+}, \quad \boldsymbol{\nu}_{\mathcal{D}}^{+} \ge \mathbf{0}, \quad \boldsymbol{\nu}_{\mathcal{D}}^{-} \ge \mathbf{0}, \\
& \boldsymbol{\pi}_{\mathcal{P}} \le \mathbf{w}_{\mathcal{P}}, \quad \boldsymbol{\nu}_{\mathcal{P}} \ge \mathbf{0}, \\
& \boldsymbol{\pi}_{\mathcal{N}} \ge -\mathbf{w}_{\mathcal{N}}, \quad \boldsymbol{\nu}_{\mathcal{N}} \ge \mathbf{0},
\end{aligned}$$

where the vectors $\widetilde{\mathbf{c}}_{\mathcal{P}}$, $\widetilde{\mathbf{c}}_{\mathcal{N}}$, $\mathbf{p}_{\mathcal{D}}^{+}$, $\mathbf{p}_{\mathcal{D}}^{-}$, $\mathbf{p}_{\mathcal{P}}$, and $\mathbf{p}_{\mathcal{N}}$ are defined as follows:

$$\widetilde{c}_n = \begin{cases} c_n - \underline{B}_n, & n \in \mathcal{P}, \\ c_n - \overline{B}_n, & n \in \mathcal{N}, \end{cases} \qquad (6.3.8a)$$

$$p_n^{+} = \overline{B}_n, \quad p_n^{-} = \left|\underline{B}_n\right|, \qquad n \in \mathcal{D}, \qquad (6.3.8b)$$

$$p_n = \overline{B}_n - \underline{B}_n, \qquad n \in \mathcal{P}, \mathcal{N}. \qquad (6.3.8c)$$

A derivation of the dual is provided in Appendix D.1. The dual is also a convex optimization problem with the same optimal value as the primal and only upper and lower bound constraints. It can be solved more readily than the primal by solvers such as `fmincon` in

MATLAB because of the nature of the constraints.

It is possible to reduce the linear relaxation (6.3.6) of problem (5.1.1) to the linear relaxation (3.3.8) of the sparsity maximization problem. First, the subsets $\mathcal{P}$ and $\mathcal{N}$ can be assumed to be empty for the sparsity maximization problem since coefficients that must be strictly positive or strictly negative can be eliminated as discussed in Section 3.2. Second, the constraints $b_n^+ \leq \overline{B}_n$ and $b_n^- \leq |\underline{B}_n|$ can be removed as they are a consequence of the integer constraint on $\mathbf{b}$ in (5.1.1), which is not present in (2.0.1). The remaining distinction between (3.3.8) and (6.3.6) lies in the objective function weights. The weights $1/B_n^+$ and $1/B_n^-$ in (3.3.8) can also be derived using the convex underestimator approach of the current subsection. The non-convex function to be approximated in that case is the zero-norm $\|b_n\|_0$ over the interval $B_n^- \leq b_n \leq B_n^+$, and hence the underestimator is an asymmetrically-weighted absolute value function with slopes $1/B_n^+$ and $1/B_n^-$. The dual (6.3.7) reduces similarly to the corresponding dual (3.3.9), specifically by eliminating the subsets $\mathcal{P}$ and $\mathcal{N}$ and taking $\overline{B}_n$ and $\underline{B}_n$ to $+\infty$ and $-\infty$ respectively, the latter being equivalent to removing the constraints $b_n^+ \leq \overline{B}_n$ and $b_n^- \leq |\underline{B}_n|$. The resulting infinite penalty on $\boldsymbol{\nu}_{\mathcal{D}}^+$ and $\boldsymbol{\nu}_{\mathcal{D}}^-$ forces them to zero and we are left with an instance of (3.3.9) with $\boldsymbol{\pi}_{\mathcal{D}}$ in (6.3.7) corresponding to $\boldsymbol{\nu}$ in (3.3.9).

### 6.3.2 Absolute upper bound on the linear relaxation (6.3.6)

We now derive an upper bound on the optimal value of the linear relaxation (6.3.6). As will be explained shortly, the bound is analogous to the upper bound on (3.3.8) discussed in Section 3.3.3.

Given that problem (6.3.6) is a minimization, the objective value corresponding to any feasible solution is an upper bound on the optimal value. Assuming that $\underline{B}_n < \overline{B}_n$ for all $n$, the solution $\mathbf{b} = \mathbf{c}$ is feasible since it satisfies both the quadratic and interval constraints. In terms of the variables used in (6.3.6), the solution $\mathbf{b} = \mathbf{c}$ can be realized by setting $b_n^+ = c_n$ and $b_n^- = 0$ for $n \in \mathcal{D}$ and $c_n \geq 0$, $b_n^- = |c_n|$ and $b_n^+ = 0$ for $n \in \mathcal{D}$ and $c_n < 0$, $\mathbf{b}_{\mathcal{P}} = \mathbf{c}_{\mathcal{P}}$, and $\mathbf{b}_{\mathcal{N}} = \mathbf{c}_{\mathcal{N}}$. The resulting objective value is

$$\sum_{\substack{n\in\mathcal{D}\\c_n>0}} w_n^+ c_n + \sum_{\substack{n\in\mathcal{D}\\c_n<0}} w_n^- |c_n| + \sum_{n\in\mathcal{P}} w_n(c_n - \underline{B}_n) + \sum_{n\in\mathcal{N}} w_n(\overline{B}_n - c_n), \tag{6.3.9}$$

neglecting the constant terms for the moment.

To gain further insight, we derive an upper bound on (6.3.9), working with each summation separately. Using (6.3.2), each term in the first summation can be bounded from above as follows:

$$w_n^+ c_n \leq \frac{c_n}{\overline{B}_n} \left\lceil \log_2(1 + \overline{B}_n) \right\rceil .$$

We proceed to show that the ratio $c_n/\overline{B}_n$ is less than 1/2 for $n \in \mathcal{D}$ and $c_n > 0$. The fact that $n \in \mathcal{D}$ implies

$$-1 \geq \underline{B}_n = \left\lceil c_n - \sqrt{\left(\mathbf{Q}^{-1}\right)_{nn}} \right\rceil \geq c_n - \sqrt{\left(\mathbf{Q}^{-1}\right)_{nn}}.$$

Rearranging and adding $c_n$ to both sides,

$$2c_n \leq c_n + \sqrt{\left(\mathbf{Q}^{-1}\right)_{nn}} - 1 < \left\lfloor c_n + \sqrt{\left(\mathbf{Q}^{-1}\right)_{nn}} \right\rfloor = \overline{B}_n,$$

and hence $c_n/\overline{B}_n < 1/2$. Similarly, we use (6.3.3) to bound each term in the second summation in (6.3.9):

$$w_n^- \left|c_n\right| \leq \frac{\left|c_n\right|}{\left|\underline{B}_n\right|} \left\lceil \log_2(1 + \left|\underline{B}_n\right|) \right\rceil < \frac{1}{2} \left\lceil \log_2(1 + \left|\underline{B}_n\right|) \right\rceil ,$$

where the inequality $\left|c_n\right| / \left|\underline{B}_n\right| < 1/2$ parallels $c_n/\overline{B}_n < 1/2$ in the first case.

For terms in the third summation in (6.3.9), we use (6.3.4) to obtain

$$w_n(c_n - \underline{B}_n) \leq \frac{c_n - \underline{B}_n}{\overline{B}_n - \underline{B}_n} \left( \left\lceil \log_2(1 + \overline{B}_n) \right\rceil - \left\lceil \log_2(1 + \underline{B}_n) \right\rceil \right) .$$

The fraction $\frac{c_n - \underline{B}_n}{\overline{B}_n - \underline{B}_n}$ can be bounded as follows:

$$\begin{aligned} \overline{B}_n - \underline{B}_n &= \left\lfloor c_n + \sqrt{\left(\mathbf{Q}^{-1}\right)_{nn}} \right\rfloor - \underline{B}_n \\ &> c_n + \sqrt{\left(\mathbf{Q}^{-1}\right)_{nn}} - 1 - \underline{B}_n \\ &\geq 2(c_n - \underline{B}_n) - 1, \end{aligned}$$

where the last inequality follows from $c_n - \underline{B}_n \leq \sqrt{\left(\mathbf{Q}^{-1}\right)_{nn}}$. A rearrangement yields

$$\frac{c_n - \underline{B}_n}{\overline{B}_n - \underline{B}_n} < \frac{1}{2} \left( 1 + \frac{1}{\overline{B}_n - \underline{B}_n} \right) .$$

Similarly for terms in the fourth summation in (6.3.9), we can show that

$$w_n(\overline{B}_n - c_n) < \frac{1}{2}\left(1 + \frac{1}{\overline{B}_n - \underline{B}_n}\right)\left(\lceil \log_2(1 + |\underline{B}_n|)\rceil - \lceil \log_2(1 + |\overline{B}_n|)\rceil\right).$$

Combining the results of the previous two paragraphs with (6.3.9), we arrive at the following upper bound on the optimal value of (6.3.6):

$$\begin{aligned}
&\frac{1}{2}\left(1 - \frac{1}{\overline{B}_n - \underline{B}_n}\right)\left[\sum_{n\in\mathcal{P}} \lceil \log_2(1 + \underline{B}_n)\rceil + \sum_{n\in\mathcal{N}} \lceil \log_2(1 + |\overline{B}_n|)\rceil\right] \\
&+ \frac{1}{2}\left[\sum_{\substack{n\in\mathcal{D}\\ c_n>0}} \lceil \log_2(1 + \overline{B}_n)\rceil + \sum_{\substack{n\in\mathcal{D}\\ c_n<0}} \lceil \log_2(1 + |\underline{B}_n|)\rceil\right] \\
&+ \frac{1}{2}\left(1 + \frac{1}{\overline{B}_n - \underline{B}_n}\right)\left[\sum_{n\in\mathcal{P}} \lceil \log_2(1 + \overline{B}_n)\rceil + \sum_{n\in\mathcal{N}} \lceil \log_2(1 + |\underline{B}_n|)\rceil\right],
\end{aligned} \tag{6.3.10}$$

where we have also restored the constant terms in (6.3.6). The quantity in the first pair of square brackets in (6.3.10) is the lower bound in (6.2.1) based on candidate ranges. The sum of the second and third bracketed quantities is the corresponding upper bound based on candidate ranges, i.e., the result of independently maximizing $C_{\text{NLZ}}(b_n)$ for each coefficient over its candidate range. To see this more explicitly, observe that for $n \in \mathcal{D}$ and $c_n > 0$ and for $n \in \mathcal{P}$, the largest candidate value is $b_n = \overline{B}_n$ and this value also has the highest cost, $\lceil \log_2(1 + \overline{B}_n)\rceil$. Similarly for $n \in \mathcal{D}$, $c_n < 0$ and for $n \in \mathcal{N}$, $\underline{B}_n$ is the costliest value. In the limit $\overline{B}_n - \underline{B}_n \gg 1$, (6.3.10) approaches the midpoint between the lower and upper bounds based on candidate ranges. This result is analogous to the upper bound of $N/2$ on (3.3.8) that was derived in Section 3.3.3. Since it was assumed throughout Chapters 2 and 3 that (2.3.3) is satisfied for all $n$, the candidate range for each coefficient (same as its feasible range) includes the value zero and the lower bound on (2.0.1) based solely on candidate ranges is equal to zero. The corresponding upper bound based on candidate ranges is $N$ and the midpoint between the two is $N/2$.

As in Section 3.3.3, the upper bound in (6.3.10) can be strengthened by scaling the solution $\mathbf{b} = \mathbf{c}$ so that it lies on the boundary of the feasible set. The scale factor $\theta$ is limited by three classes of constraints in (6.3.6): the quadratic constraint, the constraints $b_n \geq \underline{B}_n$ for $n \in \mathcal{P}$, and the constraints $b_n \leq \overline{B}_n$ for $n \in \mathcal{N}$. It follows that $\theta$ can be chosen as

$$\theta = \max\left\{1 - \sqrt{\frac{\gamma}{\mathbf{c}^T\mathbf{Q}\mathbf{c}}}, \left\{\frac{\underline{B}_n}{c_n} : n \in \mathcal{P}\right\}, \left\{\frac{\overline{B}_n}{c_n} : n \in \mathcal{N}\right\}\right\}$$

and the upper bound in (6.3.10) can be reduced by the factor $\theta$.

In Section 3.3.3, the upper bound of $N/2$ (scaled by $\theta$) was interpreted as a limitation on the approximation capability of the linear relaxation. By analogy, it would seem that the bound in (6.3.10) (also scaled by $\theta$) also represents a shortcoming of the current linear relaxation (6.3.6). It will be seen in Section 6.6 however that the bound does not appear to prevent the linear relaxation from providing good approximations in the present case.

### 6.3.3 Linear relaxation of the SPT minimization problem

In this subsection, we develop a linear relaxation of the SPT minimization problem (5.1.2). While it is possible to use the same convex underestimator technique as in Section 6.3.1, the non-monotonicity of the cost function $C_{\text{SPT}}(b_n)$ leads to smaller values for the slopes of the piecewise linear underestimators. Therefore the resulting relaxation is not as strong as the linear relaxation (6.3.6) of problem (5.1.1). We use instead an approach due to [64] in which each digit in the CSD representation of the coefficients is associated with a pair of binary-valued variables and the binary constraints are then relaxed. This alternative approach also yields a convex relaxation with a linear cost function.

Following [64], we represent each coefficient $b_n$ as

$$b_n = \sum_{p=0}^{P-1} \left(s_{np}^+ - s_{np}^-\right) 2^p, \tag{6.3.11}$$

where $s_{np}^+$ and $s_{np}^-$ are binary-valued variables. Thus each digit $s_{np} = s_{np}^+ - s_{np}^-$ can take on values of 0, +1, and −1. Using this representation, problem (5.1.2) can be reformulated as

$$\begin{aligned}
\min_{\{s_{np}^+\},\{s_{np}^-\}} \quad & \sum_{n=1}^{N}\sum_{p=0}^{P-1} \left(s_{np}^+ + s_{np}^-\right) \\
\text{s.t.} \quad & (\mathbf{b}-\mathbf{c})^T\mathbf{Q}(\mathbf{b}-\mathbf{c}) \le \gamma, \\
& b_n = \sum_{p=0}^{P-1} \left(s_{np}^+ - s_{np}^-\right) 2^p, \quad n = 1, \dots, N, \\
& s_{np}^+,\ s_{np}^- \in \{0, 1\}, \quad n = 1, \dots, N, \quad p = 0, \dots, P-1.
\end{aligned} \tag{6.3.12}$$

The form of the cost function implies that the configuration $s_{np}^{+} = s_{np}^{-} = 1$ will never occur at an optimal solution because the equivalent configuration $s_{np}^{+} = s_{np}^{-} = 0$ has a lower cost. It follows that for every pair of indices $n$ and $p$, at least one of $s_{np}^{+}$, $s_{np}^{-}$ is zero and hence the cost function counts the number of non-zero digits just as in (5.1.2). The non-adjacency property of the CSD representation can also be imposed by adding the constraints

$$s_{np}^{+} + s_{np}^{-} + s_{n,p+1}^{+} + s_{n,p+1}^{-} \leq 1, \quad n = 1, \ldots, N, \quad p = 0, \ldots, P-2. \tag{6.3.13}$$

As discussed in [64], the values of some of the variables $s_{np}^{+}$ and $s_{np}^{-}$ in (6.3.12) can be fixed given knowledge of the candidate range $\underline{B}_n, \ldots, \overline{B}_n$ for each coefficient. Toward this end, we determine the CSD representation of every candidate value. If the digit $s_{np} \geq 0$ for all candidate values, then $s_{np}^{-}$ can be fixed to zero in (6.3.12), and likewise for $s_{np}^{+}$ if $s_{np} \leq 0$. In addition, if $s_{np} = 1$ for all candidate values, then $s_{np}^{+}$ can be fixed to 1, which is equivalent to subtracting the corresponding power of two from $c_n$, i.e., $c_n$ becomes $\widetilde{c}_n = c_n - 2^p$. Similarly if $s_{np} = -1$ for all values, $s_{np}^{-}$ is fixed to 1 and $c_n$ becomes $\widetilde{c}_n = c_n + 2^p$. We define the following subsets for later reference: for each $n$, $\mathcal{Z}_n^{+}$ and $\mathcal{Z}_n^{-}$ are the subsets of powers $p$ for which $s_{np}^{+}$ and $s_{np}^{-}$ respectively have been fixed to 0, $\mathcal{U}_n^{\pm}$ is the subset of powers for which $s_{np}^{\pm}$ has been fixed to 1, and $\mathcal{F}_n^{\pm}$ is the subset corresponding to the remaining $s_{np}^{\pm}$ with no fixed value.

A linear relaxation of (6.3.12) results from relaxing the binary constraints on $s_{np}^{+}$ and $s_{np}^{-}$ and replacing them with the interval constraints $0 \leq s_{np}^{+} \leq 1$, $0 \leq s_{np}^{-} \leq 1$. To ensure that the lower bound provided by the linear relaxation is at least as large as the lower bound in (6.2.2) based on candidate ranges, we introduce an additional set of constraints to the formulation of [64] presented thus far. Specifically, we require that for each $n$, the sum of $s_{np}^{+}$ and $s_{np}^{-}$ over $p$ be no smaller than the $n$th term in (6.2.2), i.e.,

$$\left|\mathcal{U}_n^{+}\right| + \left|\mathcal{U}_n^{-}\right| + \sum_{p \in \mathcal{F}_n^{+}} s_{np}^{+} + \sum_{p \in \mathcal{F}_n^{-}} s_{np}^{-} \geq \min_{b_n = \underline{B}_n, \ldots, \overline{B}_n} C_{\mathrm{SPT}}(b_n) \quad \forall\, n : 0 \notin \underline{B}_n, \ldots, \overline{B}_n, \tag{6.3.14}$$

taking into account those variables $s_{np}^{\pm}$ with fixed values. Only the indices $n$ for which zero is not a candidate value are included in (6.3.14). If zero is a candidate value, the right-hand side of (6.3.14) is zero, but the left-hand side is already guaranteed to be non-negative.

By incorporating constraints (6.3.13) and (6.3.14) and the values of variables that have been fixed, we obtain the following relaxation:

$$
\begin{aligned}
&\sum_{n=1}^{N}\left(\left|\mathcal{U}_n^{+}\right| + \left|\mathcal{U}_n^{-}\right|\right) + \\
&\min_{\{s_{np}^{+}\},\{s_{np}^{-}\}} \quad \sum_{n=1}^{N}\left(\sum_{p\in\mathcal{F}_n^{+}} s_{np}^{+} + \sum_{p\in\mathcal{F}_n^{-}} s_{np}^{-}\right) \\
&\qquad\quad \text{s.t.} \quad (\mathbf{b}-\widetilde{\mathbf{c}})^T\mathbf{Q}(\mathbf{b}-\widetilde{\mathbf{c}}) \le \gamma, \\
&\qquad\qquad\quad b_n = \sum_{p\in\mathcal{F}_n^{+}} s_{np}^{+}2^p - \sum_{p\in\mathcal{F}_n^{-}} s_{np}^{-}2^p, \quad n = 1,\dots,N, \\
&\qquad\qquad\quad s_{np}^{+} + s_{np}^{-} + s_{n,p+1}^{+} + s_{n,p+1}^{-} \le 1, \quad n = 1,\dots,N, \quad p = 0,\dots,P-2, \\
&\qquad\qquad\quad \left|\mathcal{U}_n^{+}\right| + \left|\mathcal{U}_n^{-}\right| + \sum_{p\in\mathcal{F}_n^{+}} s_{np}^{+} + \sum_{p\in\mathcal{F}_n^{-}} s_{np}^{-} \ge \min_{b_n=\underline{B}_n,\dots,\overline{B}_n} C_{\mathrm{SPT}}(b_n) \quad \forall\, n : 0 \notin \underline{B}_n,\dots,\overline{B}_n, \\
&\qquad\qquad\quad 0 \le s_{np}^{\pm} \le 1, \quad n = 1,\dots,N, \quad p \in \mathcal{F}_n^{\pm}, \\
&\qquad\qquad\quad s_{np}^{\pm} = 0, \quad n = 1,\dots,N, \quad p \in \mathcal{Z}_n^{\pm}, \\
&\qquad\qquad\quad s_{np}^{\pm} = 1, \quad n = 1,\dots,N, \quad p \in \mathcal{U}_n^{\pm},
\end{aligned}
\tag{6.3.15}
$$

where the vector $\widetilde{\mathbf{c}}$ is defined by

$$
\widetilde{c}_n = c_n - \sum_{p\in\mathcal{U}_n^{+}} 2^p + \sum_{p\in\mathcal{U}_n^{-}} 2^p
$$

based on the modifications to $\mathbf{c}$ described earlier. Similar to (6.3.6), problem (6.3.15) is a convex optimization problem with a linear cost function and quadratic and linear constraints. The ceiling of the optimal value of (6.3.15) is a lower bound on the optimal value of (6.3.12) (equivalently (5.1.2)). The lower bound property follows from the fact that the addition of constraints (6.3.13) and (6.3.14) to (6.3.12) does not change its optimal value (recall that the CSD representation is minimal in terms of the number of SPTs so (6.3.13) does not effectively impose any new restrictions), whereas the relaxation of the binary constraints decreases the optimal value. Henceforth we will refer to (6.3.15) as the linear relaxation of (5.1.2).

Given that the variables $s_{np}^{\pm}$ are permitted to take fractional values in the linear relaxation (6.3.15), an optimal solution to (6.3.15) tends to have the following property: for each

$n$, $s^{\pm}_{np}$ is non-zero for one or two large values of $p$ and is zero for the remaining values of $p$. This behavior is a consequence of the equal weighting given to all $s^{\pm}_{np}$ in the cost function, combined with the unequal weighting by powers of two in forming the coefficients $b_n$. It is less costly therefore to realize a given value for $b_n$ using higher-order digits than with lower-order digits. With a large number of zero-valued digits, the optimal value of the linear relaxation tends to be significantly lower than that of the original problem (5.1.2), where $s^{\pm}_{np}$ must be binary-valued. Constraints (6.3.13) and (6.3.14) are intended to partially compensate for the looseness of the relaxation, in the first case by limiting the use of adjacent digits and in the second by ensuring that the optimal value of the relaxation is at least as large as the lower bound in (6.2.2).

As in Section 6.3.1, the dual of the linear relaxation (6.3.15) tends to be easier to solve because it has only linear constraints. To derive and formulate the dual more compactly, we introduce the following notation. First we form the vector $\mathbf{s}$ by concatenating $N$ vectors $\mathbf{s}_n$ of the form

$$\mathbf{s}_n = \begin{bmatrix} s^{+}_{n0} & \dots & s^{+}_{n,P-1} & s^{-}_{n0} & \dots & s^{-}_{n,P-1} \end{bmatrix}^T, \tag{6.3.16}$$

so that each subvector $\mathbf{s}_n$ corresponds to one of the coefficients. We include in (6.3.16) only the powers $p$ in $\mathcal{F}^{+}_{n}$ and $\mathcal{F}^{-}_{n}$ since the other powers correspond to fixed variables. Next we define a block-diagonal matrix $\mathbf{P} = \mathrm{Diag}(\mathbf{P}_1, \dots, \mathbf{P}_N)$, where the $n$th block $\mathbf{P}_n$ is a row vector consisting of the powers of two that make up the $n$th coefficient. More specifically,

$$\mathbf{P}_n = \begin{bmatrix} 2^0 & 2^1 & \dots & 2^{P-1} & -2^0 & -2^1 & \dots & -2^{P-1} \end{bmatrix}, \tag{6.3.17}$$

including as in (6.3.16) only the powers $p$ in $\mathcal{F}^{+}_{n}$ and $\mathcal{F}^{-}_{n}$. It can be seen from (6.3.11), (6.3.16), and (6.3.17) that the coefficient vector $\mathbf{b}$ is equal to $\mathbf{Ps}$. We also define a block-diagonal matrix $\mathbf{J} = \mathrm{Diag}(\mathbf{J}_1, \dots, \mathbf{J}_n)$ to represent constraint (6.3.13), where each block is associated with a particular index $n$. The submatrix $\mathbf{J}_n$ is formed by starting with the $(P-1) \times P$ matrix

$$P-1 \left\{ \underbrace{\begin{bmatrix} 1 & 1 & & & \\ & 1 & 1 & & \\ & & \ddots & \ddots & \\ & & & 1 & 1 \end{bmatrix}}_{P} \right. ,$$

associating the columns with the powers $p = 0, \ldots, P-1$, extracting the columns corresponding to $\mathcal{F}_n^+$, extracting the columns corresponding to $\mathcal{F}_n^-$, and horizontally concatenating the two sets of extracted columns to yield a $(P-1) \times (|\mathcal{F}_n^+| + |\mathcal{F}_n^-|)$ matrix compatible with $\mathbf{s}_n$. We may also remove from $\mathbf{J}_n$ any rows containing all zeros or a single 1 since these rows represent the redundant constraints $0 \leq 1$ and $s_{np}^{\pm} \leq 1$. Similarly for constraint (6.3.14), we define a block-diagonal matrix $\mathbf{F} = \operatorname{Diag}\left(\left\{\mathbf{F}_n,\ n : 0 \notin \underline{B}_n, \ldots, \overline{B}_n\right\}\right)$ where each submatrix $\mathbf{F}_n$ is a row vector composed of $|\mathcal{F}_n^+| + |\mathcal{F}_n^-|$ ones. To represent the constant terms in constraint (6.3.14), we use the column vector $\boldsymbol{\ell}$ with components

$$\ell_n = \min_{b_n = \underline{B}_n, \ldots, \overline{B}_n} C_{\mathrm{SPT}}(b_n) - \left|\mathcal{U}_n^+\right| - \left|\mathcal{U}_n^-\right| \quad \forall\, n : 0 \notin \underline{B}_n, \ldots, \overline{B}_n.$$

Given the definitions above and neglecting the constant term in the cost function of (6.3.15), it is shown in Appendix D.2 that the dual problem can be formulated as follows:

$$\begin{aligned} \max_{\boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{\pi}^{\pm}, \boldsymbol{\rho}} \quad & \tilde{\mathbf{c}}^T \boldsymbol{\rho} - \sqrt{\boldsymbol{\rho}^T \mathbf{Q}^{-1} \boldsymbol{\rho}} - \mathbf{e}^T \boldsymbol{\mu} + \boldsymbol{\ell}^T \boldsymbol{\nu} - \mathbf{e}^T \boldsymbol{\pi}^+ \\ \text{s.t.} \quad & \mathbf{e} + \mathbf{J}^T \boldsymbol{\mu} - \mathbf{F}^T \boldsymbol{\nu} + \boldsymbol{\pi}^+ - \boldsymbol{\pi}^- - \mathbf{P}^T \boldsymbol{\rho} = \mathbf{0}, \\ & \boldsymbol{\mu} \geq \mathbf{0}, \quad \boldsymbol{\nu} \geq \mathbf{0}, \quad \boldsymbol{\pi}^{\pm} \geq \mathbf{0}. \end{aligned} \tag{6.3.18}$$

The dimensions of the dual variables $\boldsymbol{\mu}$, $\boldsymbol{\nu}$, $\boldsymbol{\pi}^{\pm}$, and $\boldsymbol{\rho}$ can be inferred from the dimensions of $\mathbf{P}$, $\mathbf{J}$, and $\mathbf{F}$. The dual problem is a convex optimization problem with the same optimal value as (6.3.15) and only linear constraints.

## 6.4 Diagonal relaxation

In this section, we revisit the strategy of Section 3.4 in which the matrix $\mathbf{Q}$ is replaced by a diagonal matrix to yield a relaxation of the original problem. As we saw in Sections 5.2.1 and 5.2.2, problems (5.1.1) and (5.1.2) can be solved efficiently in the diagonal case. Thus diagonal relaxations are computationally tractable for bit-efficient design as well. Unlike in Section 3.4 however, it is difficult to determine a diagonal relaxation that is maximally tight in terms of the cost functions $C_{\mathrm{NLZ}}(\mathbf{b})$ and $C_{\mathrm{SPT}}(\mathbf{b})$. We use instead a substitute criterion based on the volume ratio, which we elaborate upon further in Section 6.4.1.

To ensure that the lower bound resulting from diagonal relaxation is at least as strong as the bounds in (6.2.1) and (6.2.2) based on candidate ranges, we incorporate the interval constraints $\underline{B}_n \leq b_n \leq \overline{B}_n$ in the relaxed problem. The additional constraints can be accommodated by making minor modifications to the algorithms of Sections 5.2.1 and 5.2.2. These modifications are described in Section 6.4.2.

In the remainder of the section, we analyze the approximation properties of the diagonal relaxation and develop results that parallel those of Section 3.4. The key difference is that the results in this section are stated in terms of the volume ratio and not the true cost function ($C_{\mathrm{NLZ}}(\mathbf{b})$ or $C_{\mathrm{SPT}}(\mathbf{b})$). In Section 6.4.3, we show that the optimal volume ratio is invariant under diagonal scaling of the original ellipsoid $\mathcal{E}_{\mathbf{Q}}$. In Section 6.4.4, the optimal volume ratio is bounded in terms of the eigenvalues of $\mathbf{Q}$ and this bound is shown to be tight. In Sections 6.4.5 and 6.4.6, we obtain bounds on the volume ratio in the diagonally dominant and nearly coordinate-aligned cases. Together these results help identify problem instances for which diagonal relaxation is expected to yield a good approximation.

### 6.4.1 Minimum volume criterion

A diagonal relaxation can be obtained in the same way as in Section 3.4.1. We replace the quadratic constraint (2.1.1) by a similar constraint involving a positive definite diagonal matrix $\mathbf{D}$ such that $\mathbf{D} \preceq \mathbf{Q}$. The last condition ensures that the original ellipsoid $\mathcal{E}_{\mathbf{Q}}$ is contained within the coordinate-aligned ellipsoid $\mathcal{E}_{\mathbf{D}}$ so that minimizing over $\mathcal{E}_{\mathbf{D}}$ results in a lower bound on the original optimal cost. Different values for $\mathbf{D}$ correspond to different diagonal relaxations as depicted in Fig. 3-4. Ideally, $\mathbf{D}$ should be chosen such that the minimum value of the cost function $C_{\mathrm{NLZ}}(\mathbf{b})$ or $C_{\mathrm{SPT}}(\mathbf{b})$ over $\mathcal{E}_{\mathbf{D}}$ is largest among all valid choices of $\mathbf{D}$. The determination of an optimal $\mathbf{D}$ was tractable in Section 3.4.1 because the minimum zero-norm over $\mathcal{E}_{\mathbf{D}}$ depends on an explicit and relatively simple function of $\mathbf{D}$, namely $\Sigma_K\left(\{D_{nn}c_n^2\}\right)$, the sum of the $K$ smallest $D_{nn}c_n^2$. By solving problem (3.4.3) for several values of $K$, the tightest possible diagonal relaxation could be determined. In contrast, the solutions to problems (5.1.1) and (5.1.2) are algorithmic even in the diagonal case as seen in Sections 5.2.1 and 5.2.2, and the dependence of the optimal values on $\mathbf{D}$ is unclear. Therefore it is difficult in the present context to determine the diagonal relaxation that yields the best possible lower bound.

Given the difficulty of obtaining an optimal diagonal relaxation, in the remainder of this section we use the volume of the ellipsoid $\mathcal{E}_{\mathbf{D}}$ as a substitute measure of the strength of the relaxation. The volume of $\mathcal{E}_{\mathbf{D}}$ is a reasonable criterion since it is approximately proportional to the number of integer solutions within $\mathcal{E}_{\mathbf{D}}$, each of which occupies one unit of volume. The smaller the volume, the fewer the number of solutions, and the higher the minimum cost over $\mathcal{E}_{\mathbf{D}}$ tends to be. Since the volume of $\mathcal{E}_{\mathbf{D}}$ is proportional to $\sqrt{\det\left(\mathbf{D}^{-1}\right)}$ as seen in (6.2.3), the enclosing ellipsoid of minimal volume can be determined by solving

$$\begin{aligned} \max_{\mathbf{D}} \quad & \det(\mathbf{D}) \\ \text{s.t.} \quad & \mathbf{0} \preceq \mathbf{D} \preceq \mathbf{Q}, \\ & \mathbf{D} \text{ diagonal}, \end{aligned} \tag{6.4.1}$$

where we have used the property $\det\left(\mathbf{D}^{-1}\right) = 1/\det(\mathbf{D})$. Problem (6.4.1) is a determinant maximization problem, which is closely related to semidefinite optimization. The solution of (6.4.1) is discussed in Section 6.5. In the sequel, we will assume that $\mathbf{D}$ has been determined from (6.4.1).

As with linear relaxations in Section 6.3, it is desirable for the lower bounds resulting from diagonal relaxations to be stronger than the lower bounds in (6.2.1) and (6.2.2) based on candidate ranges. This can be guaranteed by including the interval constraints $\underline{B}_n \leq b_n \leq \overline{B}_n$ in the relaxed problem. Thus we define

$$\begin{aligned} \min_{\mathbf{b}} \quad & \sum_{n=1}^{N} C_{\text{NLZ}}(b_n) \\ \text{s.t.} \quad & \sum_{n=1}^{N} D_{nn}(b_n - c_n)^2 \leq \gamma, \\ & \underline{B}_n \leq b_n \leq \overline{B}_n, \quad n = 1, \ldots, N, \\ & b_n \in \mathbb{Z}, \quad n = 1, \ldots, N \end{aligned} \tag{6.4.2}$$

as the diagonal relaxation of problem (5.1.1), and similarly

$$\begin{aligned} \min_{\mathbf{b}} \quad & \sum_{n=1}^{N} C_{\text{SPT}}(b_n) \\ \text{s.t.} \quad & \sum_{n=1}^{N} D_{nn}(b_n - c_n)^2 \leq \gamma, \\ & \underline{B}_n \leq b_n \leq \overline{B}_n, \quad n = 1, \ldots, N, \\ & b_n \in \mathbb{Z}, \quad n = 1, \ldots, N \end{aligned} \tag{6.4.3}$$

for (5.1.2). In Fig. 6-5, we interpret the feasible set for the diagonal relaxations as the set of integer points in the intersection of $\mathcal{E}_{\mathbf{D}}$ and the box $\mathcal{B}_{\mathbf{Q}}$ created by the candidate ranges $\underline{B}_n, \dots, \overline{B}_n$. Note that the box constraint can exclude integer points that would otherwise be included in $\mathcal{E}_{\mathbf{D}}$, and vice versa. It follows that the bound obtained by solving (6.4.2) is at least as large as the bound in (6.2.1) obtained by minimizing over all of $\mathcal{B}_{\mathbf{Q}}$, and likewise for (6.2.2) and (6.4.3). The addition of the interval constraints requires some modifications to the algorithms developed in Sections 5.2.1 and 5.2.2 for the diagonal case. These modifications are described in the next subsection.

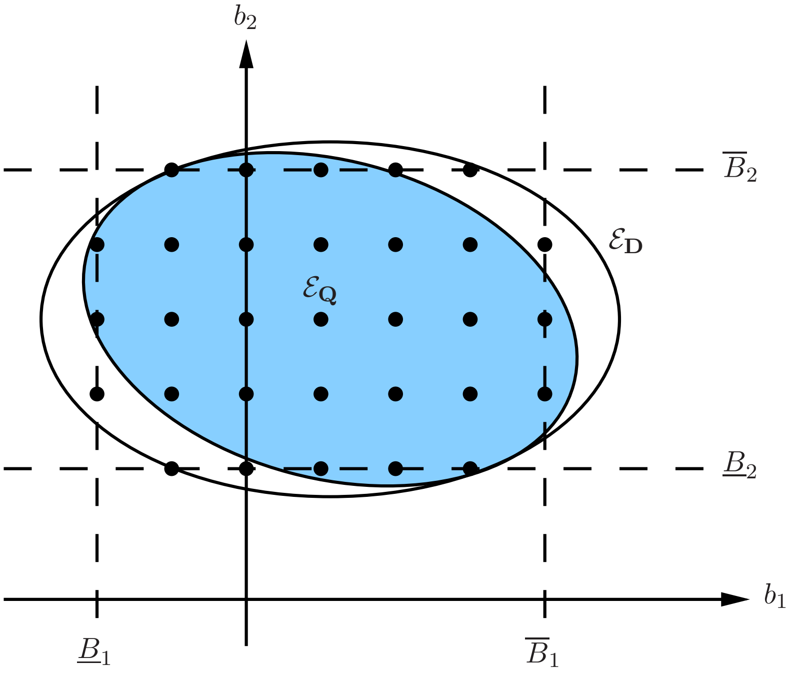


Figure 6-5: The feasible set for the diagonal relaxations (6.4.2) and (6.4.3) is the set of integer points in the intersection of the coordinate-aligned ellipsoid $\mathcal{E}_{\mathbf{D}}$ and the box $\mathcal{B}_{\mathbf{Q}}$. As the number of dimensions increases, the tightness of the enclosing ellipsoid tends to increase relative to that of the box.

### 6.4.2 Algorithms for solving diagonal relaxations

We start with the solution to problem (6.4.3) as it is simpler and similar to that discussed in Section 5.2.2. As before, we define $E_{\mathrm{SPT}}(B)$ to be the minimum quantization error subject to a total cost of $B$ SPTs, but now also subject to the interval constraints, i.e.,

$$E_{\mathrm{SPT}}(B) = \min \quad \sum_{n=1}^{N} D_{nn}(b_n - c_n)^2 \tag{6.4.4}$$

$$\begin{aligned}\text{s.t.} \quad & \sum_{n=1}^{N} C_{\text{SPT}}(b_n) = B, \\ & \underline{B}_n \leq b_n \leq \overline{B}_n \quad \forall\, n, \\ & b_n \in \mathbb{Z} \quad \forall\, n.\end{aligned}$$

Problem (6.4.3) is solved by determining the smallest value of $B$ such that $E_{\text{SPT}}(B) \leq \gamma$. Compared to (5.2.9), the presence of the interval constraints in (6.4.4) may necessitate a non-zero initial allocation of SPTs to ensure feasibility. The initial allocation $a_1, \ldots, a_N$ is determined according to

$$a_n = \min_{b_n = \underline{B}_n, \ldots, \overline{B}_n} C_{\text{SPT}}(b_n), \quad n = 1, \ldots, N. \tag{6.4.5}$$

Given $a_1, \ldots, a_N$, we determine the initial coefficient values $b_n$ and the scalar quantization errors $u_n(a_n)$ from (5.2.7). Then we set $E_{\text{SPT}}(B) = \sum_n u_n(a_n)$, where $B = \sum_n a_n$. If $E_{\text{SPT}}(B) \leq \gamma$, we are done. Otherwise, the algorithm proceeds exactly as in Section 5.2.2 since the key monotonicity condition (5.2.8) is still satisfied. In each iteration, the coefficient $b_m$ for which the number of SPTs should increase is determined from (5.2.10), $a_m$ is incremented by 1, and $E_{\text{SPT}}(B+1) = E_{\text{SPT}}(B) - \Delta u_m(a_m + 1)$. The interval constraints in (6.4.4) are no longer a concern because the value of $b_n$ can only approach $c_n$, the approximate midpoint of the candidate range, as more SPTs are allocated. We summarize the algorithm in Algorithm 5.

**Algorithm 5** Solution to the diagonal relaxation (6.4.3) of problem (5.1.2).

```
Input: Parameters c, $\gamma$ from (5.1.2), D from (6.4.1), $\underline{B}_n$, $\overline{B}_n$ from (6.1.1).
Output: Optimal solution b to (6.4.3), optimal cost $B$.
  Determine initial allocation $a_1, \ldots, a_N$ from (6.4.5).
  for $n = 1 \ldots, N$ do
    Determine $b_n$ and $u_n(a_n)$ from (5.2.7) with $B = a_n$.
  $B = \sum_n a_n$, $E_{\text{SPT}}(B) = \sum_n u_n(a_n)$.
  while $E_{\text{SPT}}(B) > \gamma$ do
    Determine $m$ from (5.2.10) and (5.2.7).
    $a_m \leftarrow a_m + 1$, update $b_m$ accordingly.
    $E_{\text{SPT}}(B+1) = E_{\text{SPT}}(B) - \Delta u_m(a_m + 1)$.
    $B \leftarrow B + 1$.
```

Next we discuss the solution of (6.4.2), which is similar to the dynamic programming solution of Section 5.2.1. The difference lies again in the minimum allowable allocation of bits to coefficients. We add the interval constraints $\underline{B}_n \leq b_n \leq \overline{B}_n$ to each of the definitions for $E_{\text{NLZ}}(B)$, $V_m(B)$, and $v_n(B)$ in (5.2.1), (5.2.2), and (5.2.3) respectively (substituting $\mathbf{D}$ for $\mathbf{Q}$ where necessary). With the additional constraint, (5.2.3) becomes infeasible unless $B$ is equal to or greater than

$$a_{n,\min} = \begin{cases} \left\lceil \log_2(1+\underline{B}_n) \right\rceil, & \underline{B}_n > 0, \\ \left\lceil \log_2(1+\left|\overline{B}_n\right|) \right\rceil, & \overline{B}_n < 0, \\ 0 & \text{otherwise}. \end{cases} \tag{6.4.6}$$

If $B < a_{n,\min}$, we may take $v_n(B) = \infty$. Hence (5.2.5) is modified as follows:

$$v_n(B) = \begin{cases} \infty, & B < a_{n,\min}, \\ D_{nn}(|c_n| - (2^B - 1))^2, & a_{n,\min} \leq B < \lceil \log_2(1+|c_n|) \rceil, \\ D_{nn}(c_n - [c_n])^2, & B \geq \lceil \log_2(1+|c_n|) \rceil. \end{cases} \tag{6.4.7}$$

The recursion (5.2.6) is modified accordingly:

$$V_m(B) = \min_{\substack{B'=a_{m,\min},\ldots,\min\{B,P\} \\ V_{m-1}(B-B')\leq\gamma}} \left\{ v_m(B') + V_{m-1}(B-B') \right\} \tag{6.4.8}$$

since $v_m(B') = \infty$ for $B' < a_{m,\min}$. It can also be seen that $V_m(B) = \infty$ for $B < \sum_{n=1}^{m} a_{n,\min}$, the minimum allowable allocation for the first $m$ coefficients.

With the new definitions in (6.4.7) and (6.4.8), the dynamic programming algorithm can proceed much as before. We start with the minimum allocation, set $B = \sum_{n=1}^{N} a_{n,\min}$, and evaluate the sequence $V_1(a_{1,\min}), V_2(a_{1,\min} + a_{2,\min}), \ldots, V_N\left(\sum_{n=1}^{N} a_{n,\min}\right) = E_{\text{NLZ}}(B)$ until either $E_{\text{NLZ}}(B) \leq \gamma$ or $V_m\left(\sum_{n=1}^{m} a_{n,\min}\right) > \gamma$ for $m < N$. In the second case, we increment $B$ by 1 and continue with the sequence $V_1(a_{1,\min}+1), V_2(a_{1,\min}+a_{2,\min}+1), \ldots, V_N\left(\left(\sum_{n=1}^{N} a_{n,\min}\right)+1\right) = E_{\text{NLZ}}(B)$. The first value of $B$ such that $E_{\text{NLZ}}(B) \leq \gamma$ is the optimal value of (6.4.2). A solution $\mathbf{b}$ that achieves the optimal value can be obtained through the same backtracking procedure discussed in Section 5.2.1. We summarize the dynamic programming algorithm under Algorithm 6. As with problem (6.4.3), the interval constraints are used only to determine the minimum allocation $a_{1,\min}, \ldots, a_{N,\min}$. They do not play a role thereafter because the value of $b_n$ can only move further inside the candidate range as the number of bits increases.

**Algorithm 6** Solution to the diagonal relaxation (6.4.2) of problem (5.1.1).

```
Input: Parameters $\mathbf{c}$, $\gamma$ from (5.1.1), $\mathbf{D}$ from (6.4.1), $\underline{B}_n$, $\overline{B}_n$ from (6.1.1).
Output: Optimal solution $\mathbf{b}$ to (6.4.2), optimal cost $B$.
  Initialize: Determine minimal allocation $a_{1,\min}, \ldots, a_{N,\min}$ from (6.4.6). Initialize
  $V_m(B) = \infty$ for all $m$, $B$.
  repeat
    if first iteration then
      $B_0 = 0$.
    else
      $B_0 \leftarrow B_0 + 1$.
    $m \leftarrow 1$.
    $B = B_0 + a_{m,\min}$.
    Evaluate $V_m(B) = v_m(B)$ using (6.4.7).
    $\widehat{a}_m(B) = B$ (for backtracking).
    while $V_m(B) \leq \gamma$ and $m < N$ do
      $m \leftarrow m + 1$.
      $B \leftarrow B + a_{m,\min}$.
      Evaluate $V_m(B)$ using (6.4.8).
      $\widehat{a}_m(B) =$ minimizing value of $B'$ corresponding to $V_m(B)$ (for backtracking).
    end while
  until $V_N(B) \leq \gamma$
  Backtrack to determine b: Initialize $B'' = B$.
  for $m = N, N-1, \ldots, 1$ do
    Allocate $\widehat{a}_m(B'')$ bits to coefficient $b_m$ and determine $b_m$ as discussed in paragraph
    preceding (5.2.5).
    $B'' \leftarrow B'' - \widehat{a}_m(B'')$.
```

### 6.4.3 Invariance of the optimal volume ratio under diagonal scaling

In Sections 6.4.4–6.4.6, we derive bounds on the ratio between the volume of the optimal enclosing ellipsoid $\mathcal{E}_{\mathbf{D}}$ and the volume of the original ellipsoid $\mathcal{E}_{\mathbf{Q}}$. The bounds on the optimal volume ratio characterize indirectly the strength of the diagonal relaxation. For simplicity, we neglect the effect of the interval constraints in (6.4.2) and (6.4.3) on the volume of the feasible set, although these constraints can sometimes have a significant impact. We show in this subsection that the volume ratio is invariant under diagonal scaling transformations of the ellipsoid $\mathcal{E}_{\mathbf{Q}}$. As in Section 3.4, this invariance property is used to strengthen some of the bounds in Sections 6.4.4–6.4.6.

We consider as in Section 3.4.3 the scaled ellipsoid $\mathbf{S}(\mathcal{E}_{\mathbf{Q}})$ for an arbitrary invertible diagonal matrix $\mathbf{S}$. The matrix $\mathbf{Q}$ is replaced by $\mathbf{S}^{-1}\mathbf{Q}\mathbf{S}^{-1}$ and the volume of $\mathbf{S}(\mathcal{E}_{\mathbf{Q}})$ is inversely proportional to $\sqrt{\det\left(\mathbf{S}^{-1}\mathbf{Q}\mathbf{S}^{-1}\right)} = \det\left(\mathbf{S}^{-1}\right)\sqrt{\det(\mathbf{Q})}$. Referring to (6.4.1), the minimum-volume enclosing ellipsoid for $\mathbf{S}(\mathcal{E}_{\mathbf{Q}})$ is determined by solving

$$\begin{aligned} \max_{\mathbf{D}} \quad & \det(\mathbf{D}) \\ \text{s.t.} \quad & \mathbf{0} \preceq \mathbf{D} \preceq \mathbf{S}^{-1}\mathbf{Q}\mathbf{S}^{-1}, \\ & \mathbf{D} \text{ diagonal.} \end{aligned} \tag{6.4.9}$$

Since $\mathbf{S}$ is invertible, problem (6.4.9) is equivalent to

$$\begin{aligned} \max_{\mathbf{D}} \quad & \det(\mathbf{D}) \\ \text{s.t.} \quad & \mathbf{0} \preceq \mathbf{SDS} \preceq \mathbf{Q}, \\ & \mathbf{D} \text{ diagonal,} \end{aligned}$$

which in turn is equivalent to

$$\begin{aligned} \det\left(\mathbf{S}^{-1}\right)^2 \max_{\mathbf{D}} \quad & \det(\mathbf{D}) \\ \text{s.t.} \quad & \mathbf{0} \preceq \mathbf{D} \preceq \mathbf{Q}, \\ & \mathbf{D} \text{ diagonal} \end{aligned} \tag{6.4.10}$$

under a change of variables. The factors of $\det\left(\mathbf{S}^{-1}\right)$ cancel in the volume ratio after taking the square root of the optimal value of (6.4.10). Thus the volume ratio between $\mathbf{S}(\mathcal{E}_{\mathbf{Q}})$ and its enclosing ellipsoid is the same as the corresponding ratio for $\mathcal{E}_{\mathbf{Q}}$.

### 6.4.4 Eigenvalue bound on the optimal volume ratio

We now relate the optimal volume ratio to the eigenvalues of the matrix $\mathbf{Q}$. The results that we obtain are analogous to the results in Section 3.4.4 for the sparsity maximization problem. According to the geometric intuition discussed in Section 3.4.4, the volume ratio is expected to be small when the condition number $\kappa(\mathbf{Q})$ is low, i.e., when the ellipsoid $\mathcal{E}_{\mathbf{Q}}$ is nearly spherical. In addition, we argued more informally and observed in the numerical experiments in Section 3.6 that eigenvalue distributions in which most of the eigenvalues are comparable to the smallest eigenvalue are preferred. The following bound on the volume ratio confirms this preference for certain eigenvalue distributions. Furthermore, the condition under which the bound is tight, namely that the eigenvector $\mathbf{v}_1$ corresponding to $\lambda_{\min}(\mathbf{Q})$ has equal-magnitude components, is the same as the one used to construct worst-case instances for diagonal relaxation in Section 3.4.2. This agrees with the intuition that the orientation of $\mathcal{E}_{\mathbf{Q}}$ is most strongly affected by the eigenvector $\mathbf{v}_1$.

**Theorem 5.** *The ratio between the volume of the ellipsoid $\mathcal{E}_{\mathbf{Q}}$ and that of the minimum-volume coordinate-aligned enclosing ellipsoid $\mathcal{E}_{\mathbf{D}}$ is bounded as follows:*

$$\frac{\operatorname{vol}(\mathcal{E}_{\mathbf{D}})}{\operatorname{vol}(\mathcal{E}_{\mathbf{Q}})} \leq \sqrt{\prod_{n=2}^{N} \frac{\lambda_n(\mathbf{Q})}{\lambda_{\min}(\mathbf{Q})}},$$

*where $\lambda_n(\mathbf{Q})$ is the nth smallest eigenvalue of $\mathbf{Q}$. Equality holds if the eigenvector $\mathbf{v}_1$ corresponding to the smallest eigenvalue $\lambda_{\min}(\mathbf{Q})$ has components of equal magnitude.*

*Proof.* To establish the bound, we note that $\mathbf{D} = \lambda_{\min}(\mathbf{Q})\mathbf{I}$ is a feasible solution to problem (6.4.1), and hence the maximum determinant is bounded from below by $\lambda_{\min}^N(\mathbf{Q})$. The bound in the theorem follows from the inverse proportionality between the volume and the square root of the determinant, and by expressing the determinant of $\mathbf{Q}$ as the product of its eigenvalues.

To show that the bound is tight, we refer to the inequality in (3.4.5), which holds for all $\mathbf{D}$ satisfying $\mathbf{0} \preceq \mathbf{D} \preceq \mathbf{Q}$ under the assumption that $\mathbf{v}_1$ has equal-magnitude components. By the arithmetic mean-geometric mean inequality, we obtain

$$\det(\mathbf{D})^{1/N} = \left(\prod_{n=1}^{N} D_{nn}\right)^{1/N} \leq \frac{1}{N}\sum_{n=1}^{N} D_{nn} \leq \lambda_{\min}(\mathbf{Q}) \quad \forall\, \mathbf{D} : \mathbf{0} \preceq \mathbf{D} \preceq \mathbf{Q},$$

where equality holds if $\mathbf{D} = \lambda_{\min}(\mathbf{Q})\mathbf{I}$. This shows that under the assumption on $\mathbf{v}_1$, the solution $\mathbf{D} = \lambda_{\min}(\mathbf{Q})\mathbf{I}$ maximizes the determinant in (6.4.1) and minimizes $\operatorname{vol}(\mathcal{E}_{\mathbf{D}})$, and therefore the bound in the theorem is met with equality. □

The bound in Theorem 5 depends on the ratio of each eigenvalue of $\mathbf{Q}$ to the smallest eigenvalue. Hence it exhibits a preference for eigenvalue distributions that are weighted toward smaller values, as claimed. Theorem 5 can also be interpreted as a bound on the equivalent dilation factor $\chi$, i.e., the factor by which $\mathcal{E}_{\mathbf{Q}}$ should be dilated to match $\mathcal{E}_{\mathbf{D}}$ in volume. Taking the $N$th root of both sides, we have

$$\chi \leq \sqrt{\left( \prod_{n=2}^{N} \frac{\lambda_n(\mathbf{Q})}{\lambda_{\min}(\mathbf{Q})} \right)^{1/N}}.$$

The right-hand side is the square root of the geometric mean of eigenvalue ratios.

As a corollary to Theorem 5, if the eigenvector $\mathbf{v}_1$ has equal-magnitude components and the condition $\lambda_{\min}(\mathbf{Q}) \|\mathbf{c}\|_2^2 \leq \gamma$ is satisfied, then the minimum values of the cost functions $C_{\mathrm{NLZ}}(\mathbf{b})$ and $C_{\mathrm{SPT}}(\mathbf{b})$ over the enclosing ellipsoid $\mathcal{E}_{\mathbf{D}}$ are equal to zero. It was shown in the proof of Theorem 2 that the optimal $\mathbf{D}$ in this case is $\mathbf{D} = \lambda_{\min}(\mathbf{Q})\mathbf{I}$. By substituting into (3.4.1), we see that the solution $\mathbf{b} = \mathbf{0}$, which has zero cost, belongs to $\mathcal{E}_{\mathbf{D}}$. Note that the presence of the interval constraints in what we have defined as the diagonal relaxations, namely (6.4.2) and (6.4.3), ensures that the resulting lower bounds are no worse than those in (6.2.1) and (6.2.2). In this worst-case example, the inclusion of the quadratic constraint in (6.4.2) and (6.4.3) offers no improvement.

A second corollary follows from the invariance of the optimal volume ratio to diagonal scaling, which was shown in Section 6.4.3. Although the volume ratio does not change, the right-hand side of the bound in Theorem 5 may vary as $\mathbf{Q}$ is replaced by $\mathbf{S}^{-1}\mathbf{Q}\mathbf{S}^{-1}$ for different choices of $\mathbf{S}$. Hence Theorem 5 can be generalized in the same way as Theorem 2.

**Corollary 2.** *For any invertible diagonal matrix* $\mathbf{S}$*, the optimal volume ratio is bounded as follows:*

$$\frac{\mathrm{vol}(\mathcal{E}_{\mathbf{D}})}{\mathrm{vol}(\mathcal{E}_{\mathbf{Q}})} \leq \sqrt{\prod_{n=2}^{N} \frac{\lambda_n\left(\mathbf{S}^{-1}\mathbf{Q}\mathbf{S}^{-1}\right)}{\lambda_{\min}\left(\mathbf{S}^{-1}\mathbf{Q}\mathbf{S}^{-1}\right)}}.$$

As with Corollary 1, $\mathbf{S}$ can be chosen to minimize the right-hand side.

### 6.4.5 The diagonally dominant case

Next we analyze the case of diagonally dominant $\mathbf{Q}$. We use the same definition of diagonal dominance as in Section 3.4.5 and follow a similar line of reasoning to relate the optimal volume ratio to the chosen measure of diagonal dominance. First we establish the following result, which is analogous to Lemma 3.

**Lemma 5.** *Fix a positive definite diagonal matrix* $\mathbf{D}_0$*, and let* $\mathbf{D} = \alpha\mathbf{D}_0$*. Then the optimal* *value of* (6.4.1) *is bounded from below by*

$$\lambda_{\min}^{N}\big(\mathbf{D}_0^{-1/2}\mathbf{Q}\mathbf{D}_0^{-1/2}\big)\prod_{n=1}^{N}(\mathbf{D}_0)_{nn}.$$

*Proof.* With $\mathbf{D} = \alpha\mathbf{D}_0$, (6.4.1) reduces to

$$\begin{aligned} \max_{\alpha} \quad & \alpha^N \prod_{n=1}^{N}(\mathbf{D}_0)_{nn} \\ \text{s.t.} \quad & \mathbf{0} \preceq \alpha\mathbf{D}_0 \preceq \mathbf{Q}. \end{aligned}$$

The rest of the proof is similar to that of Lemma 3. □

As in Section 3.4.5, we set $\mathbf{D}_0 = \operatorname{Diag}(\mathbf{Q})$ in the diagonally dominant case, where $\operatorname{Diag}(\mathbf{Q})$ is the diagonal matrix formed from the diagonal of $\mathbf{Q}$. Using Lemma 5, the optimal volume ratio can be bounded as follows:

$$\frac{\operatorname{vol}(\mathcal{E}_{\mathbf{D}})}{\operatorname{vol}(\mathcal{E}_{\mathbf{Q}})} \le \Big(\lambda_{\min}\big(\operatorname{Diag}(\mathbf{Q})^{-1/2}\mathbf{Q}\operatorname{Diag}(\mathbf{Q})^{-1/2}\big)\Big)^{-N/2}\sqrt{\frac{\det(\mathbf{Q})}{\prod_{n=1}^{N} Q_{nn}}}.$$

By Hadamard's inequality [92] and the positive definiteness of $\mathbf{Q}$, the quantity under the radical sign is no greater than unity. Hence

$$\frac{\operatorname{vol}(\mathcal{E}_{\mathbf{D}})}{\operatorname{vol}(\mathcal{E}_{\mathbf{Q}})} \le \Big(\lambda_{\min}\big(\operatorname{Diag}(\mathbf{Q})^{-1/2}\mathbf{Q}\operatorname{Diag}(\mathbf{Q})^{-1/2}\big)\Big)^{-N/2}.$$

Combining this with (3.4.14) yields the following result:

**Theorem 6.** *Assume that* $\mathbf{Q}$ *is diagonally dominant in the sense that*

$$\max_{m}\sum_{n\neq m}\frac{|Q_{mn}|}{\sqrt{Q_{mm}Q_{nn}}} < 1.$$

*Then the ratio between the volume of the ellipsoid* $\mathcal{E}_{\mathbf{Q}}$ *and that of the minimum-volume coordinate-aligned enclosing ellipsoid* $\mathcal{E}_{\mathbf{D}}$ *is bounded as follows:*

$$\frac{\operatorname{vol}(\mathcal{E}_{\mathbf{D}})}{\operatorname{vol}(\mathcal{E}_{\mathbf{Q}})} \le \left(1 - \max_{m}\sum_{n\neq m}\frac{|Q_{mn}|}{\sqrt{Q_{mm}Q_{nn}}}\right)^{-N/2}.$$

The equivalent bound on the dilation factor $\chi$ is

$$\chi \leq \left( 1 - \max_m \sum_{n \neq m} \frac{|Q_{mn}|}{\sqrt{Q_{mm} Q_{nn}}} \right)^{-1/2}.$$

As the degree of diagonal dominance increases, the quantity in parentheses approaches 1 from below and the bounds decrease. Similar to Theorem 3, replacing $\mathbf{Q}$ by $\mathbf{S}^{-1}\mathbf{Q}\mathbf{S}^{-1}$ has no effect on the measure of diagonal dominance or the bound on the optimal volume ratio.

### 6.4.6 The nearly coordinate-aligned case

We now turn to the case in which the eigenvectors of $\mathbf{Q}$ (equivalently the axes of the ellipsoid $\mathcal{E}_{\mathbf{Q}}$) are nearly aligned with the coordinate directions. We assume as in Section 3.4.6 that $\mathbf{Q}$ has a diagonalization $\mathbf{Q} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$ where the eigenvector matrix $\mathbf{V}$ is such that the spectral radius of $\mathbf{\Delta} = \mathbf{V} - \mathbf{I}$ is small. To bound the optimal volume ratio in terms of $\mathbf{\Delta}$, we let $\mathbf{D}_0 = \mathbf{\Lambda}$ in Lemma 5, which corresponds to restricting $\mathcal{E}_{\mathbf{D}}$ to be of the same shape as $\mathcal{E}_{\mathbf{Q}}$. This results in the bound

$$\frac{\operatorname{vol}(\mathcal{E}_{\mathbf{D}})}{\operatorname{vol}(\mathcal{E}_{\mathbf{Q}})} \leq \left( \lambda_{\min}\left(\mathbf{\Lambda}^{-1/2}\mathbf{Q}\mathbf{\Lambda}^{-1/2}\right) \right)^{-N/2} \sqrt{\frac{\det(\mathbf{Q})}{\prod_{n=1}^{N} \lambda_n(\mathbf{Q})}} = \left( \lambda_{\min}\left(\mathbf{\Lambda}^{-1/2}\mathbf{Q}\mathbf{\Lambda}^{-1/2}\right) \right)^{-N/2},$$

since the determinant of $\mathbf{Q}$ is equal to the product of its eigenvalues. Combining this with Lemma 4 yields a bound with the desired dependence.

**Theorem 7.** *Assume that $\mathbf{Q}$ has a diagonalization $\mathbf{Q} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$ such that $\mathbf{\Delta} = \mathbf{V} - \mathbf{I}$ is small in the sense that $\kappa(\mathbf{Q})\rho(\mathbf{\Delta}) < 1$. Then the ratio between the volume of the ellipsoid $\mathcal{E}_{\mathbf{Q}}$ and that of the minimum-volume coordinate-aligned enclosing ellipsoid $\mathcal{E}_{\mathbf{D}}$ is bounded as follows:*

$$\frac{\operatorname{vol}(\mathcal{E}_{\mathbf{D}})}{\operatorname{vol}(\mathcal{E}_{\mathbf{Q}})} \leq (1 - \kappa(\mathbf{Q})\rho(\mathbf{\Delta}))^{-N/2}.$$

In terms of the equivalent dilation factor, we have

$$\chi \leq (1 - \kappa(\mathbf{Q})\rho(\mathbf{\Delta}))^{-1/2}.$$

If the ellipsoid $\mathcal{E}_{\mathbf{Q}}$ is nearly coordinate-aligned and if the condition number $\kappa(\mathbf{Q})$ is small, then the volume ratio and dilation factor are guaranteed to be low. As with Theorem 4, the bound can be strengthened by choosing a diagonal matrix $\mathbf{S}$ that minimizes the quantity corresponding to $\kappa(\mathbf{Q})\rho(\mathbf{\Delta})$.

## 6.5 Determination of minimum-volume coordinate-aligned enclosing ellipsoids

This section discusses the solution of the determinant maximization problem in (6.4.1). The optimal solution to (6.4.1) determines the minimum-volume coordinate-aligned enclosing ellipsoid for $\mathcal{E}_{\mathbf{Q}}$. We use the primal-dual long-step algorithm of [4] to solve (6.4.1). Our purpose in this section is to summarize the algorithm as it applies to the problem of interest and to provide formulas that are used in the algorithm. The reader is referred to [4] for a general discussion of determinant maximization.

The algorithm of [4] assumes a reformulation of problem (6.4.1) that can be obtained as follows. First, the determinant is replaced by the log-determinant since the latter is a concave function whereas the former is not. The optimal solution is unchanged because the logarithm function is monotonic. Second, the constraint $\mathbf{D} \preceq \mathbf{Q}$ is removed and is enforced indirectly by adding the barrier function $\ln\det(\mathbf{Q}-\mathbf{D})$ to the objective function. The constraint $\mathbf{D} \succeq \mathbf{0}$ can also be removed because the original objective function $\ln\det(\mathbf{D})$ already acts as a barrier. The optimization problem that results from these modifications is

$$\begin{aligned} \min_{\mathbf{D}} \quad & \varphi(\mathbf{D};t) \equiv -t\ln\det(\mathbf{D}) - \ln\det(\mathbf{Q}-\mathbf{D}) = -t\sum_{n=1}^{N}\ln(D_{nn}) - \ln\det(\mathbf{Q}-\mathbf{D}) \\ \text{s.t.} \quad & \mathbf{D} \text{ diagonal}, \end{aligned} \tag{6.5.1}$$

where we have also switched from maximization to minimization. The parameter $t$ controls the strength of the original objective function relative to the barrier term. We denote by $\mathbf{D}^*(t)$ the optimal solution to (6.5.1) for a given value of $t$. As $t$ increases to $\infty$, the barrier term decreases in importance while still enforcing the constraint $\mathbf{D} \preceq \mathbf{Q}$, and consequently $\mathbf{D}^*(t)$ converges to the optimal solution of (6.4.1).

The algorithm of [4] solves (6.4.1) by following the locus of optimal solutions to (6.5.1) as illustrated in Fig. 6-6, starting from $t = 1$ and ending when $t$ exceeds a pre-specified threshold $t_{\max}$. The value of $t_{\max}$ can be chosen based on the fact that $\ln\det(\mathbf{D}^*(t))$ for $t \geq$

$t_{\max}$ is guaranteed to differ by no more than $N/t_{max}$ from the true maximum log-determinant in (6.4.1). We typically choose $t_{max} = 1000$ so that the final log-determinant value is within $10^{-3}N$ of the optimal value, where the factor of $N$ accounts for the approximately linear scaling of the log-determinant with $N$.

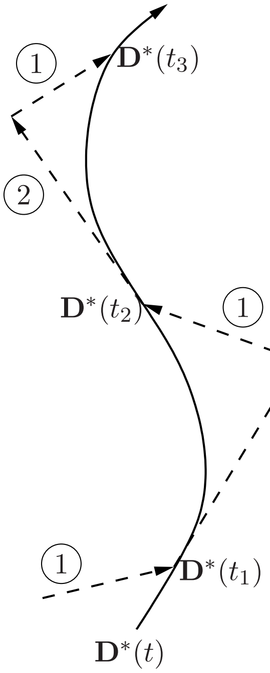


Figure 6-6: Locus of optimal solutions $\mathbf{D}^*(t)$ of (6.5.1). The dashed arrows show the movements associated with the two phases of the algorithm.

The algorithm has two phases: determining $\mathbf{D}^*(t)$ for a fixed value of $t$, and increasing $t$ while simultaneously predicting $\mathbf{D}^*(t)$ for the new value of $t$. These phases are described further in Sections 6.5.1 and 6.5.2.

### 6.5.1 First phase of algorithm

The minimization of $\varphi(\mathbf{D}; t)$ with respect to $\mathbf{D}$ is done using Newton's method, a standard algorithm that uses both first and second derivatives to enable faster convergence [98]. The specific variant proposed in [4] is summarized in Algorithm 7.

In Algorithm 7, the gradient $\nabla\varphi(\mathbf{D}; t)$ is an $N$-dimensional vector consisting of the partial derivatives with respect to the diagonal entries $D_{nn}$. The Hessian $\nabla^2\varphi(\mathbf{D}; t)$ is an $N \times N$ matrix composed of all second derivatives. The gradient and Hessian are computed as

**Algorithm 7** Newton's method for solving (6.5.1) with $t$ fixed.

```
Input: Parameter $\mathbf{Q}$, fixed value of $t$, initial solution $\mathbf{D}$.
Output: Optimal solution to (6.5.1) for fixed $t$.
  repeat
    Compute Newton direction $\Delta\mathbf{D} = -\left(\nabla^2\varphi(\mathbf{D};t)\right)^{-1}\nabla\varphi(\mathbf{D};t)$.
    if $\left|\Delta\mathbf{D}^T\nabla^2\varphi(\mathbf{D};t)\Delta\mathbf{D}\right|^{1/2} > 1/2$ then
      Compute step size $\widehat{p} = \arg\min_p \varphi(\mathbf{D}+p\Delta\mathbf{D};t)$.
    else
      $\widehat{p} = 1$.
    Update $\mathbf{D} \leftarrow \mathbf{D} + \widehat{p}\Delta\mathbf{D}$.
  until $\left|\Delta\mathbf{D}^T\nabla^2\varphi(\mathbf{D};t)\Delta\mathbf{D}\right|^{1/2} < \delta$
```

follows:

$$\nabla\varphi(\mathbf{D};t) = -t\,\mathrm{diag}\left(\mathbf{D}^{-1}\right) + \mathrm{diag}\left((\mathbf{Q}-\mathbf{D})^{-1}\right),$$
$$\nabla^2\varphi(\mathbf{D};t) = t\mathbf{D}^{-2} + (\mathbf{Q}-\mathbf{D})^{-1} \circ (\mathbf{Q}-\mathbf{D})^{-1},$$

where $\circ$ denotes the entry-wise product between matrices. We use $\delta = 10^{-3}$ as the stopping criterion as suggested in [4]. For this value of $\delta$, the output of Algorithm 7 is a very good approximation to the optimal solution $\mathbf{D}^*(t)$, and we will refer to the output as being optimal for simplicity.

The first time that Algorithm 7 is executed, i.e., for $t = 1$, the initial solution $\mathbf{D}$ is set equal to $0.99\lambda_{\min}\left(\mathrm{Diag}(\mathbf{Q})^{-1/2}\mathbf{Q}\,\mathrm{Diag}(\mathbf{Q})^{-1/2}\right)\mathrm{Diag}(\mathbf{Q})$, where $\mathrm{Diag}(\mathbf{Q})$ is as defined in Section 6.4.5 and the somewhat arbitrary factor of 0.99 ensures that $\mathbf{D} \prec \mathbf{Q}$ strictly. For subsequent executions, the initial solution is provided by the second phase of the algorithm to be discussed in Section 6.5.1.

The minimization of $\varphi(\mathbf{D}+p\Delta\mathbf{D};t)$ along the line $\mathbf{D}+p\Delta\mathbf{D}$ parallel to the Newton direction is known as a line search. It can be solved using a one-dimensional version of Newton's method. The change in the objective function, $\Delta\varphi(p) = \varphi(\mathbf{D}+p\Delta\mathbf{D};t) - \varphi(\mathbf{D};t)$, can be expressed as

$$\Delta\varphi(p) = -t\sum_{n=1}^{N}\ln\left(1 + p\frac{\Delta D_{nn}}{D_{nn}}\right) - \sum_{n=1}^{N}\ln\left(1 - p\lambda_n\left((\mathbf{Q}-\mathbf{D})^{-1}\Delta\mathbf{D}\right)\right),$$

where $\lambda_n\left((\mathbf{Q}-\mathbf{D})^{-1}\Delta\mathbf{D}\right)$ refers to an eigenvalue of the matrix $(\mathbf{Q}-\mathbf{D})^{-1}\Delta\mathbf{D}$. We perform the recursion

$$p \leftarrow p - \frac{\Delta\varphi'(p)}{\Delta\varphi''(p)}, \tag{6.5.2}$$

starting from $p = 0$ and stopping when the quantity $\big(\Delta\varphi'(p)\big)^2/\Delta\varphi''(p)$ becomes small. The first and second derivatives of $\Delta\varphi(p)$ are given by

$$\begin{aligned}\Delta\varphi'(p) &= -t\sum_{n=1}^{N}\frac{\Delta D_{nn}/D_{nn}}{1+p\Delta D_{nn}/D_{nn}} + \sum_{n=1}^{N}\frac{\lambda_n\big((\mathbf{Q}-\mathbf{D})^{-1}\Delta\mathbf{D}\big)}{1-p\lambda_n\big((\mathbf{Q}-\mathbf{D})^{-1}\Delta\mathbf{D}\big)},\\ \Delta\varphi''(p) &= t\sum_{n=1}^{N}\left(\frac{\Delta D_{nn}/D_{nn}}{1+p\Delta D_{nn}/D_{nn}}\right)^2 + \sum_{n=1}^{N}\left(\frac{\lambda_n\big((\mathbf{Q}-\mathbf{D})^{-1}\Delta\mathbf{D}\big)}{1-p\lambda_n\big((\mathbf{Q}-\mathbf{D})^{-1}\Delta\mathbf{D}\big)}\right)^2.\end{aligned}$$

It may be necessary to occasionally modify the recursion in (6.5.2) to ensure that the arguments of the ln functions are strictly positive. More specifically, $p$ must satisfy the bounds

$$\begin{aligned}p &< \min\left\{-\frac{D_{nn}}{\Delta D_{nn}} : \Delta D_{nn} < 0,\ \frac{1}{\lambda_n\big((\mathbf{Q}-\mathbf{D})^{-1}\Delta\mathbf{D}\big)} : \lambda_n\big((\mathbf{Q}-\mathbf{D})^{-1}\Delta\mathbf{D}\big) > 0\right\},\\ p &> \max\left\{-\frac{D_{nn}}{\Delta D_{nn}} : \Delta D_{nn} > 0,\ \frac{1}{\lambda_n\big((\mathbf{Q}-\mathbf{D})^{-1}\Delta\mathbf{D}\big)} : \lambda_n\big((\mathbf{Q}-\mathbf{D})^{-1}\Delta\mathbf{D}\big) < 0\right\},\end{aligned}$$

and the size of the update to $p$ in (6.5.2) should be reduced as $p$ approaches these bounds.

### 6.5.2 Second phase of algorithm

In the second phase of the algorithm, two steps are performed in an alternating fashion. First, the value of $t$ is increased from a starting value of $t_0$. As $t$ increases, the solution $\mathbf{D} = \mathbf{D}^*(t_0)$ from Newton's method, which is optimal for $t = t_0$, deviates more and more from $\mathbf{D}^*(t)$ for the new value of $t$. As is described more fully below, the size of the increase in $t$ is determined by placing a limit on the deviation of $\mathbf{D}$ from optimality. After $t$ is fixed to a new value, the second step is to update $\mathbf{D}$ so as to reduce the optimality gap. The update is restricted to be of the form

$$\mathbf{D} = \mathbf{D}^*(t_0) + p\frac{\partial\mathbf{D}^*}{\partial t}(t_0), \quad p \geq 0, \tag{6.5.3}$$

where

$$\frac{\partial\mathbf{D}^*}{\partial t}(t_0) = \left(\nabla^2\varphi\left(\mathbf{D}^*(t_0); t_0\right)\right)^{-1}\operatorname{diag}\Big(\left(\mathbf{D}^*(t_0)\right)^{-1}\Big)$$

is the tangent at $t = t_0$ to the locus of optimal solutions of (6.5.1), as depicted in Fig. 6-6. The update to $\mathbf{D}$ allows $t$ to be increased further, after which $\mathbf{D}$ can be adjusted again. By repeating these two steps several times, $t$ can often be increased substantially, thus enhancing the overall convergence, while $\mathbf{D}$ is simultaneously kept close to the optimal solution of (6.5.1), leading to faster convergence of Newton's method (Algorithm 7).

To describe the two steps in more detail, we introduce an $N \times N$ matrix $\mathbf{Z}$ defined by

$$\mathbf{Z} = \mathbf{Z}^*(t_0) + q\frac{\partial \mathbf{Z}^*}{\partial t}(t_0), \quad q \geq 0, \tag{6.5.4}$$

where

$$\begin{aligned} \mathbf{Z}^*(t_0) &= \frac{1}{t_0}\left(\mathbf{Q} - \mathbf{D}^*(t_0)\right)^{-1}, \\ \frac{\partial \mathbf{Z}^*}{\partial t}(t_0) &= -\frac{1}{t_0}\mathbf{Z}^*(t_0) + t_0\mathbf{Z}^*(t_0)\,\mathrm{Diag}\left(\frac{\partial D^*_{11}}{\partial t}(t_0), \ldots, \frac{\partial D^*_{NN}}{\partial t}(t_0)\right)\mathbf{Z}^*(t_0). \end{aligned}$$

It is shown in [4] that $\mathbf{Z}^*(t_0)$ is the optimal solution to the dual of problem (6.5.1) for $t = t_0$ and that $(\partial\mathbf{Z}^*/\partial t)(t_0)$ is the tangent to the locus of dual optimal solutions.

In the first step, $\mathbf{D}$ and $\mathbf{Z}$ are kept constant while $t$ is increased until the deviation of $\varphi(\mathbf{D}; t)$ from the minimum value in (6.5.1) reaches a threshold. To estimate the deviation from optimality, [4] uses the following function:

$$\psi(\mathbf{D}, \mathbf{Z}, t) = -t\sum_{n=1}^{N}\ln(D_{nn}) - \ln\det(\mathbf{Q} - \mathbf{D}) + t\mathbf{Q}\bullet\mathbf{Z} - t\sum_{n=1}^{N}\ln(Z_{nn}) - \ln\det(\mathbf{Z}) - N(1 + \ln t + t), \tag{6.5.5}$$

which is an upper bound on the optimality gap for arbitrary $\mathbf{Z}$. The function $\psi(\mathbf{D}, \mathbf{Z}, t)$ is equal to zero at the beginning of the second phase when $t = t_0$, $\mathbf{D} = \mathbf{D}^*(t_0)$, and $\mathbf{Z} = \mathbf{Z}^*(t_0)$. With $\mathbf{D}$ and $\mathbf{Z}$ fixed, $t$ is increased until $\psi(\mathbf{D}, \mathbf{Z}, t)$ attains a pre-specified value $\psi_{\max}$. We have found that setting $\psi_{\max} \simeq 80$ results in a good trade-off between increasing $t$ as much as possible and allowing Algorithm 7 to converge quickly. The new value of $t$ is obtained by solving $\psi(\mathbf{D}, \mathbf{Z}, t) = \psi_{\max}$ for $t$. This is a nonlinear equation of the form

$$at - N\ln t - b = 0, \tag{6.5.6}$$

with coefficients $a$ and $b$ that can be read from (6.5.5). We may again use Newton's method, this time the root-finding version, to solve (6.5.6). An initial solution of $t = \frac{b}{a} + \frac{N}{a} \ln \frac{N}{a}$ is often a good starting point.

In the second step, $t$ is kept constant while $\mathbf{D}$ and $\mathbf{Z}$ are updated, specifically by choosing values for $p$ and $q$ in (6.5.3) and (6.5.4) that minimize $\psi(\mathbf{D}, \mathbf{Z}, t)$. It is straightforward to show from (6.5.3)–(6.5.5) that the change in $\psi(\mathbf{D}, \mathbf{Z}, t)$ as a function of $p$ and $q$ is given by

$$\begin{aligned}
\Delta\psi(p, q) &\equiv \psi(\mathbf{D} + p\Delta\mathbf{D}, \mathbf{Z} + q\Delta\mathbf{Z}, t) - \psi(\mathbf{D}, \mathbf{Z}, t) \\
&= -t \sum_{n=1}^{N} \ln \left(1 + p\frac{\Delta D_{nn}}{D_{nn}}\right) - \sum_{n=1}^{N} \ln \left(1 - p\lambda_n\left((\mathbf{Q} - \mathbf{D})^{-1}\Delta\mathbf{D}\right)\right) \\
&\qquad + qt\mathbf{Q} \bullet \mathbf{Z} - t \sum_{n=1}^{N} \ln \left(1 + q\frac{\Delta Z_{nn}}{Z_{nn}}\right) - \sum_{n=1}^{N} \ln \left(1 + q\lambda_n\left(\mathbf{Z}^{-1}\Delta\mathbf{Z}\right)\right),
\end{aligned}$$

where we have set $\mathbf{D} = \mathbf{D}^*(t_0)$, $\Delta\mathbf{D} = (\partial\mathbf{D}^*/\partial t)(t_0)$, $\mathbf{Z} = \mathbf{Z}^*(t_0)$, and $\Delta\mathbf{Z} = (\partial\mathbf{Z}^*/\partial t)(t_0)$ to simplify notation. The minimization of $\Delta\psi(p, q)$ with respect to $p$ and $q$ decouples into two one-dimensional minimizations. The first minimization over $p$ is identical to the line search discussed in Section 6.5.1 while the second minimization over $q$ is similar. After this step is complete, $\psi(\mathbf{D}, \mathbf{Z}, t) < \psi_{\max}$ once again and the first step may be repeated.

In Algorithm 8, we summarize the steps of the second phase. In our experience, most of the potential increase in $t$ is realized after $I = 3$ repetitions. The final values for $t$ and $\mathbf{D}$ become the inputs to the first phase (Algorithm 7).

**Algorithm 8** Second phase of the determinant maximization algorithm of [4].

```
Input: Parameter $\mathbf{Q}$, initial value $t = t_0$, corresponding optimal solution $\mathbf{D}^*(t_0)$ to (6.5.1).
Output: Final value $t > t_0$, solution $\mathbf{D}$ such that the difference between $\varphi(\mathbf{D}; t)$ and the
  minimum value in (6.5.1) is no greater than $\psi_{\max}$.
  Initialize $p = q = 0$.
  Compute $\mathbf{D}$ and $\mathbf{Z}$ using (6.5.3) and (6.5.4).
  for $i = 1, \ldots, I$ do
    Solve (6.5.5) for $t$ with $\mathbf{D}$, $\mathbf{Z}$ fixed.
    Line searches: $\widehat{p}, \widehat{q} = \arg\min_{p,q} \Delta\psi(p, q)$.
    Update $\mathbf{D}$ and $\mathbf{Z}$ by substituting $p = \widehat{p}$ in (6.5.3) and $q = \widehat{q}$ in (6.5.4).
```

## 6.6 Numerical comparison of lower bounds

In Sections 6.2–6.4, we developed three classes of lower bounds on the optimal values of problems (5.1.1) and (5.1.2). The first of these bounds uses knowledge of the candidate ranges of the coefficients and is relatively simple to compute, while the other two improve upon the first bound through the solution of linear and diagonal relaxations. To evaluate the strength of the bounds and compare them to each other, numerical experiments similar to those in Section 3.6 were performed in which instances of (5.1.1) and (5.1.2) are created by randomly choosing values for $\mathbf{Q}$ and $\mathbf{c}$. Unlike in Section 3.6, it will be seen that the linear relaxation is competitive with and can sometimes outperform the diagonal relaxation, especially for the NLZ cost function.

As before, several problem parameters are varied to characterize the quality of the bounds under different conditions. The effects of the condition number and eigenvalue distribution of $\mathbf{Q}$ will be familiar from Section 3.6 and the analysis in Section 6.4.4. The experiments also reveal a dependence on the wordlength $P$. In addition, it will be shown that the bounds based on candidate ranges have a significant effect on the bounds due to linear and diagonal relaxations, which are built upon the former.

In the first set of experiments in this section, the matrix $\mathbf{Q}$ is created according to the first three methods in Section 3.6. The eigenvalues of $\mathbf{Q}$ are sampled from one of three distributions with the condition number $\kappa(\mathbf{Q})$ equal to $\sqrt{N}$, $N$, $10N$, or $100N$, while the eigenvectors are drawn uniformly from the unit sphere as an orthonormal set. The number of dimensions $N$ is varied between 10 and 100.

The ellipsoid center $\mathbf{c}$ is chosen differently from before because it can no longer be assumed that the value zero is included in the candidate range for each coefficient. Once $\mathbf{Q}^{-1}$ is determined, $c_n$ is drawn with equal probability either from a uniform distribution over

$$\left[-\sqrt{\left(\mathbf{Q}^{-1}\right)_{nn}}, \sqrt{\left(\mathbf{Q}^{-1}\right)_{nn}}\right]$$

as in Section 3.6, or from a power-law distribution proportional to $1/\left|c_n\right|$ over the intervals

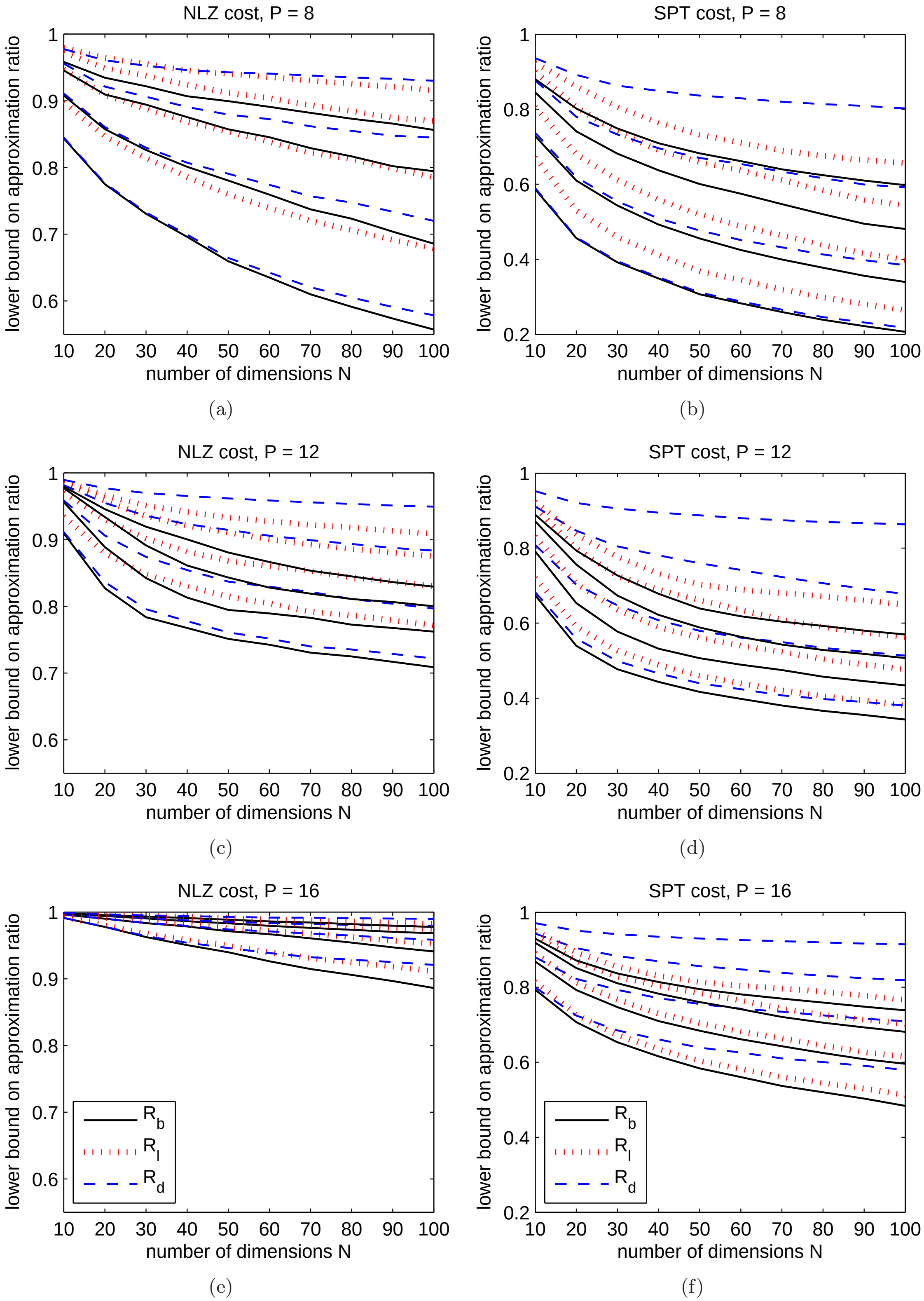
$$\left[-1000\sqrt{\left(\mathbf{Q}^{-1}\right)_{nn}}, -\sqrt{\left(\mathbf{Q}^{-1}\right)_{nn}}\right] \qquad \text{and} \qquad \left[\sqrt{\left(\mathbf{Q}^{-1}\right)_{nn}}, 1000\sqrt{\left(\mathbf{Q}^{-1}\right)_{nn}}\right].$$

With $\gamma$ initially equal to 1, the candidate range defined by (6.1.1) includes zero in the first case but not in the second. We subsequently scale $\mathbf{c}$ and $\gamma$ by powers of two as discussed in Section 5.1.1 to allow the filter coefficients to be integer-valued. Three different wordlengths $P = 8, 12, 16$ are used in the experiments.

It is often necessary to further modify the value of $\gamma$ to ensure that the quadratic constraint (2.1.1) admits at least one integer-valued solution. To do this, we use Algorithm 4 to approximately minimize the left-hand side of (2.1.1) over integer-valued $\mathbf{b}$ with $\mathbf{Q}$ and $\mathbf{c}$ determined as above. We then set $\gamma$ equal to 10/9 times the error value returned by Algorithm 4. This guarantees that the problem instance has at least one feasible solution with quantization error equal to $0.9\gamma$. Note that the new value of $\gamma$ also affects the candidate ranges through (6.1.1).

For each combination of $N$, $P$, $\kappa(\mathbf{Q})$, and eigenvalue distribution, 1000 instances are created. For each instance, we compute the lower bounds in (6.2.1) and (6.2.2) as well as the bounds resulting from linear and diagonal relaxations for both cost measures. Linear relaxations are solved via the dual formulations in (6.3.7) and (6.3.18) using the MATLAB solver `fmincon`. Diagonal relaxations are solved by determining the diagonal matrix $\mathbf{D}$ that minimizes the volume of $\mathcal{E}_{\mathbf{D}}$ as discussed in Section 6.5 and then applying either Algorithm 5 or 6. We also obtain a feasible solution using the heuristic algorithm of Section 5.3. As in Section 3.6, the ratio between each of the lower bounds and the cost of the feasible solution is used to measure the quality of approximation, with $R_\ell$ and $R_d$ denoting the ratios corresponding to linear and diagonal relaxations. The ratios corresponding to the bounds in (6.2.1) and (6.2.2) are denoted by $R_b$. As noted before, the ratios $R_b$, $R_\ell$, and $R_d$ are lower bounds on the true approximation ratios.

In Fig. 6-7, we plot the ratios $R_b$, $R_\ell$, and $R_d$, averaged over 1000 instances, for the first eigenvalue distribution $f_1(\lambda) \propto 1/\lambda$. The plots reveal a number of different effects; we focus first on the role of the condition number $\kappa(\mathbf{Q})$. The ratio $R_b$ increases as $\kappa(\mathbf{Q})$ decreases, an expected result given the box interpretation discussed in Section 6.2. The set of integer points within a nearly spherical ellipsoid can be contained in a smaller coordinate-aligned box on average compared to an oblong ellipsoid. Since the bounds arising from the box approximation are incorporated as a baseline in linear and diagonal relaxations, the ratios $R_\ell$ and $R_d$ inherit the same dependence on $\kappa(\mathbf{Q})$, a feature seen throughout this section. The variation with condition number is even more pronounced in the case of diagonal relaxations as can be predicted from Theorem 5 and related discussion in Section 6.4.4. Comparing $R_\ell$ and $R_d$, it is seen that linear relaxations sometimes outperform diagonal relaxations at higher values of $\kappa(\mathbf{Q})$, especially for the NLZ cost function. Diagonal relaxations tend to be stronger than linear relaxations for the SPT cost function. All three ratios however are significantly lower for the SPT cost than for the NLZ cost.



(a)

(b)

(c)

(d)

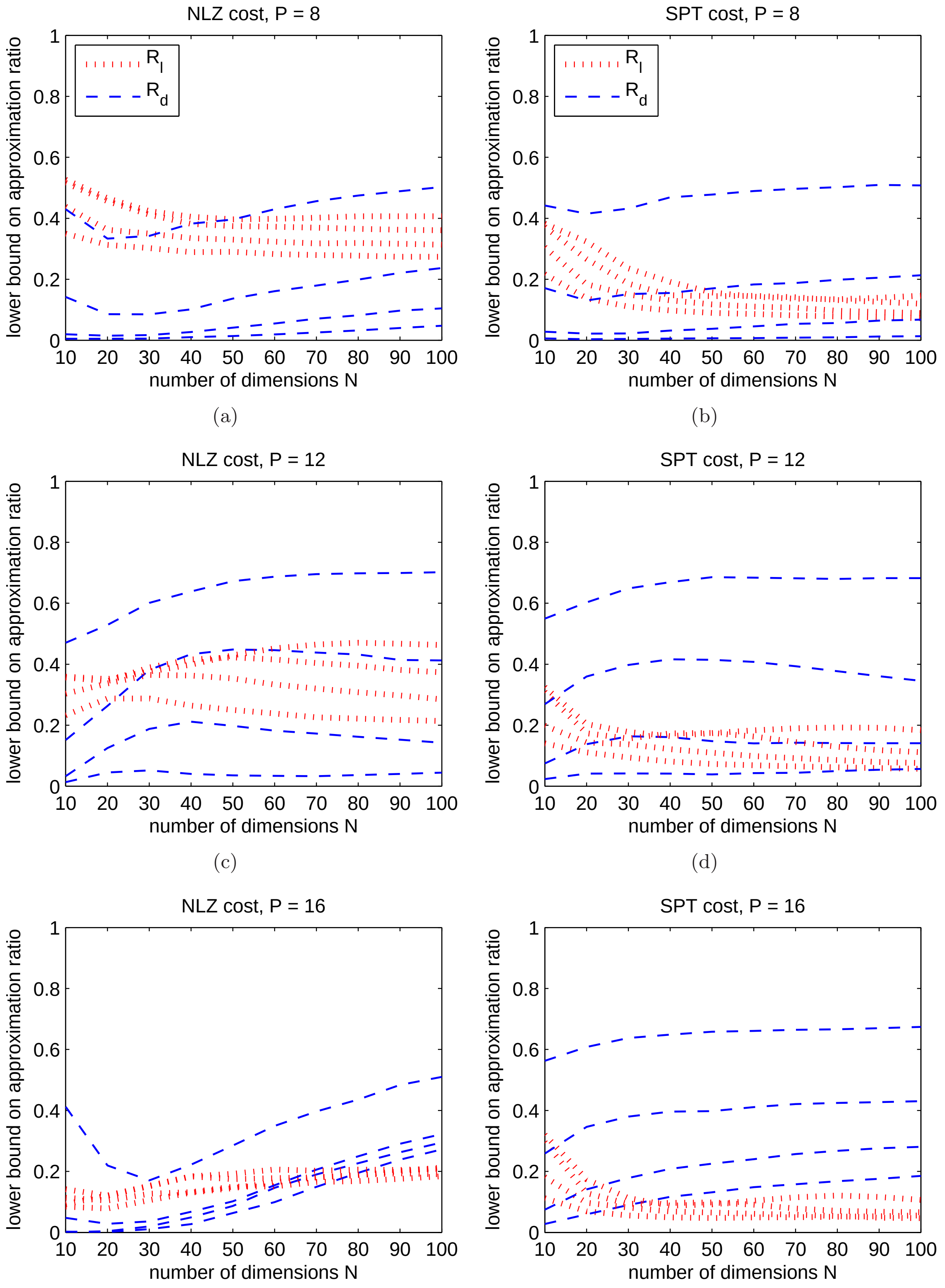
(e)

(f)

Figure 6-7: Average values of $R_b$, $R_\ell$, and $R_d$ for a $1/\lambda$ eigenvalue distribution. Within each set of curves, $\kappa(\mathbf{Q}) = \sqrt{N}, N, 10N, 100N$ from top to bottom.

Next we discuss the dependence on the number of dimensions $N$. The downward trend of the curves in Fig. 6-7 can be attributed to the deteriorating quality of the box approximation as $N$ increases. This in turn can be explained by the super-exponential growth of the box-to-ellipsoid volume ratio in (6.2.4). To show that the dependence of $R_\ell$ and $R_d$ on $N$ is largely due to that of $R_b$, we compute modified ratios $\widetilde{R}_\ell$ and $\widetilde{R}_d$ by subtracting the bound in either (6.2.1) or (6.2.2) from both the numerator and denominator in $R_\ell$ and $R_d$. Hence the numerators in $\widetilde{R}_\ell$ and $\widetilde{R}_d$ represent the improvement over (6.2.1) or (6.2.2) due to linear and diagonal relaxations, while the denominators represent the cost of a feasible solution in excess of (6.2.1) or (6.2.2). Fig. 6-8 plots the curves for $\widetilde{R}_\ell$ and $\widetilde{R}_d$ corresponding to Fig. 6-7. The dependence on $N$ is now much weaker with $\widetilde{R}_d$ even tending to increase slightly as $N$ increases. It is also apparent that the improvement in lower bounds due to relaxations is modest in most cases as measured by these ratios.

It can also be observed that the ratios in Fig. 6-7 improve as the wordlength $P$ increases. The dependence on $P$ can be explained by the associated change in the size of the ellipsoid $\mathcal{E}_{\mathbf{Q}}$ relative to the size of $\mathbf{c}$. Specifically, a smaller wordlength implies that $\mathcal{E}_{\mathbf{Q}}$ must be larger to ensure that a quantized feasible solution exists, ignoring for the moment the normalization to an integer quantization grid. Fig. 6-9 illustrates using a simplified one-dimensional example how an increase in the size of the feasible set can lead to a worse approximation ratio. We consider the cost function $C_{\text{NLZ}}(b_n)$ and approximate it by the smooth function $\log_2(1 + |b_n|)$. In panel (a), the original interval is small relative to the magnitude of the midpoint $c_n$ and an enclosing interval yields a good approximation to the true minimum cost. In panel (b), both the original and enclosing intervals are expanded about $c_n$ by a factor of 3, resulting in a significantly worse approximation even though the length ratio is preserved. The degradation is due to the more rapid decrease of the logarithm function near zero. The same intuition extends to higher dimensions and to different enclosing shapes, namely boxes and ellipsoids. This accounts for the significant improvement in the approximation ratios going down the left-hand column of Fig. 6-7. For the cost function $C_{\text{SPT}}(\mathbf{b})$, the intuition is complicated by the non-monotonicity of the cost function. Nevertheless it is still true that lower values of $C_{\text{SPT}}(b_n)$ occur more frequently when $b_n$ is small, and it is the minimum value within a given region that contributes to the approximation ratio.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 6-8: Average values of $\widetilde{R}_\ell$ and $\widetilde{R}_d$ for a $1/\lambda$ eigenvalue distribution. Within each set of curves, $\kappa(\mathbf{Q}) = \sqrt{N}, N, 10N, 100N$ from top to bottom.

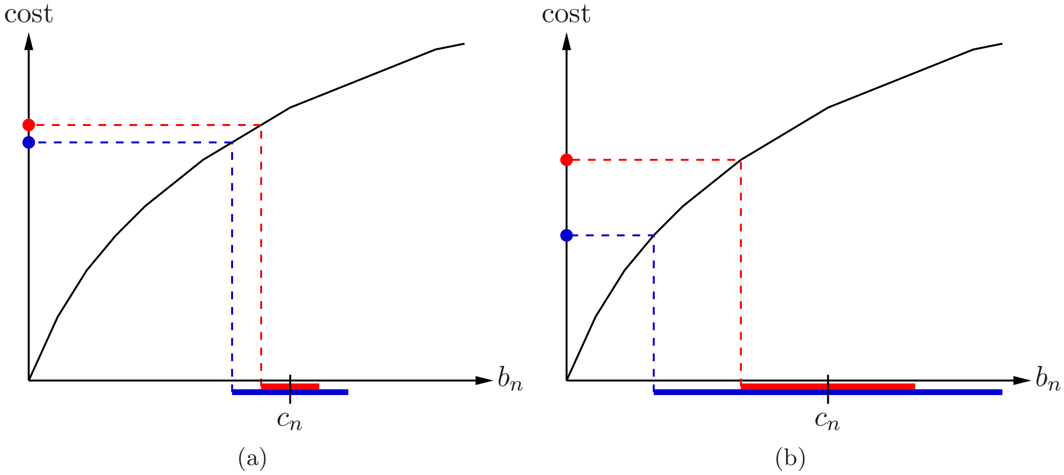


Figure 6-9: Effect of the size of the feasible set on the approximation ratio. The original (red) and enclosing (blue) intervals in (b) are 3 times larger than those in (a), leading to a worse approximation of the minimum cost.

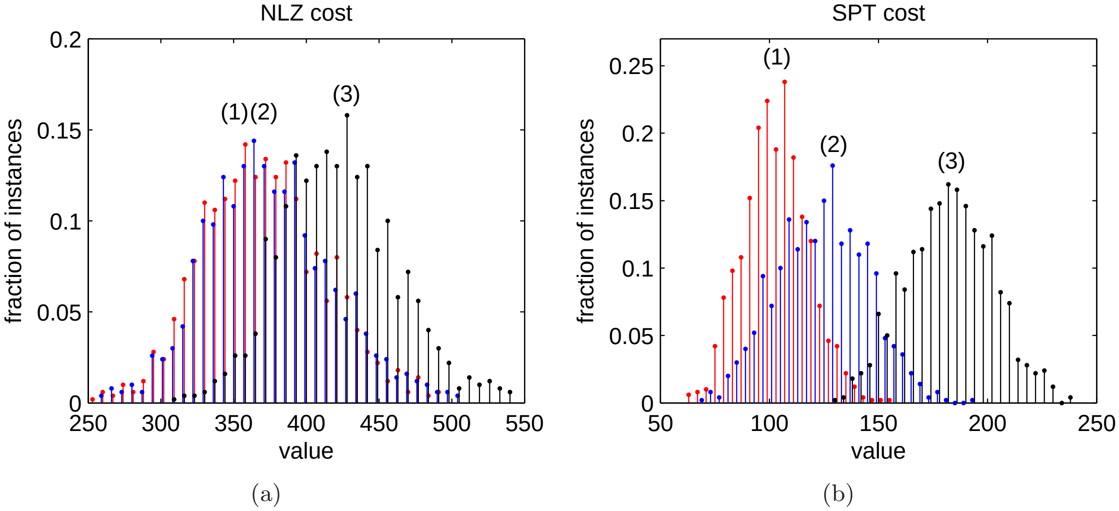
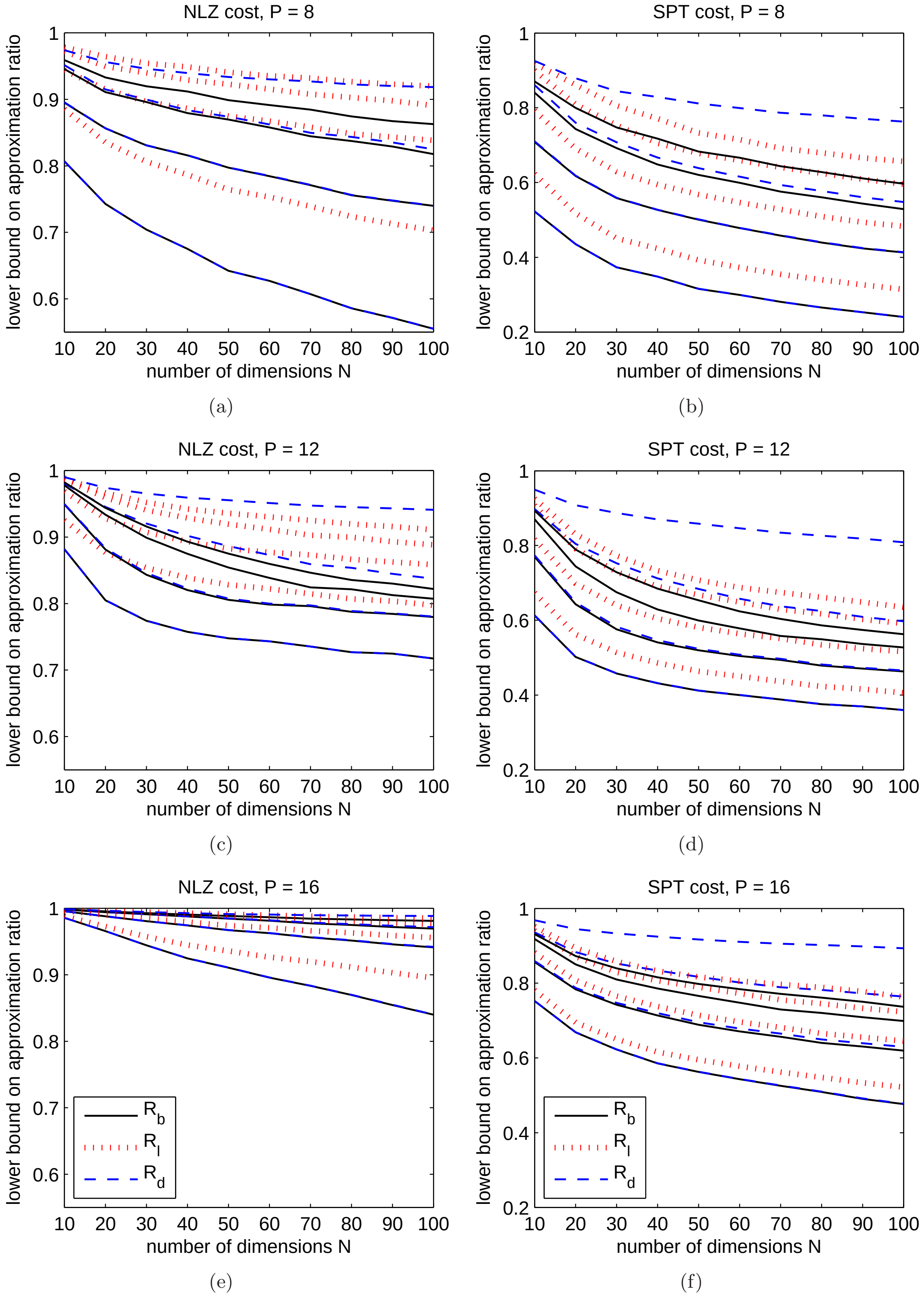


Figure 6-10: Histograms of the optimal values of linear relaxations (1), the optimal values of diagonal relaxations (2), and the objective values of feasible solutions (3) for a $1/\lambda$ eigenvalue distribution, $N = \kappa(\mathbf{Q}) = 100$, and $P = 12$.

To visualize the spread around the mean values plotted in Fig. 6-7, in Fig. 6-10 we show histograms of the optimal values of linear and diagonal relaxations and the objective values of feasible solutions obtained by the heuristic algorithm for a $1/\lambda$ eigenvalue distribution, $N = \kappa(\mathbf{Q}) = 100$, and $P = 12$. For the NLZ cost function, the histograms for linear and diagonal relaxations overlap almost completely and also overlap substantially with the histogram for feasible solutions. This agrees with the average values of $R_\ell$ and $R_d$ at $N = 100$ in panel (c) of Fig. 6-7. In contrast, the histograms for the SPT cost function are more distinct, in agreement with panel (d) of Fig. 6-7. The spread in the histograms is similar for other values of $N$, $\kappa(\mathbf{Q})$, and $P$.

Fig. 6-11 shows the average values of $R_b$, $R_\ell$, and $R_d$ under a uniform eigenvalue distribution. As predicted by Theorem 5, the approximation quality for diagonal relaxations is significantly worse than for a $1/\lambda$ eigenvalue distribution. At higher values of $\kappa(\mathbf{Q})$, the curves for $R_d$ are nearly indistinguishable from those for $R_b$, indicating a lack of improvement over the box approximation. As in Fig. 6-7, diagonal relaxations tend to be better for the SPT cost function than for the NLZ cost function. The ratio $R_b$ is also affected by the change in eigenvalue distribution, although to a lesser degree. This is to be expected since the ellipsoid $\mathcal{E}_{\mathbf{Q}}$ now tends to have more short axes and is therefore smaller compared to the box $\mathcal{B}_{\mathbf{Q}}$. The dependence on the parameters $N$, $\kappa(\mathbf{Q})$, and $P$ is similar to before.

Fig. 6-12 plots the average values of $R_b$, $R_\ell$, and $R_d$ under the eigenvalue distribution $f_2(\lambda) \propto 1/\lambda^2$. This time all of the ratios increase relative to their values in Fig. 6-7, most strikingly in the case of $R_d$. With some exceptions at small values of $N$, diagonal relaxations are now preferred over linear relaxations for all values of $N$, $\kappa(\mathbf{Q})$ and $P$. Moreover, the curves for $R_d$ no longer decrease as a function of $N$. The improvements in $R_d$ and $R_b$ are again consistent with Theorem 5 and the box interpretation in Section 6.2.



(a)

(b)

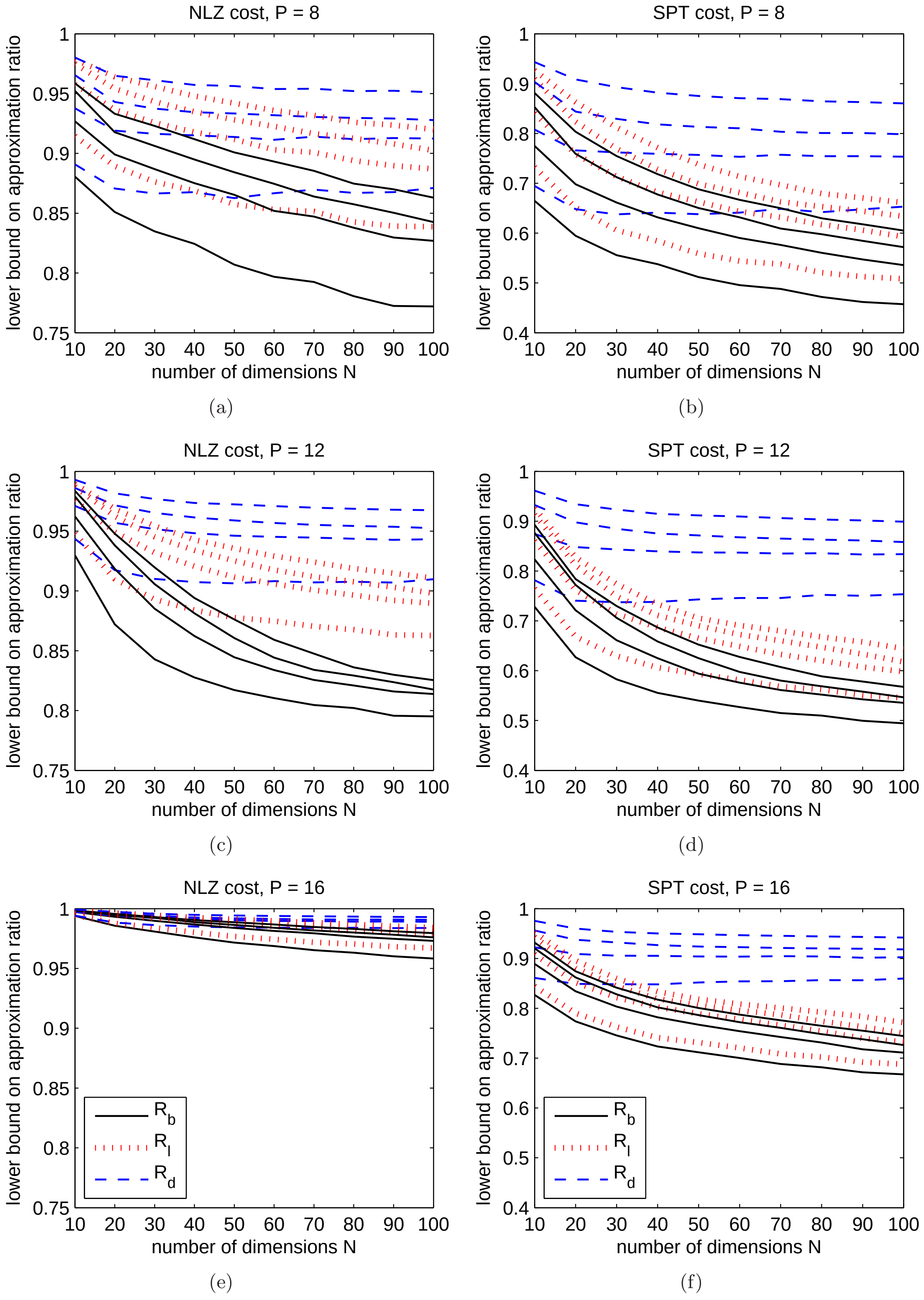
(c)

(d)

(e)

(f)

Figure 6-11: Average values of $R_b$, $R_\ell$, and $R_d$ for a uniform eigenvalue distribution. Within each set of curves, $\kappa(\mathbf{Q}) = \sqrt{N}, N, 10N, 100N$ from top to bottom.



(a)

(b)

(c)

(d)

(e)

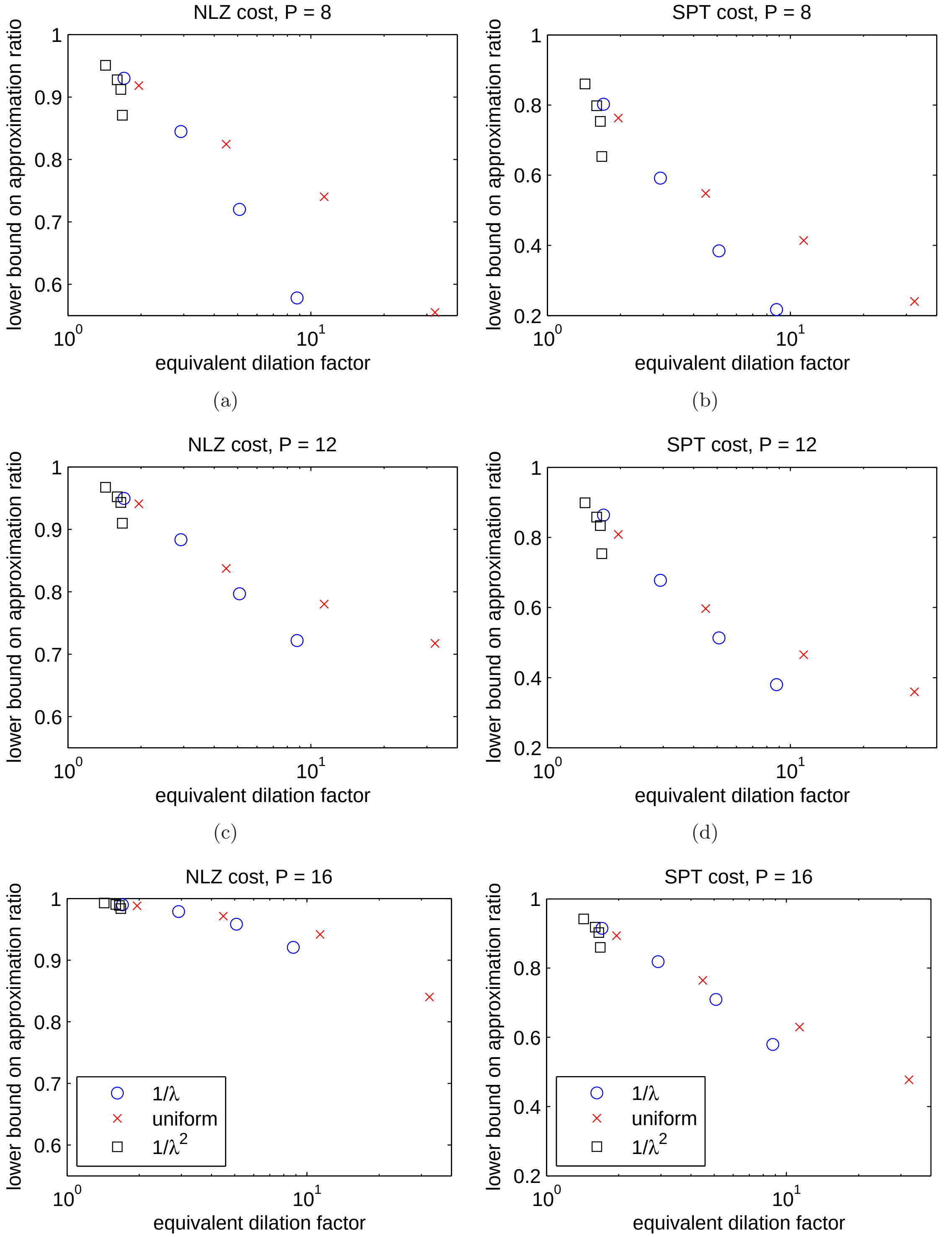
(f)

Figure 6-12: Average values of $R_b$, $R_\ell$, and $R_d$ for a $1/\lambda^2$ eigenvalue distribution. Within each set of curves, $\kappa(\mathbf{Q}) = \sqrt{N}, N, 10N, 100N$ from top to bottom.

In Fig. 6-13, we examine the relationship between the approximation ratio $R_d$ and the dilation factor $\chi$, defined in Section 6.4.4 as the $N$th root of the volume ratio between the ellipsoids $\mathcal{E}_{\mathbf{Q}}$ and $\mathcal{E}_{\mathbf{D}}$. There is a clear negative correlation between the two ratios, thus justifying our focus on the volume ratio in Section 6.4 as a substitute for the true approximation ratio. However, points corresponding to different eigenvalue distributions appear to trace out different curves, suggesting that there are additional factors affecting the ratio $R_d$. We also see another illustration of the phenomenon depicted in Fig. 6-9. The dilation factor does not depend on the wordlength $P$ as seen going down the columns of Fig. 6-13, but the ratio $R_d$ does because of the change in size of the ellipsoid $\mathcal{E}_{\mathbf{Q}}$.

In a fourth experiment, $\mathbf{Q}$ is chosen to represent an exponentially decaying autocorrelation function as was done in Section 3.6. The decay factor $\rho$ in (3.6.1) is varied between 0.1 and 0.99. Recall that it is sufficient to consider only positive $\rho$ and that $\mathbf{Q}$ is diagonally dominant in the sense assumed in Theorem 6 for $\rho \leq 1/3$. Once $\mathbf{Q}$ is determined, the parameters $\mathbf{c}$ and $\gamma$ are generated as in the earlier experiments. Fig. 6-14 shows the ratios $R_b$, $R_\ell$, and $R_d$ for selected values of $\rho$. The parameter $\rho$ is seen to have a similar effect on the approximation ratios as the condition number $\kappa(\mathbf{Q})$, with small values of $\rho$ preferred. As in Fig. 3-12, the bounds provided by diagonal relaxations are surprisingly strong even for $\rho$ close to 1.

To summarize this section, it was shown that linear relaxations can provide better lower bounds for the bit-based cost measures than for the coefficient sparsity measure. For the NLZ cost function in particular, linear and diagonal relaxations perform about equally well. The observed dependence on the condition number, eigenvalue distribution, and diagonal dominance of $\mathbf{Q}$ confirms the analysis in Sections 6.4.4 and 6.4.5 and the associated geometric intuition. The dependence on dimension $N$ is attributed to the box-to-ellipsoid volume ratio in (6.2.4), while the dependence on wordlength $P$ is due to the change in size of the feasible set as illustrated in Fig. 6-9.



(a)

(b)

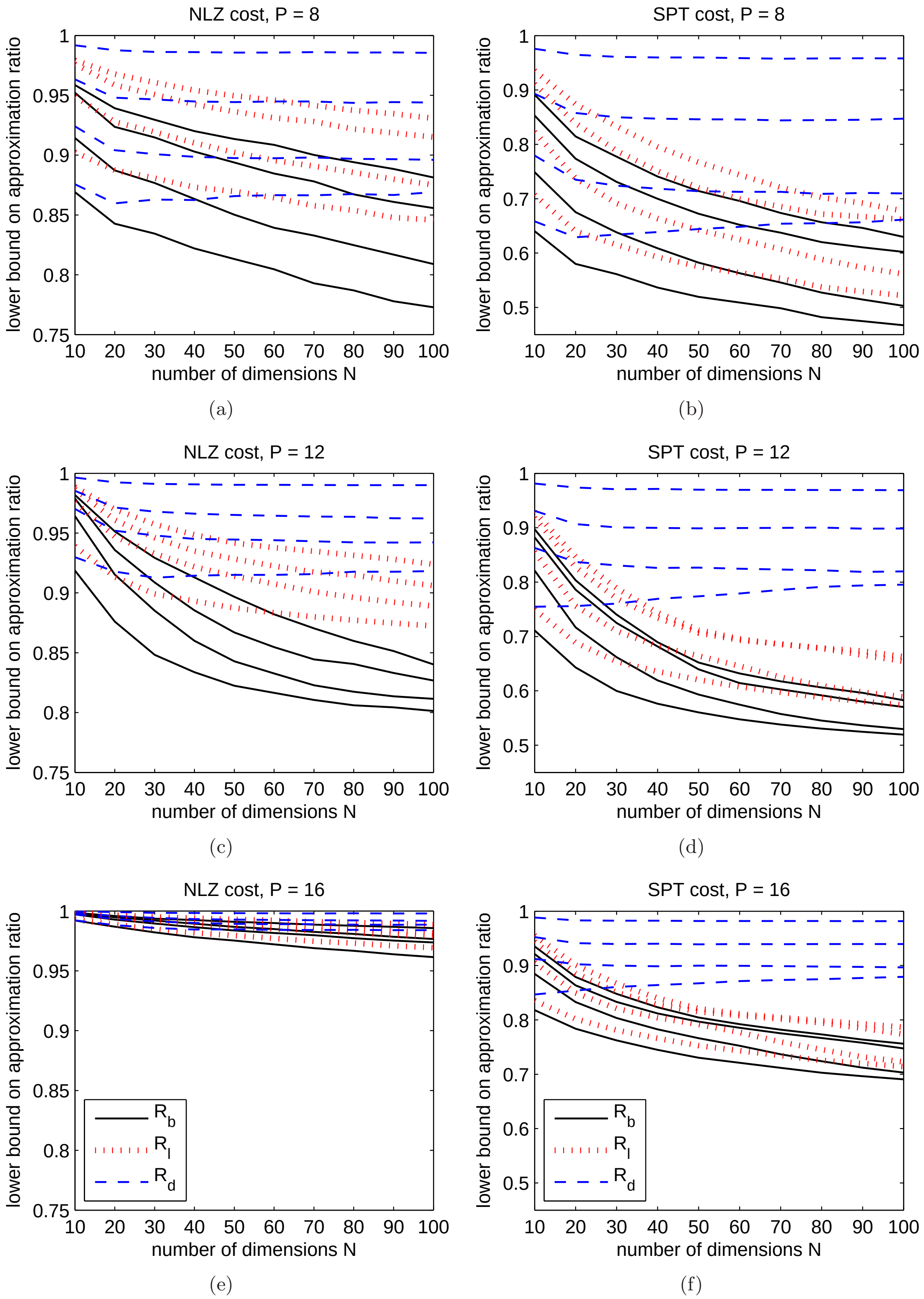
(c)

(d)

(e)

(f)

Figure 6-13: Average values of $R_d$ and the dilation factor $\chi$ for different eigenvalue distributions and $N = 100$. For each type of plotting symbol, $\kappa(\mathbf{Q}) = \sqrt{N}, N, 10N, 100N$ from left to right.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 6-14: Average values of $R_b$, $R_\ell$, and $R_d$ for $\mathbf{Q}$ corresponding to exponentially decaying autocorrelation functions. Within each set of curves, $\rho = 0.1, 0.5, 0.9, 0.99$ from top to bottom.

## 6.7 Description of branch-and-bound algorithm

We now complete our description of a branch-and-bound algorithm for solving problems (5.1.1) and (5.1.2) to optimality. The algorithm is similar to Algorithm 2, the branch-and-bound algorithm for the sparsity maximization problem (2.0.1). Beginning with the root problem, subproblems are removed one at a time from a list and are processed to improve lower bounds and possibly generate feasible solutions. Subproblems are added to the list by branching and can also be removed by pruning. The algorithm terminates when the list is empty. We summarize the algorithm in Algorithm 9 and explain each of the numbered steps in more detail, focusing on differences with respect to the corresponding steps in Algorithm 2. The notation $C(\mathbf{b})$ below can refer either to the number of NLZ bits or the number of SPTs.

1. *Select subproblem from list:* As in Algorithm 2, we choose the subproblem with the smallest lower bound inherited from its parent, for the same reasons as before. Each subproblem is specified by the subsets $\mathcal{F}$ and $\mathcal{K}$ corresponding to free and fixed coefficients respectively, the values of the fixed coefficients $\mathbf{b}_{\mathcal{K}}$, and the inherited lower bound $LB$.

2. *Subproblem parameters:* Every subproblem is equivalent to an $|\mathcal{F}|$-dimensional instance of the root problem, (5.1.1) or (5.1.2), with parameters given by (C.2.2)–(C.2.6).

3. *Identify infeasibility or coefficients that can be fixed:* We first determine the candidate range for each coefficient using (6.1.1). As discussed in Section 6.1, if $\underline{B}_n > \overline{B}_n$ for any $n \in \mathcal{F}$, the subproblem is infeasible and we move on to the next subproblem. If $\underline{B}_n = \overline{B}_n$, the coefficient $b_n$ can be fixed. The subsets $\mathcal{F}$ and $\mathcal{K}$ are updated accordingly and the cost of the fixed coefficients, $C(\mathbf{b}_{\mathcal{K}})$, is increased, which may result in an increase in the lower bound $LB$ as well if the previous value is less than $C(\mathbf{b}_{\mathcal{K}})$. We then prune the subproblem if the pruning condition $LB \geq C(\mathbf{b}_I)$ is satisfied. Since the fixing of coefficients yields a different subproblem, the parameters $\mathbf{Q}_{\text{eff}}$, $\mathbf{c}_{\text{eff}}$, $\gamma_{\text{eff}}$ and the candidate ranges must be recomputed. The current step is then repeated until no more coefficients can be fixed.

4. *Solve zero- or one-dimensional instances:* If no free coefficients remain after Step 3,

**Algorithm 9** Branch-and-bound for problems (5.1.1) and (5.1.2)

```
Input: Parameters $\mathbf{Q}$, $\mathbf{c}$, $\gamma$, wordlength $P$.
Output: Optimal solution $\mathbf{b}_I$ to (5.1.1) or (5.1.2).
  Initialize: Place root problem in list with $LB = 0$. Incumbent solution $\mathbf{b}_I = \mathbf{c}$ with cost
  $C(\mathbf{b}_I) = \infty$.
  while list not empty do
    1) Select subproblem with smallest $LB$ and remove from list.
    2) Compute subproblem parameters $\mathbf{Q}_{\text{eff}}$, $\mathbf{c}_{\text{eff}}$, $\gamma_{\text{eff}}$ using (C.2.2)–(C.2.6).
    3) Determine candidate ranges $\underline{B}_n, \dots, \overline{B}_n$ from (6.1.1) for $n \in \mathcal{F}$.
    if $\underline{B}_n > \overline{B}_n$ for any $n \in \mathcal{F}$ then
      Subproblem infeasible, go to step 1.
    while $\underline{B}_n = \overline{B}_n$ for any $n \in \mathcal{F}$ and $|\mathcal{F}| > 0$ do
      Fix $b_n$ for $n$ such that $\underline{B}_n = \overline{B}_n$. Update $\mathcal{F}$, $\mathcal{K}$, $C(\mathbf{b}_{\mathcal{K}})$, $LB = \max\{C(\mathbf{b}_{\mathcal{K}}), LB\}$.
      if $LB \geq C(\mathbf{b}_I)$ then
        Go to step 1.
      2) Update parameters $\mathbf{Q}_{\text{eff}}$, $\mathbf{c}_{\text{eff}}$, $\gamma_{\text{eff}}$.
      3) Determine candidate ranges $\underline{B}_n, \dots, \overline{B}_n$ for $n \in \mathcal{F}$.
      if $\underline{B}_n > \overline{B}_n$ for any $n \in \mathcal{F}$ then
        Go to step 1.
    if $|\mathcal{F}| \leq 1$ then
      4) Subproblem already solved or solve 1-D subproblem $\longrightarrow (\mathbf{b}_{\mathcal{K}}, \mathbf{b}_{\mathcal{F}})$.
      if $C(\mathbf{b}_{\mathcal{K}}) + C(\mathbf{b}_{\mathcal{F}}) < C(\mathbf{b}_I)$ then
        Update $\mathbf{b}_I$ and prune list. Go to step 1.
    5) (Optional for subproblems) Generate feasible solution $\mathbf{b}_{\mathcal{F}}$ using heuristic algorithm
    of Section 5.3.
    if $C(\mathbf{b}_{\mathcal{K}}) + C(\mathbf{b}_{\mathcal{F}}) < C(\mathbf{b}_I)$ then
      Update $\mathbf{b}_I$ and prune list (possibly including current subproblem).
    if $|\mathcal{F}| \geq N_{\text{min}}$ then
      6) Solve linear or diagonal relaxation and update $LB$.
      if $LB \geq C(\mathbf{b}_I)$ then
        Go to step 1.
    Determine $m$ from (6.7.1).
    for $b_m = \underline{B}_m, \dots, \overline{B}_m$ do
      7) Create new subproblem with $b_m$ fixed to current value and add to list.
    Go to step 1.
```

we have a solution $\mathbf{b}_{\mathcal{K}}$ to the subproblem. If there is only one free coefficient, the subproblem can be solved easily. In the one-dimensional case, all of the candidate values $\underline{B}_{\mathcal{F}}, \ldots, \overline{B}_{\mathcal{F}}$ are known to be feasible. We determine the lowest possible cost for the single free coefficient $b_{\mathcal{F}}$, either based on the smallest magnitude for the cost function $C_{\mathrm{NLZ}}$ or using a lookup table for $C_{\mathrm{SPT}}$, and determine the value of $b_{\mathcal{F}}$ by quantizing $c_{\mathcal{F}}$ using the minimum cost. If the solution resulting from this step has a lower cost than the incumbent solution, the incumbent solution is replaced and subproblems are pruned as appropriate.

5. *Generate a feasible solution:* In the first iteration of the algorithm in which the root problem is processed, we use the heuristic algorithm of Section 5.3 to obtain a feasible solution, which automatically becomes the incumbent solution since the initial solution $\mathbf{b}_I = \mathbf{c}$ has infinite cost. This step is optional for subsequent subproblems and we consider two variants in Section 7.1, one in which the heuristic algorithm is used only for the root problem, and the other in which it is used for all subproblems. This choice involves the same trade-off as in Algorithm 2 between more rapid improvement of the incumbent solution and increased computation per iteration. The current situation is different however because it is difficult to obtain a feasible solution without a substantial algorithm. This is to be contrasted with Algorithm 2 where we were ensured of a solution with at most $|\mathcal{F}| - 1$ non-zero components even without running an algorithm. In addition, the heuristic algorithm of Section 5.3 is significantly more complex than the successive thinning algorithm used in Algorithm 2.

6. *Solve relaxation:* As in Algorithm 2, if the dimension of the current subproblem exceeds a threshold $N_{\min}$, we solve a relaxation in an attempt to improve upon the current value of $LB$ and prune the current subproblem. For linear relaxations, the MATLAB function `fmincon` is used to solve the dual, either (6.3.7) or (6.3.18). For diagonal relaxations, we determine the matrix $\mathbf{D}$ that minimizes the volume of the enclosing ellipsoid as discussed in Section 6.5 and then apply one of the algorithms in Section 6.4.2. Since the minimum-volume enclosing ellipsoid depends only on the parameter $\mathbf{Q}_{\mathrm{eff}}$, which in turn depends only on the subset $\mathcal{F}$, the optimal $\mathbf{D}$ is the same for subproblems with the same subset $\mathcal{F}$. By saving some of the values of $\mathbf{D}$ from previous iterations, for example in a cache, we can avoid repeating the associated

computations. In Section 7.1, we compare both types of relaxations under different values of $N_{\min}$.

7. *Create new subproblems:* We wish to create as few subproblems as possible by choosing a coefficient $b_m$ for which the number of candidate values is the fewest. Since the number of candidate values is equal to $\overline{B}_n - \underline{B}_n + 1$, $m$ is determined by

$$m = \arg\min_{n \in \mathcal{F}} \overline{B}_n - \underline{B}_n. \tag{6.7.1}$$

We fix $b_m$ to each candidate value in turn and create a new subproblem with the subsets $\mathcal{F}$ and $\mathcal{K}$ and the vector of fixed coefficients $\mathbf{b}_{\mathcal{K}}$ updated accordingly. We also set the initial lower bound for each child subproblem equal to $\max\{C(\mathbf{b}_{\mathcal{K}}), LB\}$, where $C(\mathbf{b}_{\mathcal{K}})$ is the updated cost of the fixed coefficients in the child subproblem and $LB$ is the final lower bound for the parent subproblem.

## Chapter 7

# Bit-efficient filter design under a quadratic constraint: Numerical experiments and design examples

In this chapter, the heuristic algorithms of Section 5.3 and the branch-and-bound algorithm of Chapter 6 are applied to a variety of examples. The experiments and design examples are similar to those in Chapter 4. In Section 7.1, we again use randomly generated problem instances to validate properties of the design algorithms. In particular, it is verified that the relaxations discussed in Chapter 6 can reduce the complexity of the branch-and-bound algorithm. The experimental results also confirm some of the dependences seen in Section 6.6.

In Section 7.2, the algorithms are used to design bit-efficient equalizers for multipath channels. The examples considered are the same as those in Sections 4.2.1–4.2.2, and we observe a similar trade-off between the MSE and the filter complexity, now measured in terms of the number of NLZ bits or the number of SPTs. The behavior with respect to other parameters such as the input SNR is also similar to before.

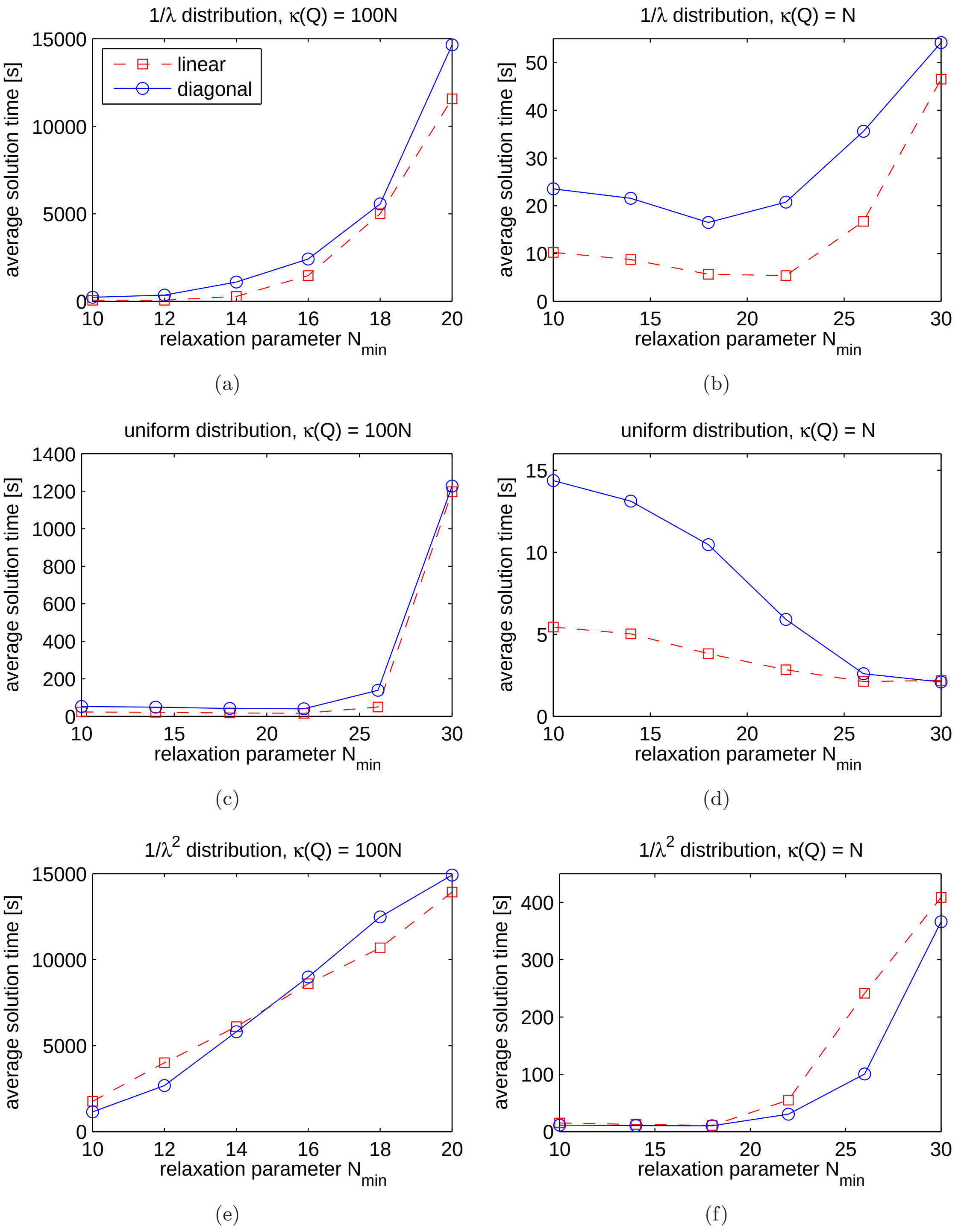
## 7.1 Numerical experiments

The algorithms for bit-efficient design are evaluated using synthetic examples generated randomly as described in Section 6.6. Briefly, the matrix $\mathbf{Q}$ is generated by drawing its eigenvalues from one of three distributions and its eigenvectors from a uniform distribution over the unit sphere. In the present experiments, the number of dimensions $N$ is fixed to either 30 or 20 and the condition number $\kappa(\mathbf{Q})$ is chosen equal to $N$ or $100N$. The values of $N$ are significantly lower than in the parallel set of experiments in Section 4.1 because bit-efficient design is much more difficult than sparse design. The components of the ellipsoid center $\mathbf{c}$ are drawn from an equally weighted mixture of a uniform distribution over small values and a power-law distribution over large values. We use maximum wordlengths of either 8 or 16 and perform scaling as described in Section 5.1.1 to convert all quantization levels to integers. The parameter $\gamma$ is chosen to ensure that each problem instance has at least one feasible solution. Further details of the instance generation procedure are given in Section 6.6.

As in Section 4.1, we consider the effect of different choices in the branch-and-bound algorithm (Algorithm 9), specifically the choice between linear and diagonal relaxations and the value of the parameter $N_{\min}$ that determines the minimum subproblem dimension for which relaxations are solved. As discussed earlier, varying $N_{\min}$ allows for an exploration of the trade-off inherent in solving relaxations: When $N_{\min}$ is small, more relaxations are solved, lower bounds are better, and more subproblems are pruned, but the average computational load per subproblem is higher. The relative efficiency of different algorithm variants is measured by the average solution time for a MATLAB implementation as in Section 4.1. The experimental results show that in most cases, solving relaxations decreases the overall complexity and the lowest average solution time occurs at a small to intermediate value of $N_{\min}$. In the remaining cases, the lowest complexity results when the number of relaxations is minimized, i.e., for $N_{\min} = N$.

The values of $N$ have been chosen low enough so that the experiments can be repeated many times and each instance can be solved repeatedly using different algorithm settings. Nevertheless, the random generation procedure can occasionally generate instances that are very difficult to solve. For practical reasons, the solution time is limited to one hour for each combination of parameters, which results in a small fraction of instances not being solved to optimality within that time. To estimate the true solution time, we assume that the optimality gap, i.e., the difference between the upper and lower bounds on the optimal cost, decreases linearly with time. The solution time can then be extrapolated from the initial and final optimality gaps and the time of termination. In our experience, this method is likely to yield an underestimate of the true solution time because the optimality gap usually does not decrease at a constant rate, instead decreasing more quickly near the beginning and more slowly thereafter.

In Fig. 7-1, we show the solution times, averaged over 45 trials, as a function of $N_{\text{min}}$ for instances of problem (5.1.1) and wordlength $P = 8$. The different panels correspond to different eigenvalue distributions and condition numbers while the two line types indicate the use of either linear or diagonal relaxations. The number of dimensions $N$ is equal to 20 in panels (a) and (e) and 30 elsewhere. In all cases, the minimum solution time is attained when $N_{\text{min}}$ is below its maximum value, implying that solving relaxations decreases complexity. Linear relaxations are seen to be slightly better than diagonal relaxations except for the $1/\lambda^2$ distribution. This differs from the situation in Section 4.1 with sparse design where linear relaxations do not reduce complexity at all and diagonal relaxations do not either at high condition numbers or under a uniform eigenvalue distribution. Also in contrast to Section 4.1, the dependence on the eigenvalue distribution and condition number is less clear here. The location of the minimum solution time does tend to shift from right to left as the eigenvalue distribution becomes more heavily weighted toward small values, implying that the relaxations are becoming more powerful. This agrees with the effect of the eigenvalue distribution on the approximation ratios seen in Section 6.6. On the other hand, it is somewhat surprising that the relaxations provide as much if not more gain when $\kappa(\mathbf{Q}) = 100N$ compared to $\kappa(\mathbf{Q}) = N$. The $\kappa(\mathbf{Q}) = 100N$ instances are more difficult as evidenced by the need to reduce $N$ from 30 to 20 and by the increase in solution times overall. An examination of solution times for individual trials suggests that the relaxations are most helpful for the most difficult instances, including ones for which the branch-and-bound algorithm does not converge within an hour unless $N_{\text{min}}$ is small. It appears therefore that the relaxations can limit the complexity of branch-and-bound in particularly unfavorable cases.



(a)

(b)

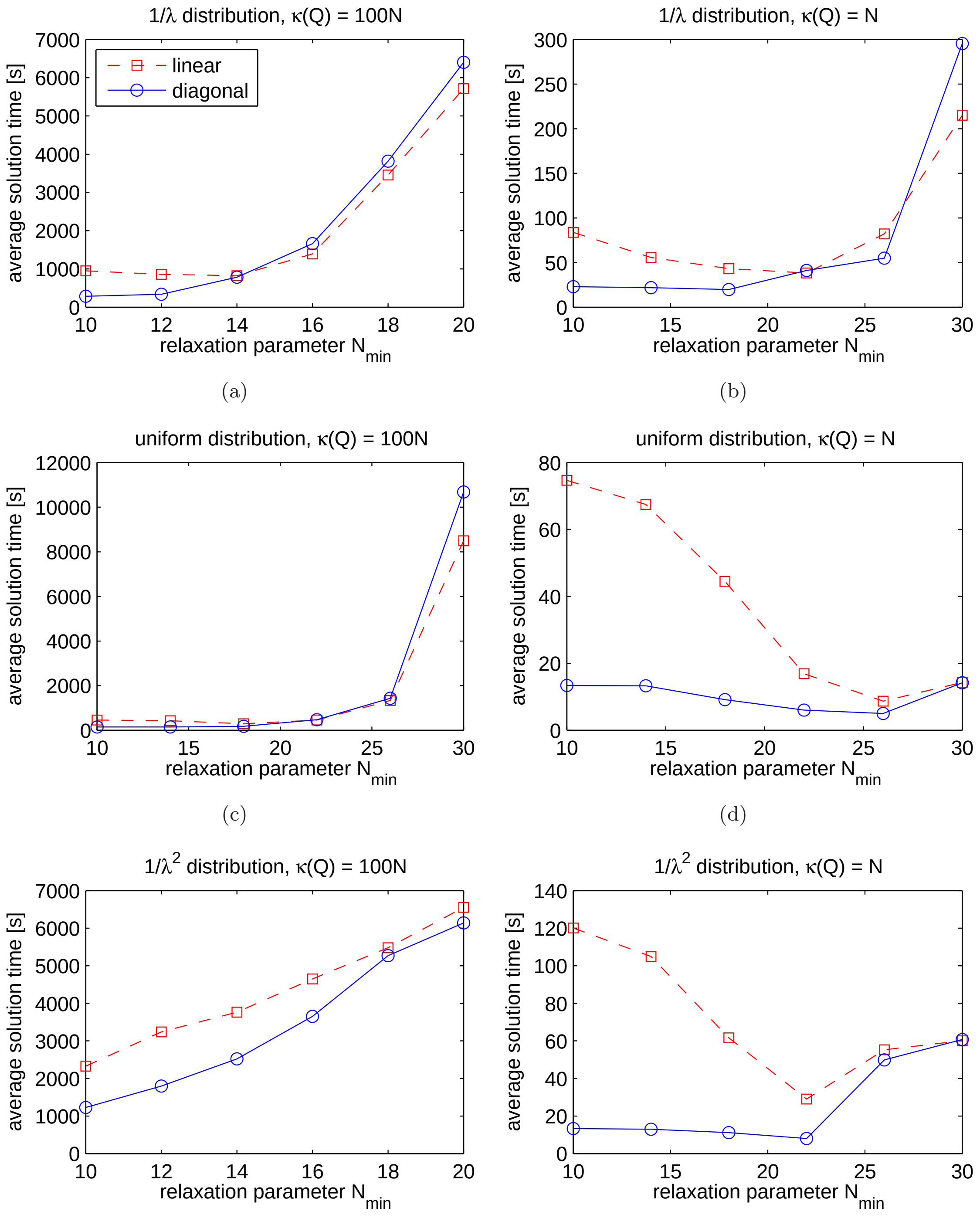
(c)

(d)

(e)

(f)

Figure 7-1: Average solution time as a function of the relaxation parameter $N_{\min}$ for problem (5.1.1) with $P = 8$.



(a)

(b)

(c)

(d)

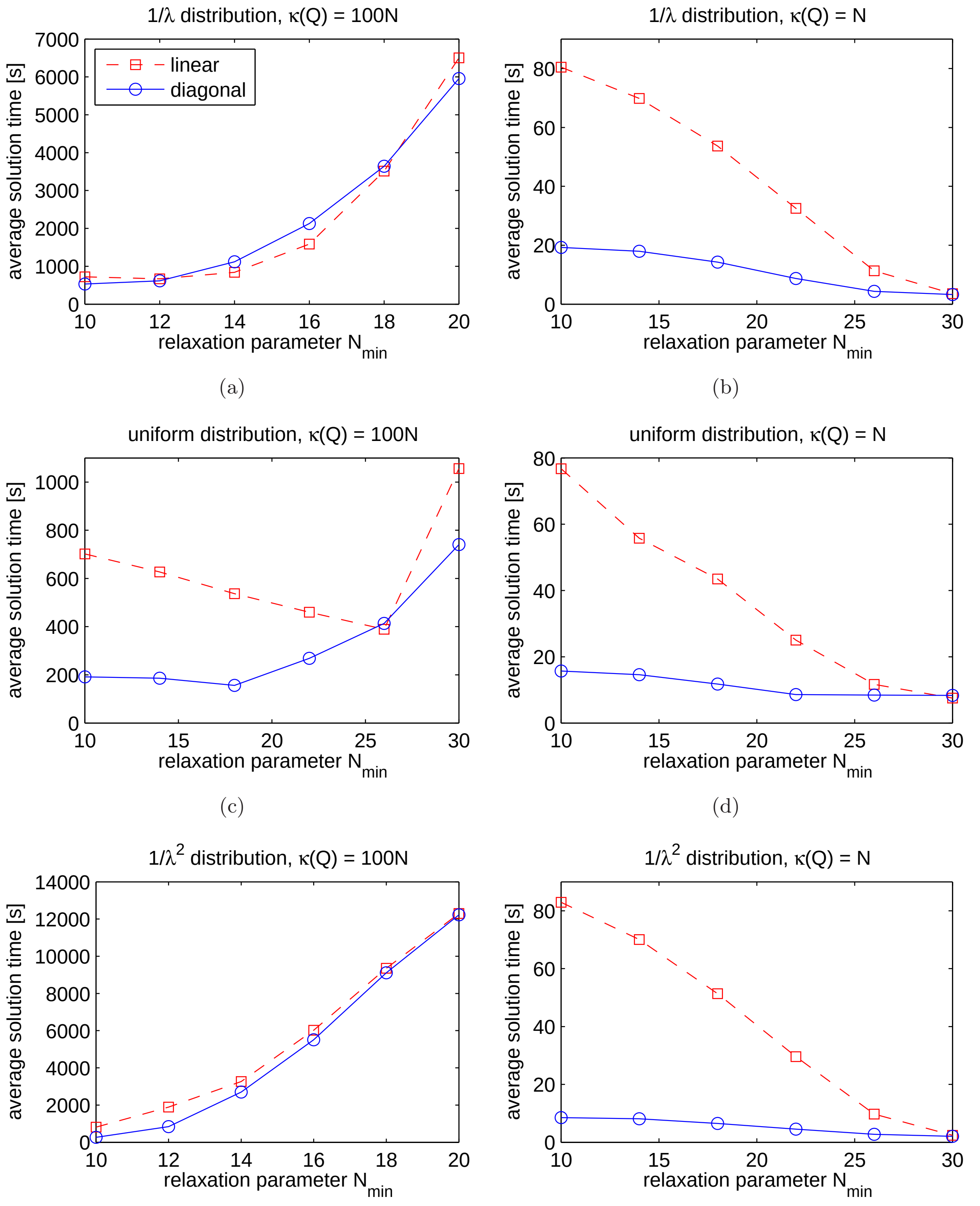
(e)

(f)

Figure 7-2: Average solution time as a function of the relaxation parameter $N_{\min}$ for problem (5.1.1) with $P = 16$.

Fig. 7-2 shows the average solution times for problem (5.1.1) and $P = 16$. As with $P = 8$, relaxations are beneficial for all eigenvalue distributions and condition numbers tested, and the location of the minimum solution time moves leftward as the eigenvalue distribution becomes more non-uniform. Diagonal relaxations are now seen to be slightly stronger than linear relaxations. Although the number of quantization levels for each coefficient has increased significantly, the increase in solution time is more modest. One explanation for this can be found in Section 6.6 where we observed that all of the lower bounds improve as the wordlength increases.

In Figs. 7-3 and 7-4, we plot the average solution times for problem (5.1.2) and wordlengths $P = 8$ and $P = 16$ respectively. It is seen that diagonal relaxations are generally preferable to linear relaxations in the SPT case as observed earlier in Section 6.6. For $P = 8$, solving relaxations is beneficial when $\kappa(\mathbf{Q}) = 100N$ and we see a similar relationship between the shape of the solution time curve and the eigenvalue distribution. However, for $\kappa(\mathbf{Q}) = N$, the lowest solution time occurs at $N_{\text{min}} = N = 30$. It can be inferred from this that the $\kappa(\mathbf{Q}) = N$ instances are relatively easy and do not require the use of relaxations, whereas the $\kappa(\mathbf{Q}) = 100N$ instances are more difficult and benefit more from the increased pruning afforded by solving relaxations. In contrast for $P = 16$, solving relaxations reduces complexity for all condition numbers and eigenvalue distributions considered. The increase in complexity relative to $P = 8$ is also milder than what we would naively predict based on the increase in quantization levels. The dependence on the eigenvalue distribution is in accordance with the previous plots.



(a)

(b)

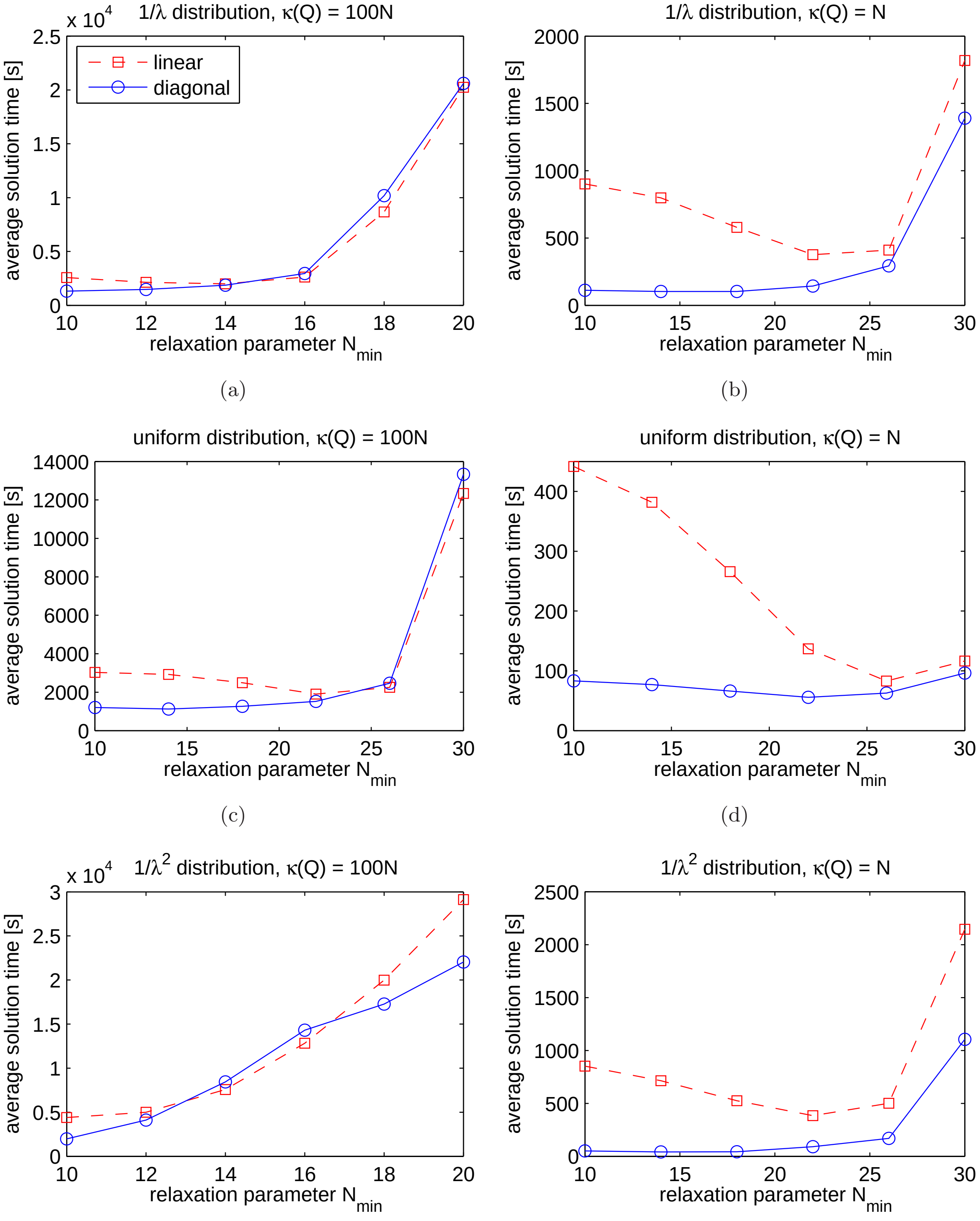
(c)

(d)

(e)

(f)

Figure 7-3: Average solution time as a function of the relaxation parameter $N_{\min}$ for problem (5.1.2) with $P = 8$.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 7-4: Average solution time as a function of the relaxation parameter $N_{\min}$ for problem (5.1.2) with $P = 16$.

A preliminary study of the growth of the solution time with dimension $N$ is shown in Fig. 7-5. For this experiment, instances were generated using a $1/\lambda$ eigenvalue distribution and a condition number of $\kappa(\mathbf{Q}) = N$. Three algorithm variants were tested: linear relaxations with $N_{\min} = 26$, diagonal relaxations with $N_{\min} = 26$ for the NLZ cost function and $N_{\min} = 14$ for the SPT cost, and no relaxations at all except for the root problem. The number of trials averaged is 50, which appears to be insufficient since the plotted solution times do not increase monotonically as expected. The results confirm that the use of relaxations can reduce complexity over a range of problem dimensions. Unfortunately, the limitation of the solution time to one hour and our conservative method of extrapolating solution times appear to distort the growth rates when the solution times exceed $10^3$ s, typically above $N = 35$. This distortion is most apparent in panel (c) where the dotted black and dashed red curves seem to undergo a change in slope. A more extensive experiment could be conducted in future work.

We also consider the behavior of the branch-and-bound algorithm acting on individual instances. Fig. 7-6 plots upper and lower bounds on the optimal cost as functions of the number of iterations. As in Fig. 4-6, the lower bounds improve more quickly near the beginning, justifying the claim that our extrapolation of solution times underestimates the true values. For the instance of (5.1.1) in (a) with $P = 8$, solving linear relaxations leads to the greatest efficiency both in terms of number of iterations and total time, which is consistent with Fig. 7-1. For the instance in (b), diagonal relaxations result in the fewest iterations but the fastest time is achieved with no relaxations.

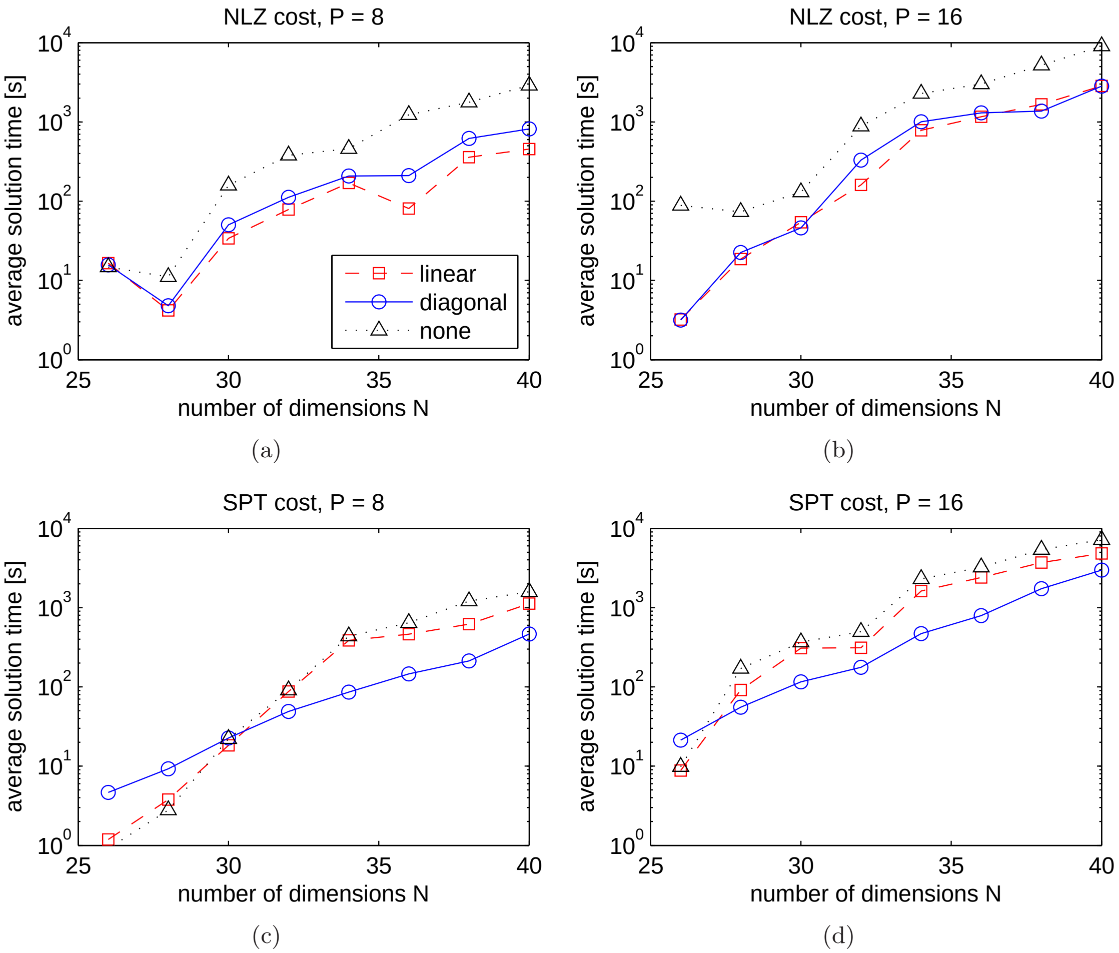
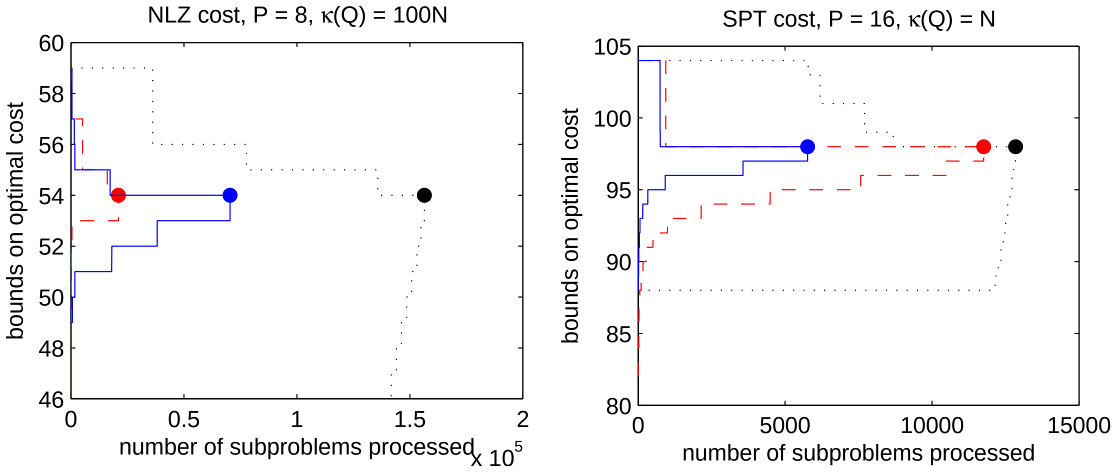


Figure 7-5: Growth of the average solution time with the number of dimensions $N$ for problems (5.1.1) and (5.1.2), a $1/\lambda$ eigenvalue distribution, and $\kappa(\mathbf{Q}) = N$.



(a)

(b)

Figure 7-6: Lower and upper bounds on the optimal cost as functions of the number of subproblems processed for specific instances of problems (5.1.1) and (5.1.2) generated using a $1/\lambda$ eigenvalue distribution. Dashed red line: linear relaxations, $N_{\min} = 18$; solid blue line: diagonal relaxations, $N_{\min} = 18$; dotted black line: no relaxations except for root problem. The filled circles indicate algorithm termination.

To evaluate the performance of the heuristic algorithms of Section 5.3, we show in Table 7.1 the average ratios between the cost of the initial heuristic solution and the final cost. The averages include only those trials in which the branch-and-bound algorithm converged to an optimal solution. The approximation ratios are better for the NLZ cost than for the SPT cost, better for $P = 16$ than for $P = 8$, and better for $\kappa(\mathbf{Q}) = N$ than for $\kappa(\mathbf{Q}) = 100N$. In particular, there is significant room for improvement in the SPT case at high condition number. These preferences were also observed in Section 6.6 for lower bounds. The effect of the eigenvalue distribution is less clear.

Table 7.1: Average approximation ratios for the heuristic algorithms of Section 5.3.

NLZ cost, $P = 8$

| | $\kappa(\mathbf{Q}) = 100N$ | $\kappa(\mathbf{Q}) = N$ |
|---|---|---|
| $\lambda(\mathbf{Q}) \sim 1/\lambda$ | 1.070 | 1.015 |
| $\lambda(\mathbf{Q}) \sim$ uniform | 1.029 | 1.008 |
| $\lambda(\mathbf{Q}) \sim 1/\lambda^2$ | 1.065 | 1.014 |

NLZ cost, $P = 16$

| | $\kappa(\mathbf{Q}) = 100N$ | $\kappa(\mathbf{Q}) = N$ |
|---|---|---|
| $\lambda(\mathbf{Q}) \sim 1/\lambda$ | 1.011 | 1.002 |
| $\lambda(\mathbf{Q}) \sim$ uniform | 1.012 | 1.001 |
| $\lambda(\mathbf{Q}) \sim 1/\lambda^2$ | 1.006 | 1.002 |

SPT cost, $P = 8$

| | $\kappa(\mathbf{Q}) = 100N$ | $\kappa(\mathbf{Q}) = N$ |
|---|---|---|
| $\lambda(\mathbf{Q}) \sim 1/\lambda$ | 1.153 | 1.032 |
| $\lambda(\mathbf{Q}) \sim$ uniform | 1.143 | 1.020 |
| $\lambda(\mathbf{Q}) \sim 1/\lambda^2$ | 1.246 | 1.029 |

SPT cost, $P = 16$

| | $\kappa(\mathbf{Q}) = 100N$ | $\kappa(\mathbf{Q}) = N$ |
|---|---|---|
| $\lambda(\mathbf{Q}) \sim 1/\lambda$ | 1.085 | 1.024 |
| $\lambda(\mathbf{Q}) \sim$ uniform | 1.069 | 1.015 |
| $\lambda(\mathbf{Q}) \sim 1/\lambda^2$ | 1.076 | 1.016 |

The overall conclusion from these experiments is that solving relaxations can significantly reduce the complexity of branch-and-bound for bit-efficient filter design, just as it does for sparse design. Moreover, complexity reductions are observed under a broader range of circumstances than with sparse design. Whereas previously only diagonal relaxations were beneficial and then only for low to moderate condition numbers and non-uniform eigenvalue distributions, now linear relaxations can also yield efficiency gains and can slightly outperform diagonal relaxations in the case of the NLZ cost function and $P = 8$. In addition, substantial improvements are seen at both high and low condition numbers. It should be noted however that the gains reported are relative and bit-efficient design remains much more computationally intensive than sparse design.

## 7.2 Design examples

In this section, we present examples in which the heuristic and branch-and-bound algorithms are used to design bit-efficient equalizers for multipath communication channels. In Section 7.2.1, we consider the idealized channel of Section 4.2.1, while in Section 7.2.2, the more realistic example of Section 4.2.2 is considered.

In general, the problem instances encountered in this section are very complex computationally, even more so than the synthetic examples in Section 7.1. The reason for this is that the MSE constraints are loose enough relative to the minimum MSE to permit hundreds or even thousands of potential quantization levels for each coefficient. Thus we do not aim to solve instances to optimality, instead terminating the branch-and-bound algorithm after a specified period of time. For the idealized example in Section 7.2.1, it will be seen that fairly tight lower bounds on the true optimal cost can be established relatively quickly. For the example in Section 7.2.2 however, the final lower bounds are not as strong even after a significantly longer time period. Nevertheless, the results clearly illustrate the nature of the trade-off between filter complexity and MSE.

### 7.2.1 Equalizers for an idealized multipath communication channel

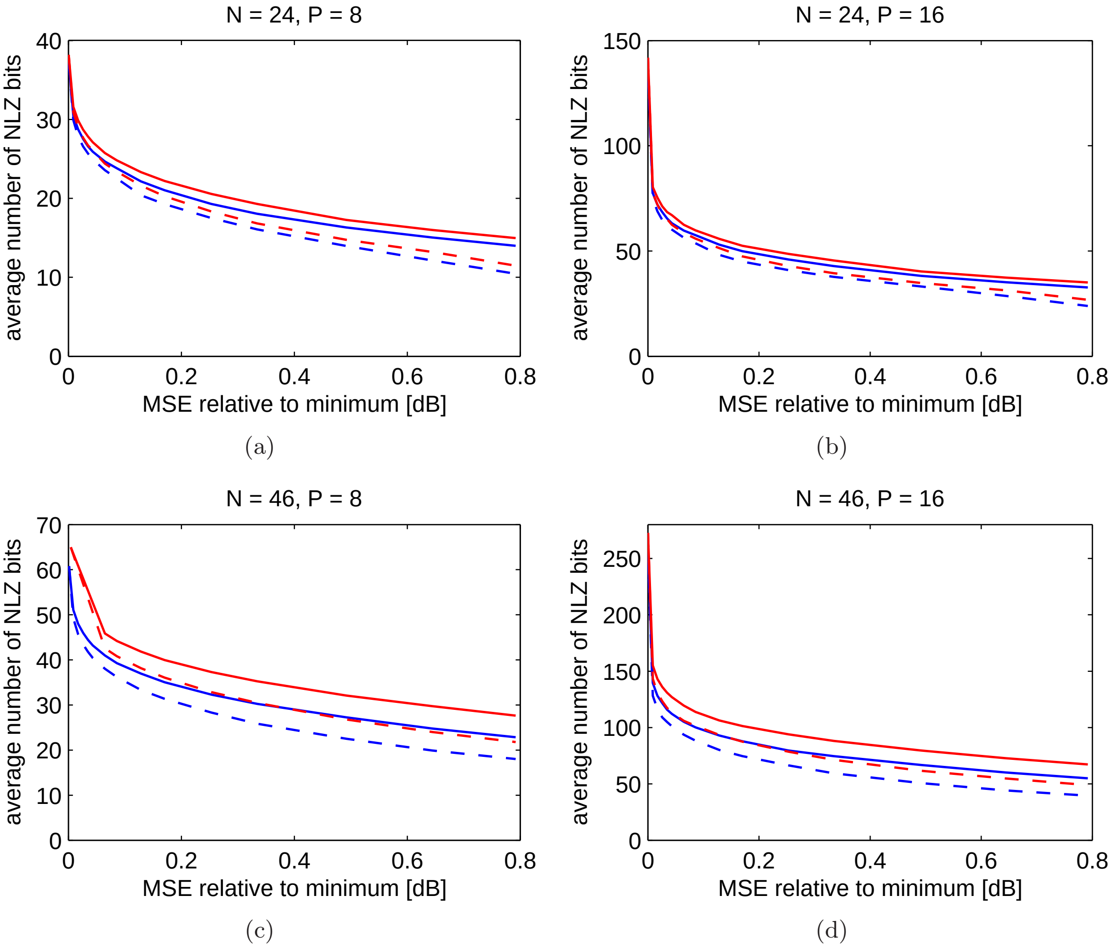
In the first example, the channel model is the same as in Section 4.2.1, and the parameters $\mathbf{Q}$, $\mathbf{f}$, and $\beta$ in constraint (2.1.3) are determined from the channel parameters as detailed in Section 2.1.2. We examine as before the trade-off between the MSE and the equalizer complexity, now measured in terms of the bit-based metrics. To conform with the assumption that the coefficient vector $\mathbf{b}$ is integer-valued, the parameters $\mathbf{Q}$ and $\mathbf{c}$ are rescaled as described in Section 5.1.1. The value of $\gamma$ used to calculate the scale factor is given by $\delta_{\max} - \delta_{\min}$, where $\delta_{\max}$ is the largest MSE tolerance to be considered and $\delta_{\min}$ is the MMSE corresponding to $\mathbf{b} = \mathbf{c}$. We consider two values for the maximum wordlength $P$, 8 and 16.

At one end of the trade-off, the MMSE equalizer corresponding to $\mathbf{b} = \mathbf{c}$ has infinite cost in general because the components of $\mathbf{c}$ can have arbitrary real values. To obtain a baseline design with finite cost, we use Algorithm 4 to determine a quantized solution $\mathbf{b}$ that approximately minimizes the left-hand side of (2.1.1). Recall from the discussion at the beginning of Section 5.2 that the value of this quadratic form can be viewed as the error due to quantizing $\mathbf{c}$, the MMSE solution. The allocation of bits in this initial quantization step is the maximal one, i.e., $(P, \ldots, P)$ for the NLZ cost and $(\lceil P/2 \rceil, \ldots, \lceil P/2 \rceil)$ for the SPT cost. We denote by $\hat{\delta}_{\text{min}}$ the MSE corresponding to the baseline solution.

To design equalizers with reduced cost, the MSE tolerance $\delta$ is chosen to be slightly larger than the baseline value $\hat{\delta}_{\text{min}}$. The parameter $\gamma$ is then given by $\delta - \delta_{\text{min}}$, where $\delta_{\text{min}}$ is still the continuous-valued MMSE. We will use the ratio $\delta/\delta_{\text{min}}$ as before to measure the performance degradation of the reduced-cost filters. With $\mathbf{Q}$, $\mathbf{c}$, $\gamma$, and $P$ determined as discussed above, the branch-and-bound algorithm of Section 6.7 is used to obtain a solution with reduced cost subject to the specified MSE tolerance. The branch-and-bound algorithm includes the heuristic algorithm of Section 5.3 as a first step and uses diagonal relaxations with the relaxation parameter $N_{\text{min}}$ equal to 15 for the NLZ cost function and 20 for the SPT cost.

Fig. 7-7 plots the number of NLZ bits, averaged over 100 amplitude pairs $(a_1, a_2)$, as a function of the MSE ratio $\delta/\delta_{\text{min}}$ for multipath delays $N_1 = 7$ and $N_2 = 23$, equalizer lengths $N = N_2 + 1$ and $N = 2N_2$, wordlengths $P = 8$ and $P = 16$, and $\text{SNR}_0 = 10, 25$ dB. Solid and dashed lines represent respectively average upper and lower bounds on the true optimal cost as determined by the branch-and-bound algorithm. The solution time was limited to only one minute in these experiments to permit a large number of repetitions. Similar to Fig. 4-7, there is a steep decrease in the equalizer cost as soon as the MSE ratio exceeds 0 dB, followed by a much more gradual asymptote. The number of NLZ bits is slightly higher for the larger SNR value and the shape of the trade-off curve is very similar for different lengths and wordlengths.

Fig. 7-8 shows the trade-off between the average number of SPTs and the MSE under the same conditions as in Fig. 7-7. Similar behavior is observed. The main difference is that the number of SPTs required for $P = 16$ is no greater than for $P = 8$, except for the leftmost points corresponding to the baseline solutions. This is because any feasible solution for $P = 8$ is also feasible for $P = 16$ with the same cost, assuming that the highest power of two is the same (before integer rescaling) so that the powers of two available in the $P = 16$ case are a superset of those in the $P = 8$ case. In light of this fact, we have used the final solution for $P = 8$ to initialize the branch-and-bound algorithm for the corresponding $P = 16$ instance, resulting in solutions with the same or lower cost.



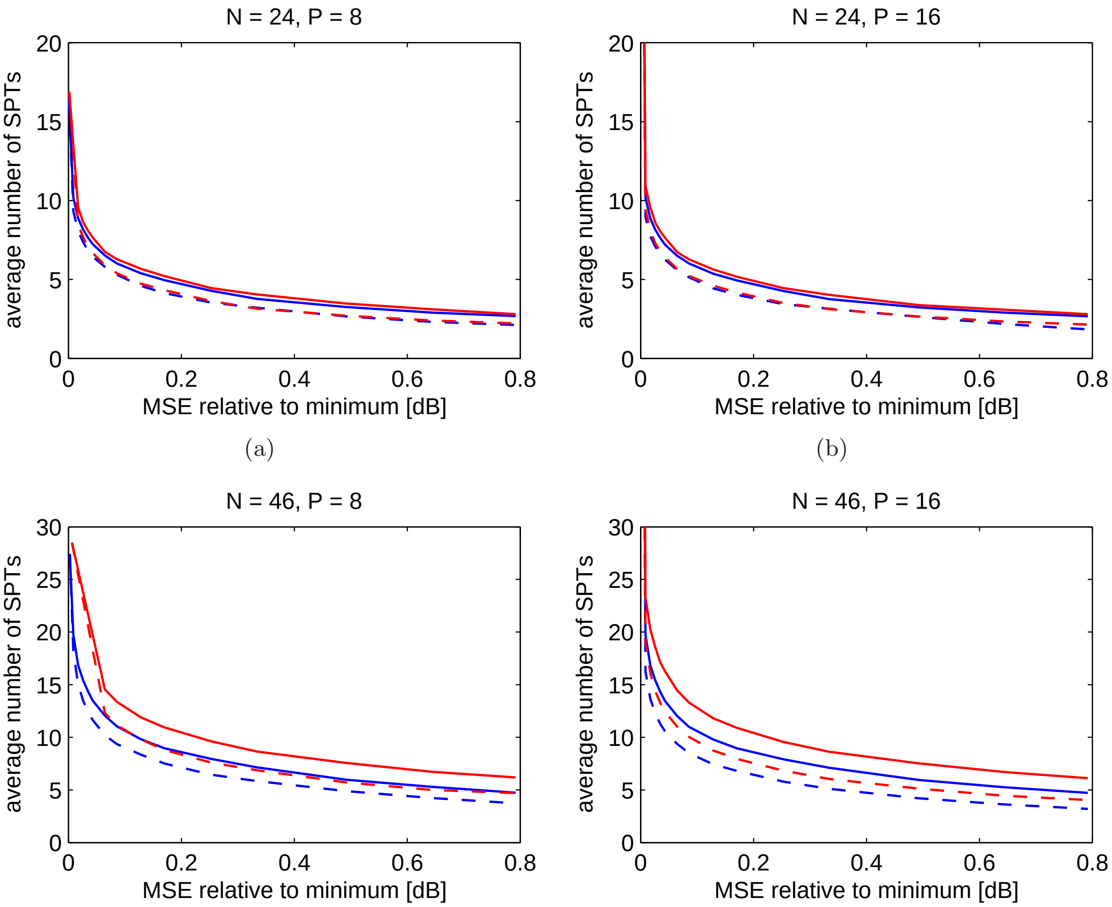
(a)

(b)

(c)

(d)

Figure 7-7: Average number of NLZ bits as a function of the MSE ratio $\delta/\delta_{\min}$. The lower blue curves correspond to $\text{SNR}_0 = 10$ dB and the upper red curves to $\text{SNR}_0 = 25$ dB. Solid and dashed lines represent upper and lower bounds on the true optimal cost.



(a) (b)

(c) (d)

Figure 7-8: Average number of SPTs as a function of the MSE ratio $\delta/\delta_{\text{min}}$. The lower blue curves correspond to $\text{SNR}_0 = 10$ dB and the upper red curves to $\text{SNR}_0 = 25$ dB. Solid and dashed lines represent upper and lower bounds on the true optimal cost. In panel (b), the largest (leftmost) values are 52.7 and 54.5 (beyond upper limit of plot) for $\text{SNR}_0 = 10, 25$ dB respectively, while in panel (d), the largest values are 98.5 and 103.6.

In Fig. 7-9, we plot the MMSE normalized by the signal power $\sigma_x^2$, the number of NLZ bits, and the number of SPTs as a function of the equalizer length $N$ for $P = 8$, $\delta/\delta_{\min} = 1.05$, and $\mathrm{SNR}_0 = 10$ dB. Circles and crosses represent upper and lower bounds on the optimal cost, obtained again by running the branch-and-bound algorithm for one minute and averaging over 100 $(a_1, a_2)$ pairs. As in Fig. 4-8, all three quantities display a staircase dependence on $N$ with transitions at integer combinations of $N_1$ and $N_2$, e.g. $30 = N_2 + N_1$, $39 = 2N_2 - N_1$ and $46 = 2N_2$ for $N_1 = 7$ and $N_2 = 23$. This effect is again due to the presence of large values at these locations in the MMSE equalizer as explained in Section 4.2.1.

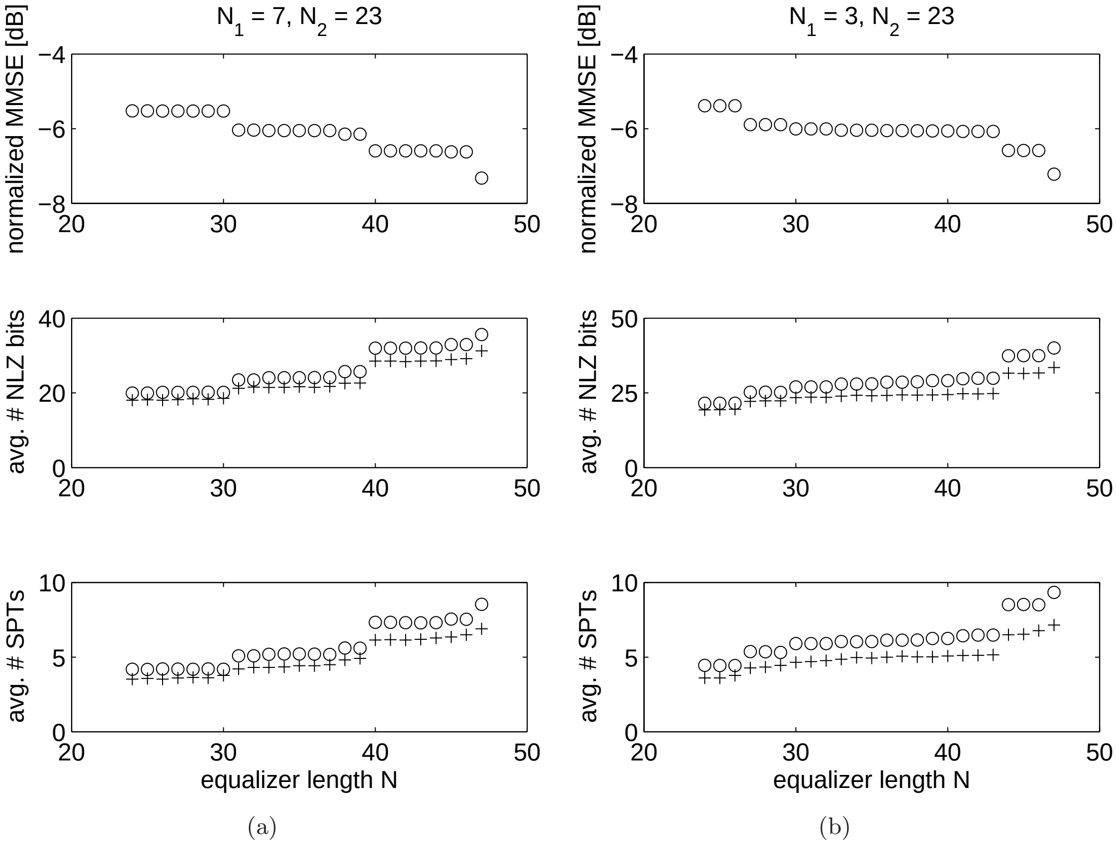
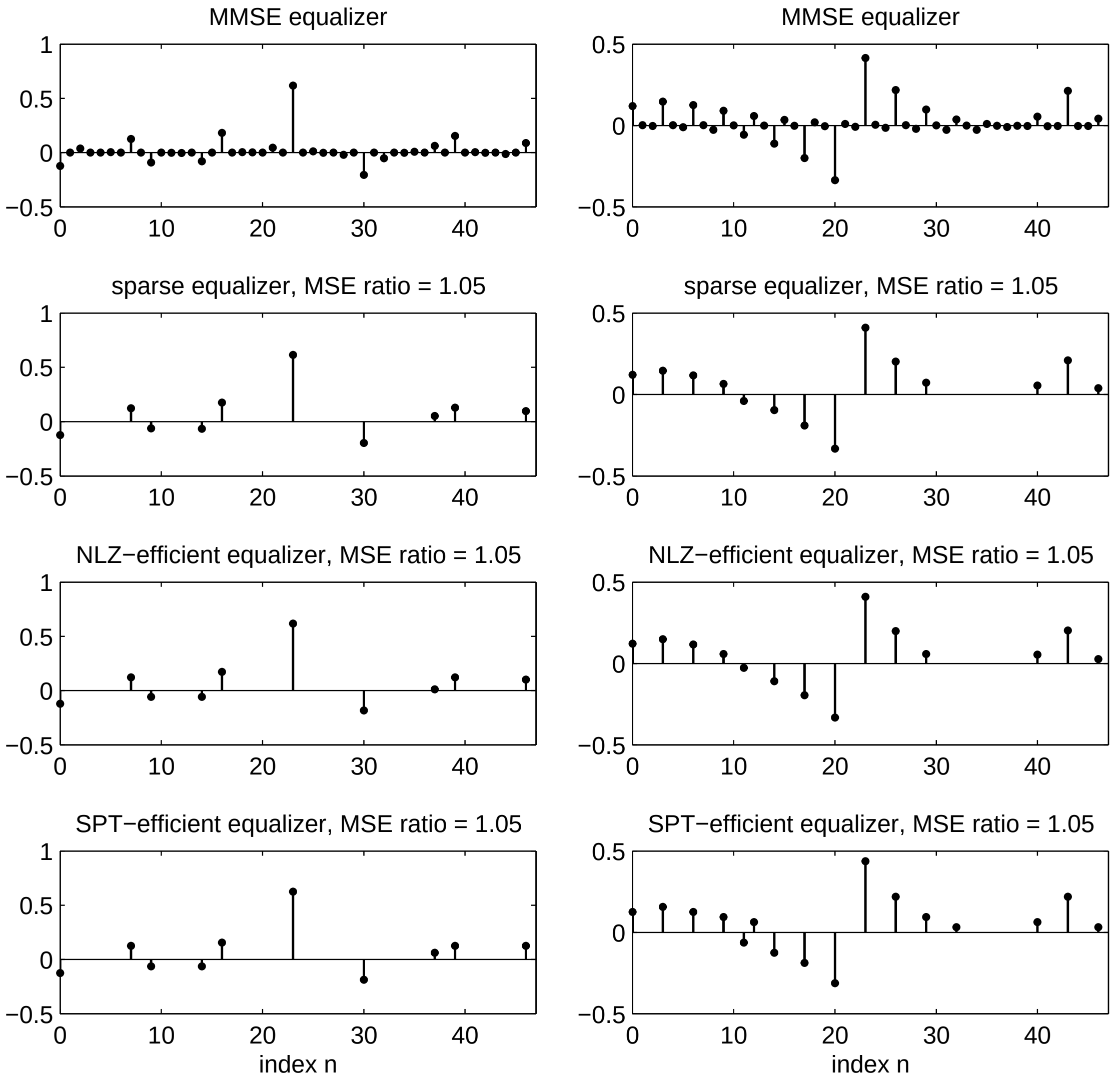


Figure 7-9: MMSE normalized by $\sigma_x^2$, average number of NLZ bits, and average number of SPTs as functions of the equalizer length $N$ with $P = 8$. Circles and crosses represent upper and lower bounds on the true optimal cost. The MSE for the bit-efficient equalizers is 5% higher than the corresponding MMSE.

Fig. 7-10 compares the MMSE, sparse, and bit-efficient equalizers obtained for two specific channel realizations with $N_1 = 7$, $N_2 = 23$ and $N_1 = 3$, $N_2 = 23$. The equalizer length is $N = 47$ and all other parameters are the same as in Fig. 7-9. The MMSE equalizer plots confirm that significant non-zero values occur only at integer combinations of $N_1$ and $N_2$. As seen before in Fig. 4-9, these are the only non-zero values retained in the sparse equalizers. Furthermore, the bit-efficient equalizers are equally as sparse, with the exception of the SPT-efficient equalizer in panel (b) that has two additional non-zero coefficients. It appears therefore that setting small coefficients to zero is an efficient choice in terms of the bit-based metrics as well. The remaining non-zero coefficients are also quantized in a cost-efficient way, for example by using only one or two SPTs.



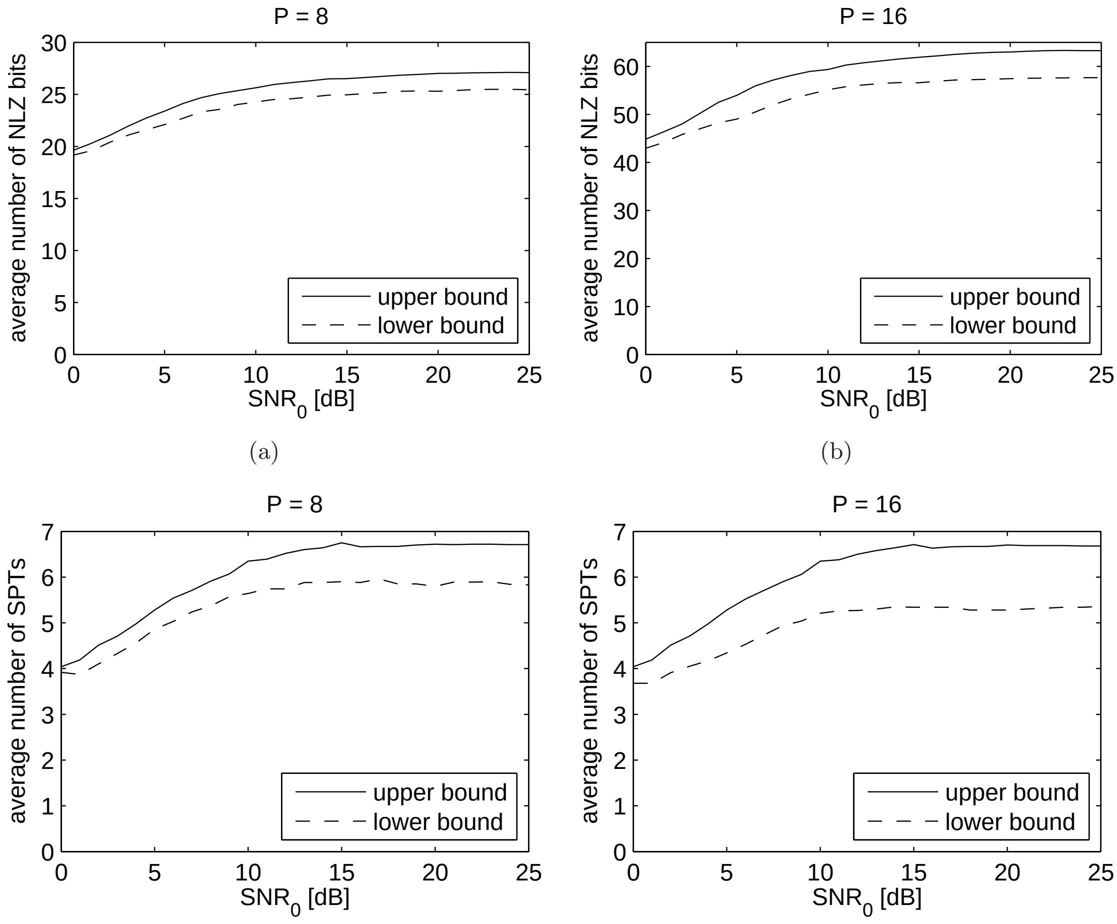
(a)

(b)

Figure 7-10: Coefficient values of MMSE, sparse, NLZ-efficient, and SPT-efficient equalizers for $N = 47$ and (a) $N_1 = 7$, $N_2 = 23$, (b) $N_1 = 3$, $N_2 = 23$. Zero values are omitted.

In Fig. 7-11, we examine the dependence of the number of NLZ bits and the number of

SPTs on the input signal-to-noise ratio $\mathrm{SNR}_0$. As in previous plots, we show upper and lower bounds on the optimal cost obtained after running the branch-and-bound algorithm for one minute and averaging 100 $(a_1, a_2)$ pairs. The equalizer cost increases nearly monotonically before saturating around $\mathrm{SNR}_0 = 15$ or 20 dB. This behavior is similar to that in Fig. 4-10 and can also be explained as an interpolation between the limits $\mathrm{SNR}_0 \to \infty$, in which case the MMSE equalizer converges to the channel inverse, and $\mathrm{SNR}_0 \to 0$, in which case it converges to a matched filter for the channel response.



(a)

(b)

(c)

(d)

Figure 7-11: Average number of NLZ bits and number of SPTs as functions of $\mathrm{SNR}_0$ for $N = 30$, $N_1 = 7$, $N_2 = 23$, and $\delta/\delta_{\min} = 1.02$.

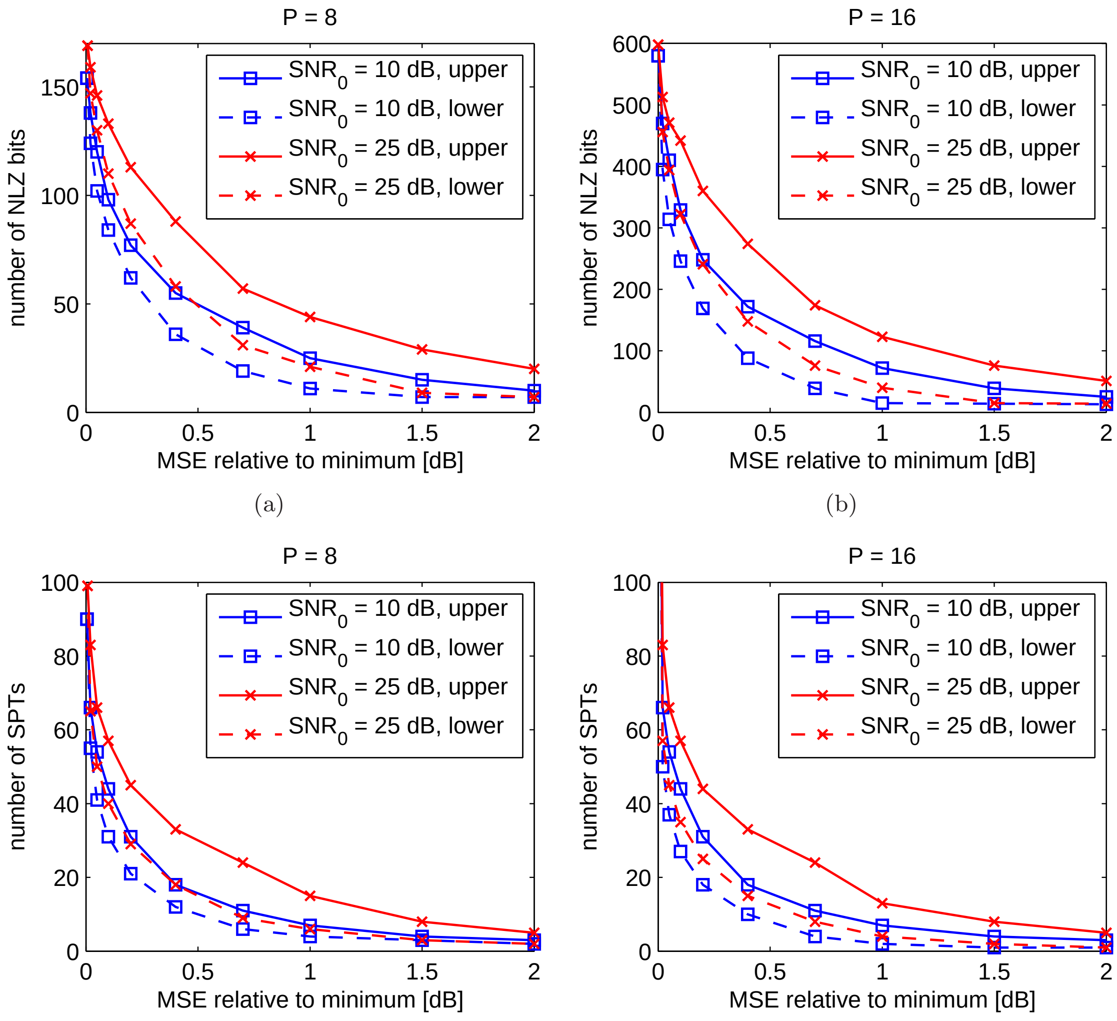
The closeness of the upper and lower bounds in Figs. 7-7–7-9 and 7-11, especially given the limited solution time, indicates that diagonal relaxations provide good approximations to the class of instances corresponding to idealized multipath channels. The quality of the bounds is due in part to the diagonal dominance of the matrix $\mathbf{Q}$ in this setting. Although the lower bounds likely do not coincide with the true optimal cost, they do suggest that the solutions provided by the heuristic algorithms of Section 5.3 (essentially represented by the upper bound curves) are not far from optimal. Despite the tightness of the bounds however, actually solving the instances to optimality is very computationally intensive in our experience because of the large number of quantization levels to be considered for each coefficient, i.e., the degree of branching in the branch-and-bound tree is high.

### 7.2.2 Equalizers for a realistic wireless communication channel

We now consider the HDTV broadcast channel used in Section 4.2.2. The reader is referred to Section 4.2.2 for the details of the channel model. The experimental setup, including the rescaling of parameters and the determination of baseline quantized solutions, is the same as in Section 7.2.1. Only the smallest equalizer length from Section 4.2.2, $N = 55$, is considered along with wordlengths of $P = 8$ and $P = 16$.

In Fig. 7-12, we plot the number of NLZ bits and the number of SPTs against the MSE ratio $\delta/\delta_{\min}$ for $P = 8, 16$ and $\mathrm{SNR}_0 = 10, 25$ dB. The MMSE values (normalized by $\sigma_x^2$) are $-5.74$ and $-7.37$ dB respectively for $\mathrm{SNR}_0 = 10, 25$ dB as reported in Section 4.2.2. For this example, the upper and lower bounds shown are the result of running the branch-and-bound algorithm for four hours. While the curves are not as steep as for the idealized channel, the trade-off between equalizer cost and MSE is still quite favorable. In particular, for any MSE tolerance above the minimum value, the number of SPTs required for $P = 16$ is no greater than that for $P = 8$, as noted in Section 7.2.1. We also see that the lower bounds are not as tight as in Figs. 7-7 and 7-8 despite the much longer solution time. This is not unexpected given the greater richness of the current example. Furthermore, the progress of the lower bounds is very slow due to the large number of branches created in the branch-and-bound tree. Even doubling the solution time is unlikely to significantly improve the bounds.

Fig. 7-13 compares the MMSE, sparse, and bit-efficient equalizers for $P = 8$ and $\mathrm{SNR}_0 = 10$ dB, with $\delta/\delta_{\min} = 0.2$ dB for the non-MMSE equalizers. As seen earlier in Fig. 7-10, minimizing the number of NLZ bits or the number of SPTs also tends to result in sparse designs. In this example, the bit-efficient equalizers are slightly less sparse than the maximally sparse equalizer. In sparse design, the only way to reduce the filter cost is to set coefficients to zero, but there are additional, finer-grained ways of doing so in bit-efficient



(a)

(b)

(c)

(d)

Figure 7-12: Number of NLZ bits and SPTs as functions of the MSE ratio $\delta/\delta_{\min}$ for an equalizer length of $N = 55$. In panel (d), the largest (leftmost) values are 223 and 253 (beyond upper limit of plot) for $\mathrm{SNR}_0 = 10, 25$ dB respectively.

design. Specifically, the number of NLZ bits can be decreased by shrinking non-zero coefficient values, and the number of SPTs can be decreased by coarsely quantizing to values composed of only one or two SPTs.

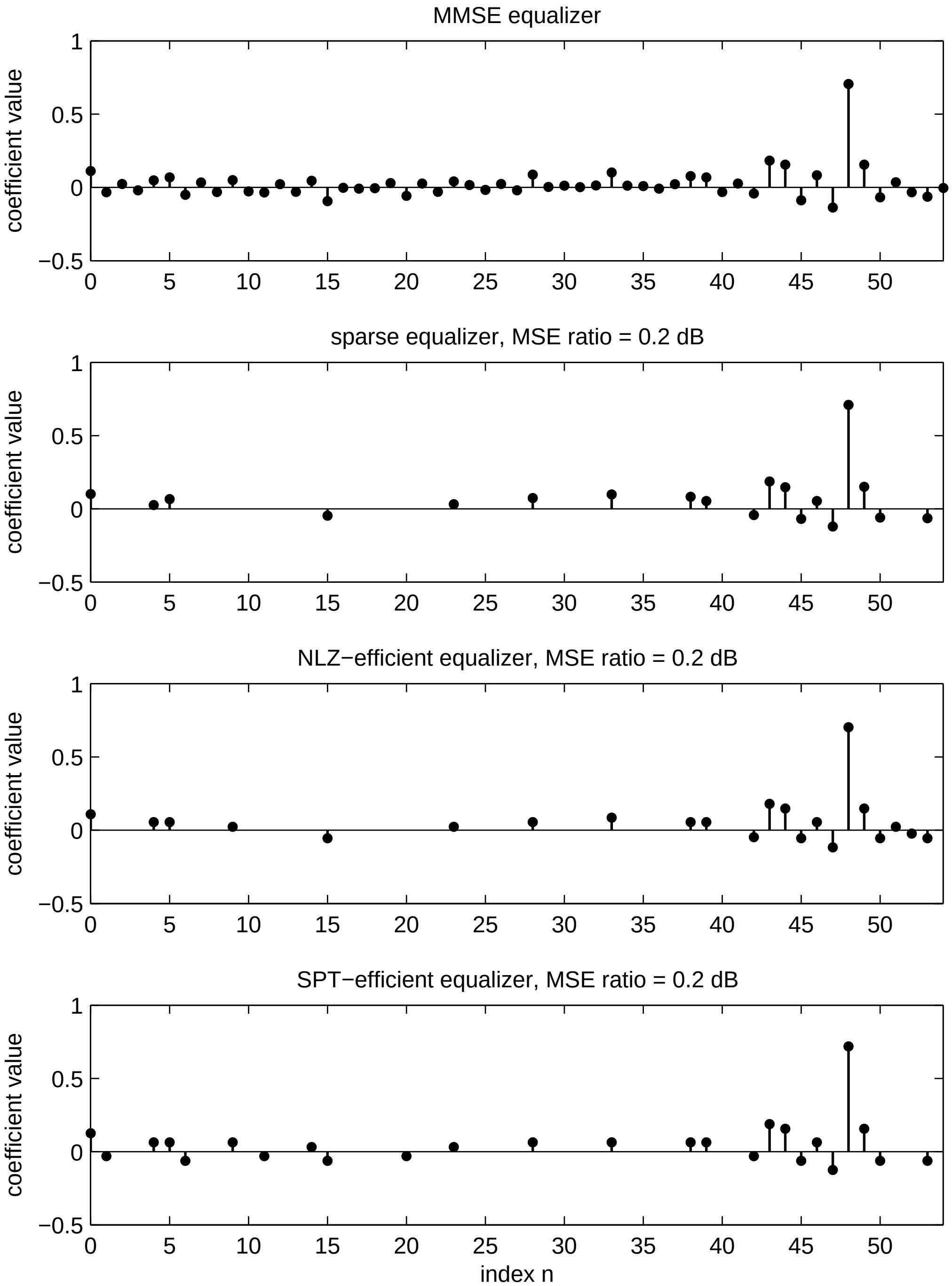


Figure 7-13: Coefficient values for the length 55 MMSE equalizer for $\mathrm{SNR}_0 = 10$ dB and corresponding sparse and bit-efficient equalizers with MSE ratio $\delta/\delta_{\min} = 0.2$ dB and $P = 8$. Zero values are omitted.

# Chapter 8

# Sparse filter design under a Chebyshev constraint

In this chapter, we address the design of sparse filters under a Chebyshev constraint on the frequency response. Several of the techniques in Chapters 2–3 for the quadratically constrained problem can be extended to the Chebyshev case, including the methods of successive thinning, branch-and-bound, and linear relaxation. On the other hand, we have not been able to identify special cases that admit efficient and exact solutions. The diagonal relaxation also does not appear to generalize in a tractable manner. Even for the techniques that do generalize, the computational complexity is now significantly higher than in the quadratic case. For this reason, we do not attempt to develop a complete optimal algorithm in this chapter. Optimal algorithms may be considered in future work.

In Section 8.1, the design problem considered in this chapter is formulated in greater detail and is also contrasted with the sparse linear inverse problem discussed in Section 2.1. In Section 8.2, we extend the successive thinning algorithm of Section 2.3 to the Chebyshev error criterion. An alternative thinning rule and other efficiency enhancements are proposed to combat the increased computational complexity. Section 8.3 develops an alternative approximate algorithm based on the minimization of the family of functions known as the $p$-norms for $0 < p \leq 1$. Sections 8.4–8.6 extend the techniques of Sections 3.1–3.3, namely branch-and-bound, identification of coefficients for which a zero value is infeasible, and linear relaxation. In Section 8.7, we discuss why a generalization of the diagonal relaxation does not seem to be tractable. The performance of the approximate algorithms in Sections 8.2–8.3 and the quality of the lower bounds in Sections 8.5–8.6 are evaluated by means of several design examples in Section 8.8. Both frequency-selective filters and beamformers as well as a frequency response equalizer are considered.

## 8.1 Problem formulation

We restrict our attention in this chapter to the design of linear-phase filters, in which case the problem reduces to real-valued approximation of a desired amplitude response. To further simplify the presentation, the formulation in this section focuses on causal, Type I linear-phase filters, i.e., filters with causal impulse responses that are even-symmetric about an integer index. The formulation can be generalized to other types of linear-phase filters through minor modifications.

As in earlier chapters, the non-zero impulse response values are represented by an $N$-dimensional vector $\mathbf{b}$. Taking into account the linear-phase constraint, $\mathbf{b}$ is now defined in terms of the impulse response values $h[n]$ as follows:

$$\begin{aligned} b_0 &= h[N-1], \\ b_n &= 2h[N-1-n] = 2h[N-1+n], \quad n = 1, 2, \dots, N-1. \end{aligned} \tag{8.1.1}$$

The frequency response corresponding to (8.1.1) takes the form $H(e^{j\omega}) = A(e^{j\omega})e^{-j\omega(N-1)}$, where

$$A(e^{j\omega}) = \sum_{n=0}^{N-1} b_n \cos(n\omega) \tag{8.1.2}$$

is the real-valued amplitude response used to approximate a desired response $D(e^{j\omega})$. With $A(e^{j\omega})$ defined in terms of $b_n$ in (8.1.2), the problem of sparse filter design under a Chebyshev constraint on the frequency response can be formulated as

$$\begin{aligned} \min_{\mathbf{b}} \quad & \|\mathbf{b}\|_0 \\ \text{s.t.} \quad & W(\omega)\left|A(e^{j\omega}) - D(e^{j\omega})\right| \le \delta_d \quad \forall\, \omega \in \mathcal{W}, \end{aligned} \tag{8.1.3}$$

where $W(\omega)$ is a strictly positive weighting function and $\mathcal{W}$ is a closed subset of $[0, \pi]$. The constraints in (8.1.3) ensure that the maximum weighted frequency response error over $\mathcal{W}$ is no greater than a desired tolerance $\delta_d$. As before, the parameter $N$ should be chosen large enough for solutions satisfying the frequency response specifications to exist. The length

$N$ corresponds to the maximum allowable number of delay elements, $2N-2$ to be precise, although fewer delays may be required in the final design if coefficients at the ends of the impulse response are zero.

Problem (8.1.3) has an infinite number of constraints, one for each frequency in $\mathcal{W}$. In the sequel, we will often approximate these constraints by a finite subset corresponding to frequencies $\omega_1, \omega_2, \ldots, \omega_K$. Following common practice, a good approximation can be achieved using $K \sim 10N$ frequencies distributed uniformly over $\mathcal{W}$ and including the endpoints of all intervals [1,25,110]. To represent the resulting constraints more compactly, we introduce the matrix $\mathbf{A}$ and the vector $\mathbf{d}$ with components given by

$$A_{kn} = W(\omega_k)\cos(n\omega_k), \quad k = 1, \ldots, K, \quad n = 0, \ldots, N-1, \tag{8.1.4a}$$

$$d_k = W(\omega_k)D(e^{j\omega_k}), \quad k = 1, \ldots, K. \tag{8.1.4b}$$

Then the approximation to (8.1.3) can be written as

$$\begin{aligned} \min_{\mathbf{b}} \quad & \|\mathbf{b}\|_0 \\ \text{s.t.} \quad & -\delta_d \mathbf{e} \leq \mathbf{A}\mathbf{b} - \mathbf{d} \leq \delta_d \mathbf{e}, \end{aligned} \tag{8.1.5}$$

where $\mathbf{e}$ is a $K$-dimensional vector of ones and each constraint on the absolute error has been rewritten as two linear constraints. Since problem (8.1.5) has a finite number of linear constraints, the set of feasible solutions is a polytope, and as will be shown in Section 8.5, this polytope is also bounded provided that $\mathbf{A}$ has full rank. We will make use of these properties of the feasible set at several points in this chapter.

In Chapter 2, it was argued that sparse filter design under a quadratic constraint, i.e. problem (2.0.1), differs significantly from the problem of obtaining sparse solutions to an underdetermined system of linear equations, i.e. (2.1.2). The same is true for the linearly constrained problem in (8.1.5). The most notable difference between (8.1.5) and (2.1.2) is due to the different shapes of the matrices $\mathbf{A}$ and $\mathbf{\Phi}$. Given that the system of equations $\mathbf{\Phi}\mathbf{x} = \mathbf{y}$ in (2.1.2) is underdetermined, $\mathbf{\Phi}$ has many fewer rows than columns and the set of feasible $\mathbf{x}$ is unbounded. In contrast, the number of rows in $\mathbf{A}$, $K$, must be much larger than the number of columns, $N$, in order for (8.1.5) to be a close approximation to (8.1.3), and consequently the set of feasible $\mathbf{b}$ is bounded. In addition, in (2.1.2), the residual $\mathbf{y} - \mathbf{\Phi}\mathbf{x}$ is bounded in terms of its 2-norm, whereas in (8.1.5), it is the weighted $\infty$-norm of the error that is constrained.

Problem (8.1.5) is computationally difficult because of the combinatorial nature of minimizing the zero-norm. Furthermore, unlike in Chapters 2 or 5, there do not appear to be special cases of (8.1.5) in which optimal solutions can be determined efficiently. Thus we are faced with two basic approaches: restricting attention to low-complexity algorithms and sacrificing a guarantee on optimality, or pursuing optimal solutions at significantly higher complexity. In Sections 8.2 and 8.3, we present two low-complexity algorithms based on different approximations to (8.1.5). A branch-and-bound framework for obtaining optimal solutions is outlined in Section 8.4.

## 8.2 Successive thinning

In this section, we apply the method of successive thinning described in Section 2.3 to the linearly constrained problem in (8.1.3). A previous version of the content of this section can be found in [111]. The overall strategy remains the same as before: in the $K$th iteration, we search for a feasible solution with $K$ zero-valued coefficients, restricting the search to those subsets of zero-valued coefficients that contain the subset of size $(K-1)$ chosen in the previous iteration. As discussed in Section 2.3, this search strategy is a substantial simplification of the combinatorial search required to ensure optimality. The algorithm terminates when the simplified search fails to yield a solution with one additional zero-valued coefficient.

An important difference between the present problem (8.1.3) and the quadratically constrained problem (2.0.1) in Chapter 2 is that it is more difficult computationally to determine whether a given subset of zero-valued coefficients is feasible. In the case of (2.0.1), feasibility could be verified through a closed-form expression (e.g. (2.2.3)), whereas for (8.1.3), an iterative method is required to solve the associated optimization problem. Motivated by the increased complexity, two different rules are proposed in this section for selecting the new coefficient to be constrained to zero in each iteration. The first rule is the same as the one used in Section 2.3. The new zero-valued coefficient is chosen to minimize the increase in the weighted Chebyshev error, and we will accordingly refer to the first rule as the *minimum-increase rule*. The second rule simplifies the search even further, constraining to zero the smallest coefficient in absolute value of the filter obtained in the current iteration.

We will refer to the second rule as the *smallest-coefficient rule*.

The successive thinning algorithm under the minimum increase rule requires in the $K$th iteration the solution of at most $N - K + 1$ linear optimization problems, one for each of the coefficients that are candidates for being constrained to zero. The total number of linear optimization problems is quadratic in $N$. Under the smallest coefficient rule, only one linear optimization is solved in each iteration for a total of at most $N$. Both rules result in dramatically lower complexity compared to an exact algorithm. Unlike in Section 2.3, we have not been able to identify special cases in which either of the thinning rules is guaranteed to yield an optimal solution. Notwithstanding the lack of guarantees, it will be demonstrated in Section 8.8 that successive thinning according to either of the rules can often produce filters with significantly fewer non-zero coefficients than conventional designs.

To describe the algorithm in more detail, we use $\mathcal{Z}$ as in Section 2.3 to denote the subset of coefficients constrained to a zero value, and $\mathcal{Y}$ to denote the complement of $\mathcal{Z}$. The iteration number is represented by a superscript $K$. We will assume in this section that the initial subset $\mathcal{Z}^{(0)} = \emptyset$ is empty, but this is not always the case as certain coefficients may be fixed to zero a priori, for example in systems with broken multipliers or array elements, or when successive thinning is used as a follow-on optimization as discussed in Section 8.3. In each iteration, an index $m$ is removed from $\mathcal{Y}^{(K)}$ and added to $\mathcal{Z}^{(K)}$, resulting in new subsets $\mathcal{Y}^{(K+1)}$ and $\mathcal{Z}^{(K+1)}$ respectively. We defer discussion of rules for selecting the index $m$ until Section 8.2.1.

Each iteration involves the solution of one or more instances of the following minimax optimization problem:

$$\begin{aligned} P: \quad & \min_{\delta, \mathbf{b}_{\mathcal{Y}}} \quad \delta \\ & \text{s.t.} \quad \delta \mathbf{e} + \mathbf{A}_{\mathcal{Y}} \mathbf{b}_{\mathcal{Y}} \geq \mathbf{d}, \\ & \qquad\;\; \delta \mathbf{e} - \mathbf{A}_{\mathcal{Y}} \mathbf{b}_{\mathcal{Y}} \geq -\mathbf{d}, \end{aligned} \tag{8.2.1}$$

where $\mathbf{A}_{\mathcal{Y}}$ is the submatrix of $\mathbf{A}$ formed from the columns indexed by $\mathcal{Y}$. The presence of $\mathbf{A}_{\mathcal{Y}}$ and $\mathbf{b}_{\mathcal{Y}}$ in (8.2.1) in place of $\mathbf{A}$ and $\mathbf{b}$ reflects the requirement that $b_n = 0$ for $n \in \mathcal{Z}$. If $\mathcal{Y} = \{0, \ldots, N-1\}$, i.e., no coefficients have been constrained to zero, (8.2.1) can be solved using the Parks-McClellan algorithm. Otherwise (8.2.1) can be solved by a general-purpose linear program solver.

The linear programming dual of problem (8.2.1) is given by

$$\begin{aligned} D: \quad \max_{\mathbf{p}^+,\mathbf{p}^-} \quad & \mathbf{d}^T(\mathbf{p}^+ - \mathbf{p}^-) \\ \text{s.t.} \quad & \mathbf{e}^T(\mathbf{p}^+ + \mathbf{p}^-) = 1, \\ & \mathbf{A}_{\mathcal{Y}}^T(\mathbf{p}^+ - \mathbf{p}^-) = \mathbf{0}, \\ & \mathbf{p}^+ \geq \mathbf{0}, \quad \mathbf{p}^- \geq \mathbf{0}, \end{aligned} \tag{8.2.2}$$

and has the same optimal value as the primal problem. The dual problem may be more efficient to solve than the primal depending on the linear program solver used. If the dual problem is solved, the optimal coefficient values $b_n$ for $n \in \mathcal{Y}$ are available as the Lagrange multipliers corresponding to the constraint $\mathbf{A}_{\mathcal{Y}}^T(\mathbf{p}^+ - \mathbf{p}^-) = \mathbf{0}$ in (8.2.2); the coefficients $b_n$ for $n \in \mathcal{Z}$ are zero by design.

Define $\delta^{(K)}$ to be the optimal value of (8.2.1) with $\mathcal{Y} = \mathcal{Y}^{(K)}$, i.e., the minimum error under the constraints $b_n = 0$ for $n \in \mathcal{Z}^{(K)}$, and $b_n^{(K)}$ to be the coefficient values corresponding to $\delta^{(K)}$. As alluded to in Section 8.1, we assume that $N$ is large enough so that the initial error $\delta^{(0)}$ is strictly less than the allowable tolerance $\delta_d$. Since problem (8.2.1) has one fewer variable when $\mathcal{Y} = \mathcal{Y}^{(K+1)}$ than with $\mathcal{Y} = \mathcal{Y}^{(K)}$, $\delta^{(K+1)} \geq \delta^{(K)}$ and the sequence $\{\delta^{(K)}\}$ is non-decreasing. Equivalently, the dual has one fewer constraint with $\mathcal{Y} = \mathcal{Y}^{(K+1)}$ than with $\mathcal{Y} = \mathcal{Y}^{(K)}$ and hence its optimal value cannot decrease. The algorithm terminates when $\delta^{(K+1)}$ first exceeds $\delta_d$ for some $K$, at which point the last feasible solution $\mathbf{b}^{(K)}$ is taken to be the final design. Note that this last solution cannot have zero values for any $n \in \mathcal{Y}^{(K)}$, as otherwise we would have a feasible solution with more than $K$ zero coefficients and the algorithm could continue. Furthermore, the final solution almost always satisfies the frequency response constraints with non-zero margin, i.e., the final error $\delta^{(K)}$ is strictly less than $\delta_d$. Thus the final design usually satisfies the constraints at all frequencies and not only the finite set of constraints in (8.1.5).

Given the framework established above, we discuss next the two variants of the algorithm under the minimum-increase and smallest-coefficient rules. Other coefficient selection rules are also possible.

### 8.2.1 Selection rules

Under the minimum-increase rule, the index $m$ is chosen to minimize the increase in the error $\delta^{(K+1)}$ relative to $\delta^{(K)}$. As in Section 2.3, the subset $\mathcal{Y}^{(K)}$ is divided into two subsets: the subset $\mathcal{F}^{(K)}$ consisting of candidates for addition to the subset $\mathcal{Z}^{(K)}$ of zero-valued coefficients, and the subset $\mathcal{U}^{(K)}$ corresponding to coefficients for which a zero value is no longer feasible. Initially, $\mathcal{F}^{(0)}$ is equal to $\mathcal{Y}^{(0)}$ and $\mathcal{U}^{(0)}$ is empty. In iteration $K$, we determine for every $p \in \mathcal{F}^{(K)}$ the minimum error $\delta^{(K)}(p)$ that results from removing $p$ from $\mathcal{Y}^{(K)}$, specifically by solving (8.2.1) with $\mathcal{Y} = \mathcal{Y}^{(K)} \backslash p$. The lowest error value becomes $\delta^{(K+1)}$ and $m$ is chosen as the minimizing index, i.e.,

$$\delta^{(K+1)} = \min_{p \in \mathcal{F}^{(K)}} \delta^{(K)}(p), \tag{8.2.3a}$$

$$m = \arg \min_{p \in \mathcal{F}^{(K)}} \delta^{(K)}(p). \tag{8.2.3b}$$

Then $m$ is added to $\mathcal{Z}^{(K)}$ and removed from $\mathcal{F}^{(K)}$. In addition, we also move from $\mathcal{F}^{(K)}$ to $\mathcal{U}^{(K)}$ those indices $p$ for which $\delta^{(K)}(p) > \delta_d$, i.e., those coefficients that no longer yield feasible solutions when set to zero. The subsets resulting from these modifications become the new subsets $\mathcal{Z}^{(K+1)}$, $\mathcal{F}^{(K+1)}$, and $\mathcal{U}^{(K+1)}$.

One difference compared to the quadratically constrained problem in Section 2.3 is that while it is possible to identify coefficients for which a zero value is no longer feasible, it is not possible to eliminate them from the problem. In particular, the coefficients $b_n$ for $n \in \mathcal{U}$ are still included in the optimization problem (8.2.1) since $\mathcal{U}$ is a subset of $\mathcal{Y}$. In the quadratic case and specifically in Appendix A.3, the subvector $\mathbf{b}_{\mathcal{U}}$ could be eliminated because we had a closed-form expression for the value that maximizes the margin in the constraint and thus makes the set of feasible $\mathbf{b}_{\mathcal{F}}$ as large as possible. A closed-form solution is not available in the present case and the optimization of $\mathbf{b}_{\mathcal{U}}$ is instead incorporated in (8.2.1).

In the case of the smallest-coefficient rule, the index $m$ in the $K$th iteration is chosen to correspond to the smallest of the optimal coefficients $b_n^{(K)}$ for $n \in \mathcal{Y}^{(K)}$, i.e.,

$$m = \arg \min_{n \in \mathcal{Y}^{(K)}} \left| b_n^{(K)} \right|. \tag{8.2.4}$$

The smallest-coefficient rule can yield results different from those of the minimum-increase rule because it does not take into full account the sensitivity of the error to each coefficient, which may be large even when the coefficient value is small. However, the smallest-coefficient rule can be regarded as a simplification of the minimum-increase rule in the following sense: Recall that the coefficients $b_n^{(K)}$, $n \in \mathcal{Y}^{(K)}$, can be interpreted as the Lagrange multipliers associated with the optimal solution to the dual problem (8.2.2). According to this interpretation, if the right-hand side of the constraint $\mathbf{A}_n^T(\mathbf{p}^+ - \mathbf{p}^-) = 0$ in (8.2.2) is changed from zero to a small value $y$, the optimal value of (8.2.2) is changed by an amount $b_n^{(K)} y$. Hence, if coefficient $b_n$ is constrained to be zero, or equivalently, if the constraint $\mathbf{A}_n^T(\mathbf{p}^+ - \mathbf{p}^-) = 0$ in (8.2.2) is relaxed, the marginal rate of increase of the optimal error is given by $\left|b_n^{(K)}\right|$. Choosing $m$ to correspond to the smallest $\left|b_n^{(K)}\right|$ thus yields the smallest marginal rate of increase and the smallest-coefficient rule is therefore an approximation to the minimum-increase rule in a marginal or local sense. In Section 8.8, it will be seen that the smallest-coefficient rule often yields a level of sparsity comparable to that of the minimum-increase rule.

We summarize below the steps in the successive thinning algorithm under both selection rules.

**Algorithm 10** Successive thinning under minimum-increase rule

```
Input: Parameters $\mathbf{A}$, $\mathbf{d}$, $\delta_d$.
Output: Sparse solution $\mathbf{b}$ to (8.1.5)
  Initialize: $K = 0$, $\mathcal{Y}^{(0)} = \mathcal{F}^{(0)} = \{0, \dots, N-1\}$, $\mathbf{b}^{(0)}$ = feasible solution to (8.1.5),
  $\delta^{(0)}$ = error associated with $\mathbf{b}^{(0)}$.
  while $\delta^{(K)} \le \delta_d$ do
    for $p \in \mathcal{F}^{(K)}$ do
      Compute $\delta^{(K)}(p)$ by solving (8.2.1) with $\mathcal{Y} = \mathcal{Y}^{(K)} \backslash p$.
    Determine $m$ and $\delta^{(K+1)}$ from (8.2.3).
    $\mathbf{b}^{(K+1)}$ = optimal coefficient values corresponding to $\delta^{(K+1)}$.
    $\mathcal{Y}^{(K+1)} = \mathcal{Y}^{(K)} \backslash m$.
    $\mathcal{F}^{(K+1)} = \mathcal{F}^{(K)} \backslash \left\{ m \cup \left\{ p : \delta^{(K)}(p) > \delta_d \right\} \right\}$.
    $K \leftarrow K + 1$.
  Return solution: $\mathbf{b} = \mathbf{b}^{(K-1)}$.
```

**Algorithm 11** Successive thinning under smallest-coefficient rule

```
Input: Parameters $\mathbf{A}$, $\mathbf{d}$, $\delta_d$.
Output: Sparse solution $\mathbf{b}$ to (8.1.5)
  Initialize: $K = 0$, $\mathcal{Y}^{(0)} = \{0, \ldots, N-1\}$, $\mathbf{b}^{(0)}$ = feasible solution to (8.1.5), $\delta^{(0)}$ = error associated with $\mathbf{b}^{(0)}$.
  while $\delta^{(K)} \leq \delta_d$ do
    Determine $m$ from (8.2.4).
    $\mathcal{Y}^{(K+1)} = \mathcal{Y}^{(K)} \backslash m$.
    Compute $\delta^{(K+1)}$ and $\mathbf{b}^{(K+1)}$ by solving (8.2.1) for $\mathcal{Y} = \mathcal{Y}^{(K+1)}$.
    $K \leftarrow K + 1$.
  Return solution: $\mathbf{b} = \mathbf{b}^{(K-1)}$.
```

### 8.2.2 Efficiency enhancements

In carrying out the successive thinning algorithm under either of the selection rules of Section 8.2.1, we are presented with linear optimization problems that differ by only one variable or one constraint. For example, with the minimum-increase rule, we start each iteration with the solution to (8.2.1) for $\mathcal{Y} = \mathcal{Y}^{(K)}$ and then re-solve (8.2.1) with $\mathcal{Y} = \mathcal{Y}^{(K)} \backslash p$ for all $p \in \mathcal{F}^{(K)}$. Similarly under the smallest-coefficient rule, the number of variables in (8.2.1) decreases by one in going from $\mathcal{Y} = \mathcal{Y}^{(K)}$ to $\mathcal{Y} = \mathcal{Y}^{(K+1)}$. In this subsection, we discuss methods for solving the linear optimization problems more efficiently given an optimal solution to a closely related problem, thereby improving the efficiency of the overall successive thinning algorithm. These methods are adapted from standard linear programming techniques [96].

In the remainder of the section, we denote by $\mathcal{Y}_1$ a generic subset of coefficients allowed to be non-zero, and by $\mathcal{Y}_2$ a subset that is identical to $\mathcal{Y}_1$ except for the absence of the index $m$. We assume that an optimal solution to either the primal (8.2.1) or the dual (8.2.2) has been obtained for $\mathcal{Y} = \mathcal{Y}_1$ and an optimal solution for $\mathcal{Y} = \mathcal{Y}_2$ is desired. In the case of the dual, the matrix $\mathbf{A}_{\mathcal{Y}_2}$ has one fewer column than $\mathbf{A}_{\mathcal{Y}_1}$, and therefore (8.2.2) with $\mathcal{Y} = \mathcal{Y}_2$ has one fewer constraint than with $\mathcal{Y} = \mathcal{Y}_1$. As a consequence, the existing solution for $\mathcal{Y}_1$ is also feasible for $\mathcal{Y}_2$ and can be used directly as an initial solution. The reader is referred to [96] for the details of this procedure.

The corresponding situation with the primal problem is not as straightforward. Since the coefficient $b_m$ is not constrained to zero in (8.2.1) when $\mathcal{Y} = \mathcal{Y}_1$, an optimal solution $(\delta^*, \mathbf{b}^*_{\mathcal{Y}_1})$ for $\mathcal{Y} = \mathcal{Y}_1$ is usually infeasible for $\mathcal{Y} = \mathcal{Y}_2$ and cannot be used directly as an initialization. To address this, we propose solving a modified problem that is identical to (8.2.1) except for an additional penalty in the objective function on the absolute value of $b_m$. To formulate the modified problem, we use (3.3.5) to express $b_m$ in terms of its positive and negative parts and then decompose the product $\mathbf{A}_{\mathcal{Y}_1}\mathbf{b}_{\mathcal{Y}_1}$ as

$$\mathbf{A}_{\mathcal{Y}_1}\mathbf{b}_{\mathcal{Y}_1} = \mathbf{A}_{\mathcal{Y}_2}\mathbf{b}_{\mathcal{Y}_2} + \mathbf{A}_m b_m = \mathbf{A}_{\mathcal{Y}_2}\mathbf{b}_{\mathcal{Y}_2} + \mathbf{A}_m(b_m^+ - b_m^-), \quad b_m^+ \geq 0,\ b_m^- \geq 0.$$

Then the modified problem can be formulated as

$$\begin{aligned} \hat{P}: \quad \min_{\delta,\mathbf{b}_{\mathcal{Y}_2},b_m^+,b_m^-} \quad & \delta + C\left(b_m^+ + b_m^-\right) \\ \text{s.t.} \quad & \delta\mathbf{e} + \mathbf{A}_{\mathcal{Y}_2}\mathbf{b}_{\mathcal{Y}_2} + \mathbf{A}_m\left(b_m^+ - b_m^-\right) \geq \mathbf{d}, \\ & \delta\mathbf{e} - \mathbf{A}_{\mathcal{Y}_2}\mathbf{b}_{\mathcal{Y}_2} - \mathbf{A}_m\left(b_m^+ - b_m^-\right) \geq -\mathbf{d}, \\ & b_m^+ \geq 0, \quad b_m^- \geq 0, \end{aligned} \tag{8.2.5}$$

where $C$ is a penalty constant.

An optimal solution to (8.2.1) for $\mathcal{Y} = \mathcal{Y}_1$ can be used to initialize the solution of (8.2.5). When $C$ is sufficiently large, it is expected that the final solution to (8.2.5) will have $b_m^+ = b_m^- = 0$, and consequently solving (8.2.5) becomes equivalent to solving (8.2.1) with $\mathcal{Y} = \mathcal{Y}_2$. The following theorem specifies values of $C$ sufficient for this equivalence to be exact.

**Theorem 8.** *If*

$$C > \max_{k=1,\dots,K} W(\omega_k), \tag{8.2.6}$$

*then* $\left(\hat{\delta}, \hat{\mathbf{b}}_{\mathcal{Y}_2}, \hat{b}_m^+, \hat{b}_m^-\right)$ *is an optimal solution to problem* (8.2.5) *if and only if* $\hat{b}_m^+ = \hat{b}_m^- = 0$ *and* $\left(\hat{\delta}, \hat{\mathbf{b}}_{\mathcal{Y}_2}\right)$ *is an optimal solution to problem* (8.2.1) *for* $\mathcal{Y} = \mathcal{Y}_2$.

The proof of Theorem 8 can be found in Appendix E.1.

In MATLAB implementations using the solver `linprog`, the techniques presented in this subsection can increase the speed of successive thinning algorithms by about 2–3 times compared to an implementation in which all linear programming problems are solved independently. Similar gains are expected for more specialized linear program solvers.

## 8.3 Sequential $p$-norm minimization

In this section, an alternative approximate algorithm for sparse filter design is developed based on a different approximation to problem (8.1.5). We consider the family of functions defined by

$$\|\mathbf{b}\|_p = \left( \sum_{n=0}^{N-1} |b_n|^p \right)^{1/p} . \tag{8.3.1}$$

for $0 < p \leq 1$. For convenience, we refer to $\|\mathbf{b}\|_p$ as a *$p$-norm* for all $p$ even though (8.3.1) defines a valid norm only for $p \geq 1$. The $p$-norm has the desirable property of providing an arbitrarily close approximation to the zero-norm as $p$ approaches zero. More precisely,

$$\lim_{p \to 0} \|\mathbf{b}\|_p^p = \|\mathbf{b}\|_0 . \tag{8.3.2}$$

We are thus led to consider optimization problems of the form

$$\begin{aligned} \min_{\mathbf{b}} \quad & \|\mathbf{b}\|_p^p \\ \text{s.t.} \quad & -\delta_d \mathbf{e} \leq \mathbf{A}\mathbf{b} - \mathbf{d} \leq \delta_d \mathbf{e} \end{aligned} \tag{8.3.3}$$

for small values of $p$. We refer to an optimal solution of (8.3.3) as a minimum $p$-norm solution, noting that the minimizer is not affected by replacing $\|\mathbf{b}\|_p^p$ with $\|\mathbf{b}\|_p$.

Problem (8.3.3) has also been considered in the context of sparse beamformer design, specifically in [29], and several of the results reviewed in this section have been presented in [29]. The current algorithm differs from that of [29] in its use of multiple values of $p$ in succession, leading to a sequence of instances of (8.3.3). Each instance is initialized using the solution for the previous value of $p$ in an effort to enhance the sparsity of the final solution. In terms of theoretical content, this section also includes a characterization of optimality for problem (8.3.3) that is more precise than the one in [29]. An earlier version of the content in this section has appeared in [112].

To further motivate the approximation of the 0-norm by the $p$-norm, we discuss the two-dimensional example shown in Fig. 8-1. The feasible region for (8.3.3) is polyhedral as noted in Section 8.1 and remains the same for all values of $p$. Consider first the case $p = 1$ in (8.3.3). An optimal solution can be determined graphically by constructing the smallest $\ell^1$ ball, which has a diamond shape, that intersects the feasible region. In this example, the minimum 1-norm solution occurs at a vertex that does not correspond to a sparse solution. Next consider the same minimization for $p < 1$. As $p$ decreases from 1, the boundaries of the $\ell^p$ ball curve inward and extend farther along the coordinate axes than they do elsewhere. Consequently, the minimum $p$-norm solutions tend toward the axes, and for $p$ sufficiently small, the solution converges to the true sparsest solution.

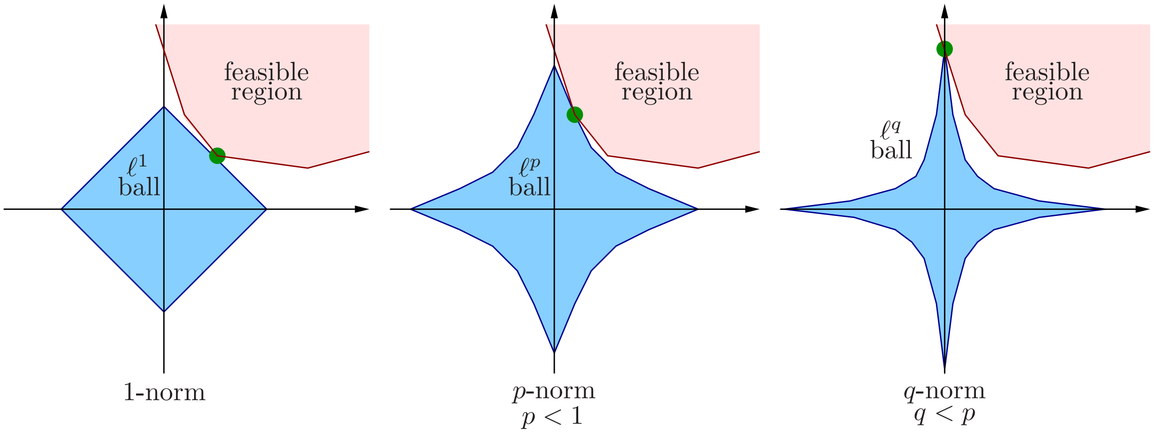


Figure 8-1: Graphical minimization of different $p$-norms. The optimal solutions are indicated by green circles.

The behavior seen in the preceding example can be formalized. It can be shown that an optimal solution to (8.3.3) is also an optimal solution to (8.1.5) for $p$ sufficiently small but finite [29, Thm. 4]. The convergence of optimal solutions is essentially due to the convergence of the $p$-norms in (8.3.2). The sufficiency of finite values of $p$ for convergence is due to the existence of optimal solutions of (8.3.3) at vertices of the polyhedral feasible set, which will be justified shortly, and the finiteness of the number of vertices. The upper bound on sufficient values for $p$ given in [29, Thm. 4] is impractical to compute however. The authors in [29] give little guidance regarding the choice of $p$ beyond reporting that reasonable values (say on the order of 0.1) can generate sparse solutions in practice.

A more significant issue with the $p$-norm approach is the non-convexity of the objective function in (8.3.3) for $p < 1$. As a consequence, (8.3.3) is difficult to solve when $p < 1$. To mitigate the lack of convexity, we propose solving a sequence of $p$-norm minimization problems as opposed to a single minimization, beginning with $p = 1$ and decreasing $p$ gradually thereafter toward zero. For $p = 1$, (8.3.3) is a convex problem and can be solved efficiently using linear programming to yield a global minimum. For $p$ slightly less than

1, one might expect that a minimum $p$-norm solution should be close in some sense to the minimum 1-norm solution already determined, and therefore the latter could be a promising initial solution toward obtaining the former. Generalizing this idea, for $q$ slightly less than $p$, a minimum $p$-norm solution can be used to initialize the minimization of the $q$-norm. It is conjectured that if the sequence of $p$ values decreases slowly enough, the sequence of solutions resulting from this initialization strategy will remain globally optimal for $p$ significantly below 1.

The above initialization strategy can be partially justified by examining the deviation from the optimal value when a minimum $p$-norm solution is evaluated in terms of the $q$-norm for $q < p$. Suppose that a minimum $p$-norm solution $\mathbf{b}^p$ has been determined, and let $\mathbf{b}^q$ denote a minimum $q$-norm solution that is desired. Then the following inequalities hold:

$$\|\mathbf{b}^p\|_p \leq \|\mathbf{b}^q\|_p \leq \|\mathbf{b}^q\|_q \leq \|\mathbf{b}^p\|_q \,. \tag{8.3.4}$$

The outer inequalities are due to the optimality of $\mathbf{b}^p$ and $\mathbf{b}^q$ under their respective norms, while the middle inequality results from the property that $\|\mathbf{b}\|_p$ is non-decreasing as $p$ decreases for fixed $\mathbf{b}$. By raising all quantities in (8.3.4) to the power $q$, we obtain

$$\|\mathbf{b}^p\|_p^q \leq \|\mathbf{b}^q\|_q^q \leq \|\mathbf{b}^p\|_q^q \,,$$

which implies that if $\mathbf{b}^p$ is used to initialize the minimization of $\|\mathbf{b}\|_q^q$, the initial optimality gap is no greater than $\|\mathbf{b}^p\|_q^q - \|\mathbf{b}^p\|_p^q$. Hence the optimality gap is small if $q$ is close to $p$. It should be emphasized however that closeness in objective value does not necessarily imply closeness of the solutions $\mathbf{b}^p$ and $\mathbf{b}^q$, and it is possible to construct two-dimensional examples in which $\mathbf{b}^p$ and $\mathbf{b}^q$ can be arbitrarily far apart. Furthermore, the bound in (8.3.4) relies on the global optimality of $\mathbf{b}^p$ in terms of the $p$-norm, which is difficult to ensure in practice if $p < 1$.

In the remainder of this section, the proposed algorithm based on sequential $p$-norm minimization is developed further. In Section 8.3.1, the problem of $p$-norm minimization for fixed $p$ is analyzed and a necessary condition of optimality is derived for the case $p < 1$. Based on the optimality condition, an algorithm for $p$-norm minimization is developed In Section 8.3.2 along with further details of the overall sequential procedure.

### 8.3.1 Analysis of $p$-norm minimization

We now focus on problem (8.3.3) for a fixed value of $p \in (0, 1]$, which is a recurring subproblem in the proposed method. Problem (8.3.3) is recast into an equivalent form by using (3.3.5) to express each coefficient $b_n$ in terms of its positive and negative parts $b_n^+$ and $b_n^-$. Under the condition that at most one of $b_n^+$, $b_n^-$ is non-zero, i.e., $b_n^+ b_n^- = 0$, we also have

$$|b_n| = b_n^+ + b_n^- \tag{8.3.5}$$

for all $n$. Using (3.3.5), (8.3.5), and (8.3.1), problem (8.3.3) can be transformed into

$$\begin{aligned} \min_{\mathbf{b}^+, \mathbf{b}^-} \quad & F(\mathbf{b}^+, \mathbf{b}^-) \\ \text{s.t.} \quad & -\delta_d \mathbf{e} \leq \mathbf{A}(\mathbf{b}^+ - \mathbf{b}^-) - \mathbf{d} \leq \delta_d \mathbf{e}, \\ & \mathbf{b}^+ \geq \mathbf{0}, \quad \mathbf{b}^- \geq \mathbf{0}, \end{aligned} \tag{8.3.6}$$

where the objective function $F(\mathbf{b}^+, \mathbf{b}^-)$ is defined as

$$F(\mathbf{b}^+, \mathbf{b}^-) = \sum_{n=0}^{N-1} \left(b_n^+ + b_n^-\right)^p. \tag{8.3.7}$$

Problems (8.3.6) and (8.3.3) are equivalent in the sense of having the same optimal value and a one-to-one correspondence between optimal solutions. The nonlinear constraints $b_n^+ b_n^- = 0$, $n = 0, \ldots, N-1$, do not have to be included in (8.3.6) because they are automatically satisfied by all optimal solutions. The justification for this property is similar to the one given in Section 3.3 in the paragraph following (3.3.6). As a consequence, the feasible set for (8.3.6) is also a polytope, which we will denote as $\mathcal{P}$ for convenience.

When $p = 1$, (8.3.6) is a linear programming problem and can be solved using standard techniques [96]. We focus therefore on the case $p < 1$. It can be verified that for $p < 1$, the functions $(b_n^+ + b_n^-)^p$ are concave. We may observe for instance that the Hessian of $(b_n^+ + b_n^-)^p$ is negative semidefinite except at $b_n^+ = b_n^- = 0$ where it does not exist but the function is still continuous. It follows that $F(\mathbf{b}^+, \mathbf{b}^-)$ is a concave function and can be shown to attain a minimum at a vertex of $\mathcal{P}$ [98, Prop. B.20] [29]. The location of optimal solutions at vertices forms the basis for a simplex-like algorithm for solving (8.3.6) in the case $p < 1$. This algorithm is described in Section 8.3.2.

In the remainder of the current subsection, the vertex condition of optimality for (8.3.6) is refined and the result is interpreted geometrically in the context of Fig. 8-1. To state the result, we introduce some additional definitions. Given a local minimum $(\mathbf{b}^{+*}, \mathbf{b}^{-*})$ of (8.3.6), define $\mathcal{Y}$ and $\mathcal{Z}$ to be the sets of indices $n$ such that $b_n^{+*} + b_n^{-*} > 0$ and $b_n^{+*} = b_n^{-*} = 0$ respectively, i.e., the index sets corresponding to non-zero and zero coefficients as before. Also define $\mathcal{P}_{\mathcal{Y}}$ to be the restriction of $\mathcal{P}$ to the hyperplane defined by $b_n^+ = b_n^- = 0$ for $n \in \mathcal{Z}$, i.e.,

$$\mathcal{P}_{\mathcal{Y}} = \left\{ (\mathbf{b}_{\mathcal{Y}}^+, \mathbf{b}_{\mathcal{Y}}^-) : (\mathbf{b}^+, \mathbf{b}^-) \in \mathcal{P};\ b_n^+ = b_n^- = 0,\ n \in \mathcal{Z} \right\}.$$

The following is a further characterization of optimality for (8.3.6):

**Theorem 9.** *If $(\mathbf{b}^{+*}, \mathbf{b}^{-*})$ is a local minimum of problem (8.3.6) with $0 < p < 1$, then*

$$\nabla F\left(\mathbf{b}_{\mathcal{Y}}^{+*}, \mathbf{b}_{\mathcal{Y}}^{-*}\right)^T \begin{bmatrix} \mathbf{b}_{\mathcal{Y}}^{+} - \mathbf{b}_{\mathcal{Y}}^{+*} \\ \mathbf{b}_{\mathcal{Y}}^{-} - \mathbf{b}_{\mathcal{Y}}^{-*} \end{bmatrix} > 0 \quad \forall\, (\mathbf{b}_{\mathcal{Y}}^+, \mathbf{b}_{\mathcal{Y}}^-) \in \mathcal{P}_{\mathcal{Y}},\ (\mathbf{b}_{\mathcal{Y}}^+, \mathbf{b}_{\mathcal{Y}}^-) \neq (\mathbf{b}_{\mathcal{Y}}^{+*}, \mathbf{b}_{\mathcal{Y}}^{-*}), \qquad (8.3.8)$$

*which implies that $(\mathbf{b}_{\mathcal{Y}}^{+*}, \mathbf{b}_{\mathcal{Y}}^{-*})$ is a vertex of $\mathcal{P}_{\mathcal{Y}}$.*

The proof of Theorem 9 is given in Appendix E.2. The result can be regarded as a generalization of the usual condition of optimality (see e.g. [98]),

$$\nabla F\left(\mathbf{b}^{+*}, \mathbf{b}^{-*}\right)^T \begin{bmatrix} \mathbf{b}^{+} - \mathbf{b}^{+*} \\ \mathbf{b}^{-} - \mathbf{b}^{-*} \end{bmatrix} \geq 0 \quad \forall\, (\mathbf{b}^+, \mathbf{b}^-) \in \mathcal{P}. \qquad (8.3.9)$$

Condition (8.3.9) may not apply at a local minimum of (8.3.6) because the gradient $\nabla F(\mathbf{b}^+, \mathbf{b}^-)$ is not defined at points where $b_n^+ = b_n^- = 0$ for some $n$. Theorem 9 shows that a strict version of (8.3.9) does hold over the space of non-zero coefficients.

Theorem 9 may be interpreted geometrically using the two-dimensional example of Fig. 8-1. According to the theorem, if an optimal solution $\mathbf{b}^*$ has no zero-valued components, i.e., $\mathcal{Y} = \{0, \ldots, N-1\}$, then it must occur at a vertex of the polytope $\mathcal{P}$. This property can be seen in the left and centre panels of Fig. 8-1. If some of the components of $\mathbf{b}^*$ are zero, the vector formed from the non-zero components must be a vertex of a restriction of $\mathcal{P}$. This property is illustrated in the right panel of Fig. 8-1, in which the restriction of the polyhedron is its intersection with the vertical axis and the optimal solution occurs at one extremity of the restriction.

### 8.3.2 Algorithm for sequential $p$-norm minimization

The overall algorithm combines the sequential procedure outlined at the beginning of Section 8.3 with an algorithm for $p$-norm minimization based on the vertex optimality condition given in Section 8.3.1. For concreteness, we assume that $p$ decreases according to

$$p^{(i+1)} = \alpha p^{(i)}, \quad p^{(0)} = 1,$$

where $i$ is an index for the subproblems and $\alpha$ is slightly less than 1. The first instance of (8.3.6) with $p = p^{(0)} = 1$ is a linear programming problem and can be solved using any standard solver.[1] Each subsequent subproblem is initialized with the final solution to the previous subproblem. The process terminates when $p$ has decreased to an acceptably small value $p_{\min}$ or when the solution is deemed to have converged.

To solve (8.3.6) when $p < 1$, we propose a local search algorithm in which the search is restricted to the vertices of the feasible polyhedron $\mathcal{P}$. In each iteration, we begin at a vertex solution and search all adjacent vertices for lower values of the objective function $F(\mathbf{x})$. If none of the adjacent vertices have lower objective values, the algorithm terminates. Otherwise the algorithm moves to the vertex with the lowest value and the search continues.

The local search algorithm is similar to the simplex method for linear programming in that it searches for lower function values by moving from one vertex to another along edges of the polyhedron. As a consequence, the algebraic characterization of vertices and the procedure for moving between them are the same as in the simplex method. For completeness, a summary is given of the computations involved in moving between adjacent vertices. The reader is referred to linear programming texts (e.g. [96]) for a more complete treatment.

For convenience and in keeping with convention, the constraints in (8.3.6) are converted

[1] To facilitate the initialization of the next subproblem, the linear program solver should return a vertex solution, which is guaranteed to exist.

to standard form, yielding

$$\begin{aligned}
\min_{\mathbf{b}^+,\mathbf{b}^-,\mathbf{r},\mathbf{s}} \quad & F(\mathbf{b}^+,\mathbf{b}^-) \\
\text{s.t.} \quad & \underbrace{\begin{bmatrix} \mathbf{A} & -\mathbf{A} & \mathbf{I} & \mathbf{0} \\ -\mathbf{A} & \mathbf{A} & \mathbf{0} & \mathbf{I} \end{bmatrix}}_{\mathbf{C}} \underbrace{\begin{bmatrix} \mathbf{b}^+ \\ \mathbf{b}^- \\ \mathbf{r} \\ \mathbf{s} \end{bmatrix}}_{\mathbf{z}} = \underbrace{\begin{bmatrix} \delta_d\mathbf{e}+\mathbf{d} \\ \delta_d\mathbf{e}-\mathbf{d} \end{bmatrix}}_{\mathbf{y}}, \\
& \mathbf{z} \geq \mathbf{0},
\end{aligned} \tag{8.3.10}$$

where $\mathbf{r}$ and $\mathbf{s}$ are non-negative slack variables. In standard form, each vertex is associated with a set $\mathcal{B}$ of $2K$ basic indices $\mathcal{B}(1),\ldots,\mathcal{B}(2K)$, with the property that the square matrix $\mathbf{C}_{\mathcal{B}}$, composed of the columns of $\mathbf{C}$ indexed by $\mathcal{B}$, is invertible. Denoting the corresponding vector of basic variables $\left(z_{\mathcal{B}(1)},\ldots,z_{\mathcal{B}(2K)}\right)$ by $\mathbf{z}_{\mathcal{B}}$, we have $\mathbf{z}_{\mathcal{B}} = \mathbf{C}_{\mathcal{B}}^{-1}\mathbf{y}$ and $z_n = 0$ for all non-basic indices $n$, i.e., those not in $\mathcal{B}$.

A move from a given vertex $\mathbf{z}$ to an adjacent vertex is accomplished by increasing the value of a non-basic variable, say with index $m$, and adjusting the values of the basic variables to preserve the equality constraints in (8.3.10). The neighboring vertex is reached when one of the basic variables becomes zero. The new vertex $\mathbf{z}'$ is given by

$$\mathbf{z}'_{\mathcal{B}} = \mathbf{z}_{\mathcal{B}} - \theta\Delta\mathbf{z}, \qquad z_m = \theta, \qquad z_n = 0 \quad \forall\, n \notin \mathcal{B},\; n \neq m,$$

where $\Delta\mathbf{z} = \mathbf{C}_{\mathcal{B}}^{-1}\mathbf{C}_m$ and

$$\theta = \min_{k:\Delta z_k > 0} \frac{z_{\mathcal{B}(k)}}{\Delta z_k}.$$

The objective function $F(\mathbf{b}^+,\mathbf{b}^-)$ may be evaluated at the new vertex by substituting those components of $\mathbf{z}'$ corresponding to the vectors $\mathbf{b}^+$ and $\mathbf{b}^-$ into (8.3.7). If the new vertex has a lower objective value and is chosen as the next iterate, the set of basic indices must also be updated. The basic variable that has been reduced to zero leaves the basis and is replaced by the previously non-basic variable with index $m$. The index of the exiting basic variable is

$$l = \arg\min_{k:\Delta z_k > 0} \frac{z_{\mathcal{B}(k)}}{\Delta z_k}$$

and thus the new basis $\mathcal{B}'$ is given by $\mathcal{B}'(l) = m$ and $\mathcal{B}'(k) = \mathcal{B}(k)$ for $k \neq l$.

Not every non-basic variable corresponds to an adjacent vertex. In particular, if $z_n$ is basic and $z_{n+N}$ is non-basic for $n \in \{0, 1, \ldots, N-1\}$ (i.e., a $(b_n^+, b_n^-)$ pair), then applying the foregoing procedure with $m = n + N$ yields a direction in which both $z_n$ and $z_{n+N}$ are increased by the same amount and all other variables are unchanged. This direction does not lead to another vertex and also results in an unbounded increase in $F(\mathbf{b}^+, \mathbf{b}^-)$, and therefore does not need to be considered. The case where $z_{n+N}$ is basic and $z_n$ is non-basic is similar. In addition, a change of basis may not always result in a change of vertex because of degeneracy, which we do not discuss here.

The local search algorithm may be made more efficient and numerically stable by exploiting the structure of the matrix $\mathbf{C}$ when inverting the matrix $\mathbf{C}_{\mathcal{B}}$. Since $N \ll K$, most of the columns of $\mathbf{C}_{\mathcal{B}}$ are columns of a $2K \times 2K$ identity matrix. It can be shown that he rows of $\mathbf{C}_{\mathcal{B}}$ can be reordered to form the matrix

$$\widetilde{\mathbf{C}}_{\mathcal{B}} = \begin{bmatrix} \mathbf{C}_{11} & \mathbf{0} \\ \mathbf{C}_{21} & \mathbf{I} \end{bmatrix},$$

where $\mathbf{C_{11}}$ is square and invertible. The original system of equations corresponding to $\mathbf{C}_{\mathcal{B}}$ may now be solved by first solving the system corresponding to $\mathbf{C}_{11}$ and then substituting the result into the equations specified by the second row of $\widetilde{\mathbf{C}}_{\mathcal{B}}$. Since the dimension of $\mathbf{C}_{11}$ is never greater than $N \times N$, this alternative procedure is considerably more efficient than the direct solution of a $2K \times 2K$ system of equations.

In terms of overall complexity, the sequential $p$-norm algorithm is less complex than the minimum-increase successive thinning algorithm discussed in Section 8.2, and can be less complex than the smallest-coefficient successive thinning algorithm as well depending on the value of $N$. The complexity of the $p$-norm algorithm is equivalent to a fixed number of linear programs. More precisely, the equivalent number of linear programs depends on the number of values of $p$ used, but does not depend on the dimension $N$. In contrast, the number of linear programs solved in the smallest-coefficient successive thinning algorithm is linear in $N$.

In our experience with the algorithm, the number of non-zero coefficients decreases more rapidly when $p$ is near 1 and less rapidly as $p$ decreases. Around $p = 0.1$, the algorithm often converges to a solution that appears to be locally minimal for all smaller values of $p$ since further local searches do not generate new iterates. To determine whether additional coefficients can be set to zero after convergence, the successive thinning algorithm of Section 8.2 (e..g. using the smallest-coefficient rule for simplicity) can be run starting from the final subset of non-zero coefficients resulting from $p$-norm minimization. This re-optimization is occasionally able to generate one or two additional zero-valued coefficients after the $p$-norm algorithm converges. In addition, the re-optimized design almost always satisfies the frequency response constraints with non-zero margin, a benefit of the successive thinning algorithm discussed in Section 8.2.

## 8.4 Branch-and-bound

In Sections 8.4–8.7, we explore at a preliminary level a branch-and-bound approach to solving problem (8.1.5) exactly. Several of the techniques of Chapter 3 can be extended to the Chebyshev error criterion considered in the current chapter. The computational complexity however is significantly higher than before, albeit still polynomial except for the diagonal relaxation as discussed in Section 8.7. Overcoming the increased complexity and developing a full branch-and-bound algorithm for problem (8.1.5) is a potential subject for future work.

We begin in this section by reformulating (8.1.5) to facilitate the application of branch-and-bound. As in Section 3.1, binary variables $i_n$ are used to indicate whether or not the corresponding coefficient $b_n$ is non-zero. Thus (8.1.5) becomes

$$\begin{aligned} \min_{\mathbf{b},\mathbf{i}} \quad & \sum_{n=0}^{N-1} i_n \\ \text{s.t.} \quad & -\delta_d \mathbf{e} \le \mathbf{A}\mathbf{b} - \mathbf{d} \le \delta_d \mathbf{e}, \\ & |b_n| \le B_n i_n \quad \forall\, n, \\ & i_n \in \{0,1\} \quad \forall\, n, \end{aligned} \tag{8.4.1}$$

in exact analogy with (3.1.1). The constants $B_n$ must be chosen to be sufficiently large as before; specific values are given in Section 8.6.

The mixed-integer optimization problem in (8.4.1) is solved by subdividing it into a tree of subproblems as depicted in Fig. 3-1. In keeping with previous notation, we denote by $\mathcal{Z}$ and $\mathcal{U}$ the sets of indices such that $i_n = 0$ and $i_n = 1$ respectively; $\mathcal{F}$ denotes the set corresponding to the indicator variables that remain free. One difference from the quadratically constrained case in Chapter 3 is that it is no longer possible to eliminate the coefficients in $\mathcal{U}$ from the subproblems. The same issue was encountered in Section 8.2.1 and appears again in Section 8.5.

Recalling the discussion in Section 3.1, the efficiency of the branch-and-bound procedure is highly dependent on the quality of initial feasible solutions to (8.1.5) as well as the quality of lower bounds on the optimal values of the subproblems. As seen in Section 8.8, both the successive thinning and $p$-norm minimization algorithms are capable of producing sparse solutions, which are optimal at least in some cases, and hence either algorithm is well-suited to providing initial solutions for a branch-and-bound algorithm.

As for lower bounds, most of the techniques in Chapter 3 can be extended to yield bounds for problem (8.1.5). In Section 8.5, we extend the methods of Section 3.2, while the same is done in Section 8.6 for the method of linear relaxation. It will be seen that the bounds in Section 8.5 require the solution of $\mathcal{O}(N)$ linear programs as opposed to the evaluation of closed-form expressions such as (2.3.3) in Chapter 3. The determination of a maximally tight linear relaxation in Section 8.6 also can no longer be done in closed form and requires $\mathcal{O}(N)$ linear programs as well. Furthermore, while the complexity of evaluating the bounds in Sections 8.5 and 8.6 remains polynomial in $N$, this is not true for the extension of the diagonal relaxation discussed in Section 8.7. Due to the significant increases in complexity, a complete branch-and-bound algorithm for (8.1.5) is not developed in this thesis. Possible steps in this direction are outlined in Chapter 9.

## 8.5 Low-complexity lower bounds

In this section, the bounding methods of Section 3.2 are applied to problem (8.1.5) and its subproblems. The first method involves identifying coefficients in the subset $\mathcal{F}$ for which a zero value is no longer feasible. As before, the indicator variables for these coefficients can be set to 1, i.e., the corresponding indices are moved from $\mathcal{F}$ to $\mathcal{U}$. The cardinality of $\mathcal{U}$, which is always a lower bound on the optimal value, increases as a result.

One way to determine whether individual coefficients can be feasibly set to zero is to solve problem (8.2.1) for each $\mathcal{Y}$ of the form $\mathcal{Y} = \mathcal{U} \cup \mathcal{F} \backslash n$, $n \in \mathcal{F}$. If the optimal value of (8.2.1) for $\mathcal{Y} = \mathcal{U} \cup \mathcal{F} \backslash n$ is greater than the tolerance $\delta_d$, then $b_n = 0$ is not a feasible value since the minimal error with $b_n = 0$ exceeds the tolerance. Checking every coefficient in $\mathcal{F}$ requires the solution of $|\mathcal{F}|$ linear programs. An alternative method is to determine the minimum and maximum feasible values for each coefficient, specifically by solving the following pair of linear programs for each $n \in \mathcal{F}$:

$$\min_{\mathbf{b}_{\mathcal{Y}}} \quad b_n \qquad \text{s.t.} \qquad -\delta_d \mathbf{e} \leq \mathbf{A}_{\mathcal{Y}} \mathbf{b}_{\mathcal{Y}} - \mathbf{d} \leq \delta_d \mathbf{e}, \tag{8.5.1a}$$

$$\max_{\mathbf{b}_{\mathcal{Y}}} \quad b_n \qquad \text{s.t.} \qquad -\delta_d \mathbf{e} \leq \mathbf{A}_{\mathcal{Y}} \mathbf{b}_{\mathcal{Y}} - \mathbf{d} \leq \delta_d \mathbf{e}, \tag{8.5.1b}$$

where $\mathcal{Y} = \mathcal{U} \cup \mathcal{F}$. The minimum and maximum values in (8.5.1) define the interval of feasible values and it suffices to check whether zero belongs to the interval. The second method is less efficient than the first because it requires the solution of $2\,|\mathcal{F}|$ linear programs. However, as discussed later in Section 8.6, the optimal values of (8.5.1) are also used to obtain the tightest possible linear relaxation of (8.1.5).

The form of the linear programs (8.2.1) and (8.5.1) reflects the fact that the variables $b_n$ for $n \in \mathcal{U}$ cannot be eliminated as in Chapter 3. As discussed in Section 8.2.1, the variables $b_n$, $n \in \mathcal{U}$ must remain in (8.2.1), (8.5.1), and other optimization problems because an analytical expression for their optimal values is not available. Thus the dimensionality reduction comes only from the zero-value constraints $b_n = 0$ for $n \in \mathcal{Z}$.

As noted in Section 3.2, the above tests are only necessary for subproblems generated from a parent by fixing an indicator variable to zero since fixing an indicator variable to one does not change the set of feasible $\mathbf{b}$. In addition, the tests can be generalized to subsets larger than a single coefficient with a corresponding increase in computation.

It can be shown that the optimal values of (8.5.1) are finite as long as the matrix $\mathbf{A}$ has full rank. A slight generalization of this result verifies the earlier claim in Section 8.1 that the polytope specified by the constraints in (8.1.5) is bounded if $\mathbf{A}$ has full rank. To see this, consider the linear programming dual of (8.5.1a), given as follows:

$$\begin{aligned} \max_{\mathbf{p}^+,\mathbf{p}^-} \quad & \mathbf{d}^T(\mathbf{p}^+ - \mathbf{p}^-) - \delta_d \mathbf{e}^T(\mathbf{p}^+ + \mathbf{p}^-) \\ \text{s.t.} \quad & \mathbf{A}_{\mathcal{Y}}^T(\mathbf{p}^+ - \mathbf{p}^-) = \mathbf{e}_n, \\ & \mathbf{p}^+ \geq \mathbf{0}, \quad \mathbf{p}^- \geq \mathbf{0}, \end{aligned} \tag{8.5.2}$$

where $\mathbf{e}_n$ is the $n$th standard basis vector. If $\mathbf{A}$ has full rank, then so too does $\mathbf{A}_{\mathcal{Y}}$, and this combined with the fact that the combination $\mathbf{p}^+ - \mathbf{p}^-$ can generate any vector in $\mathbb{R}^K$ implies that (8.5.2) always has a feasible solution. By linear programing duality, the optimal value of (8.5.1a) is guaranteed to be finite [96]. A similar result holds for (8.5.1b) and indeed for any linear function of $\mathbf{b}$ optimized over the same feasible set, and hence the feasible set is bounded.

The second bounding method discussed in Section 3.2 involves determining whether feasible solutions with small numbers of non-zero coefficients exist. This second method again requires linear programming, unlike in Section 3.2. To determine whether the minimal subset $\mathcal{Y} = \mathcal{U}$ of non-zero coefficients is feasible, we may solve (8.2.1) with $\mathcal{Y} = \mathcal{U}$. Similarly, determining the feasibility of $\mathcal{Y} = \mathcal{U} \cup \{n\}$ for all $n \in \mathcal{F}$ requires $|\mathcal{F}|$ linear programs.

In summary, while the methods discussed in this section were computationally simple to implement in the quadratically constrained case, they now require substantial linear optimizations in the present case. In Chapter 9, a possible approach to improving the computational efficiency is suggested.

## 8.6 Linear relaxation

In this section, we apply the method of linear relaxation to problem (8.1.5), and more specifically to its mixed-integer formulation in (8.4.1). Similar to the derivation in Section 3.3.1, the relaxation of the binary constraints on the indicator variables $i_n$ in (8.4.1) to unit-interval constraints allows $i_n$ to be eliminated from the optimization. The resulting problem is

$$\begin{aligned} \min_{\mathbf{b}} \quad & \sum_{n=0}^{N-1} \frac{|b_n|}{B_n} \\ \text{s.t.} \quad & -\delta_d \mathbf{e} \leq \mathbf{A}\mathbf{b} - \mathbf{d} \leq \delta_d \mathbf{e}, \end{aligned}$$

which is analogous to (3.3.1). Thus linear relaxation again yields a weighted $\ell^1$ minimization problem whose optimal value is a lower bound on the optimal value of (8.4.1). For a general subproblem defined by the subsets $\mathcal{U}$ and $\mathcal{F}$, a linear relaxation takes the form

$$\begin{aligned} \min_{\mathbf{b}_{\mathcal{Y}}} \quad & |\mathcal{U}| + \sum_{n \in \mathcal{F}} \frac{|b_n|}{B_n} \\ \text{s.t.} \quad & -\delta_d \mathbf{e} \leq \mathbf{A}_{\mathcal{Y}} \mathbf{b}_{\mathcal{Y}} - \mathbf{d} \leq \delta_d \mathbf{e}, \end{aligned} \tag{8.6.1}$$

where $\mathcal{Y} = \mathcal{U} \cup \mathcal{F}$. It is seen that the optimal value of (8.6.1) is in general a better lower bound than the bound of $|\mathcal{U}|$ in Section 8.5.

As in Section 3.3.1, the lower bound resulting from linear relaxation can be tightened by representing each coefficient as the difference between its positive and negative parts. Following the same development as before leads to a relaxation that is analogous to (3.3.8):

$$\begin{aligned} \min_{\mathbf{b}_{\mathcal{U}}, \mathbf{b}_{\mathcal{F}}^{+}, \mathbf{b}_{\mathcal{F}}^{-}} \quad & |\mathcal{U}| + \sum_{n \in \mathcal{F}} \left( \frac{b_n^{+}}{B_n^{+}} + \frac{b_n^{-}}{B_n^{-}} \right) \\ \text{s.t.} \quad & -\delta_d \mathbf{e} \leq \mathbf{A}_{\mathcal{U}} \mathbf{b}_{\mathcal{U}} + \mathbf{A}_{\mathcal{F}} (\mathbf{b}_{\mathcal{F}}^{+} - \mathbf{b}_{\mathcal{F}}^{-}) - \mathbf{d} \leq \delta_d \mathbf{e}, \\ & \mathbf{b}_{\mathcal{F}}^{+} \geq \mathbf{0}, \quad \mathbf{b}_{\mathcal{F}}^{-} \geq \mathbf{0}. \end{aligned} \tag{8.6.2}$$

The split into positive and negative parts is done only for the coefficients in $\mathcal{F}$ since they are responsible for the non-constant contribution to the lower bound. The coefficients in $\mathcal{U}$ remain as variables in (8.6.2), continuing the pattern seen earlier in this chapter. To obtain the tightest possible lower bound, the constants $B_n^{\pm}$ should be made as small as possible, but must also be large enough to not impose further constraints on $\mathbf{b}^{+}$ and $\mathbf{b}^{-}$ beyond the original linear constraints in (8.1.5). It follows that the best choices for $B_n^{+}$ and $B_n^{-}$ are given by

$$\begin{aligned} B_n^{+} &= \max_{\mathbf{b}_{\mathcal{Y}}} \quad b_n \quad \text{s.t.} \quad -\delta_d \mathbf{e} \leq \mathbf{A}_{\mathcal{Y}} \mathbf{b}_{\mathcal{Y}} - \mathbf{d} \leq \delta_d \mathbf{e}, \\ B_n^{-} &= \max_{\mathbf{b}_{\mathcal{Y}}} \quad -b_n \quad \text{s.t.} \quad -\delta_d \mathbf{e} \leq \mathbf{A}_{\mathcal{Y}} \mathbf{b}_{\mathcal{Y}} - \mathbf{d} \leq \delta_d \mathbf{e}, \end{aligned}$$

which involve the same optimization problems as in (8.5.1) except for a sign change because of the definition of $b_n^{-}$ as a non-negative variable.

In Section 8.8, the lower bound of $|\mathcal{U}|$ and the lower bound resulting from (8.6.2) are computed for several filter design examples. It will be seen that while the value of $|\mathcal{U}|$ can account for a significant fraction of the non-zero coefficients in a solution, the additional contribution due to linear relaxation tends to be small. The examples in Section 8.8 motivate the development of relaxations that are better approximations to the original problem. As discussed in the next section, strong diagonal relaxations do not appear to be tractable for problem (8.1.5). Alternative relaxations are mentioned briefly in Chapter 9.

## 8.7 Diagonal relaxation

We now consider the possibility of applying diagonal relaxation to problem (8.1.5). We first observe that the direct substitution of a diagonal matrix for $\mathbf{A}$ in (8.1.5) leads to the lower bound of $|\mathcal{U}|$ discussed previously in Section 8.5. With $\mathbf{A}$ replaced by a diagonal matrix, the constraints in (8.1.5) become upper and lower bounds on individual components of $\mathbf{b}$, i.e., a box constraint. To obtain a lower bound on the optimal value of (8.1.5), this box should be chosen to enclose the original feasible set. It can be seen that minimizing the zero-norm over the smallest enclosing box yields the same result, namely the cardinality of $\mathcal{U}$, as counting the number of coefficients for which a zero value is infeasible.

A second type of diagonal relaxation involves replacing the constraint in (8.1.5) with the diagonal quadratic constraint in (3.4.1), i.e., substituting a coordinate-aligned ellipsoidal feasible set for the original polyhedral set. As with the box approximation above, to derive a lower bound an ellipsoid that encloses the original polyhedron is desired. However, for a polyhedron described by a set of linear inequalities as in (8.1.5), obtaining an enclosing ellipsoid that is also a reasonably tight approximation is a difficult computational problem. A natural choice might be an enclosing ellipsoid of minimal volume, but unfortunately there are no efficient algorithms for determining a minimum-volume enclosing ellipsoid for a polyhedron specified by linear inequalities, and indeed the problem is thought to be NP-hard [113]. An alternative method for obtaining an enclosing ellipsoid is to first determine the ellipsoid of maximal volume that can be inscribed in the polyhedron, which is known to have polynomial complexity [113, 114] and can be done efficiently in practice [4, 115]. Dilating the maximum-volume inscribed ellipsoid by a factor of $N$ is guaranteed to yield an enclosing ellipsoid [113]. However, this dilation is unlikely to result in an ellipsoid that is a good approximation to the polyhedron. Given the lack of appropriate algorithms, we do not consider diagonal relaxations of (8.1.5) any further in this thesis.

## 8.8 Design examples

In this section, we present a number of design examples to illustrate the performance of the algorithms in Sections 8.2 and 8.3, and also of the lower bounds in Sections 8.5 and 8.6. The first three examples in Sections 8.8.1–8.8.3 explore the dependence of the level of sparsity on the characteristics of the desired frequency response. We consider angle-selective beamformers, bandpass filters, and an acoustic equalizer. For frequency-selective filters in particular, the results suggest that the relative decrease in the number of non-zero coefficients is smaller for higher stopband attenuations. For bandpass filters, we observe that the sparsity does not seem to vary much with the passband center frequency except at certain special values.

In Sections 8.8.4 and 8.8.5, we compare the algorithms of this chapter to a commercial integer programming solver used in [3] and to a heuristic algorithm described in [1]. The comparison with integer programming shows that our algorithms are capable of producing optimally sparse solutions. The comparison with [1] shows that our algorithms are somewhat less adept at automatically discovering $n$th-band structure, but perform better on a more generic example. Throughout this section, it is seen that the algorithms presented in Sections 8.2–8.3 perform very similarly. In a few instances, the $p$-norm algorithm and the smallest-coefficient rule give slightly worse results than the minimum-increase rule, but the first two algorithms also have lower complexity. The examples also suggest that the lower bounds in Sections 8.5–8.6 are not very tight.

### 8.8.1 Angle-selective beamformer

As is well known, the design of uniform linear beamformers is mathematically identical to the design of discrete-time FIR filters [116]. For a length $N$ linear array with uniform spacing $d$, the beam pattern at a wavelength $\lambda$ is given by

$$B(\theta, \lambda) = \sum_{n=0}^{2N-2} w_n e^{jn\left[\frac{2\pi d}{\lambda}\cos\theta\right]}, \quad 0 \leq \theta \leq \pi, \tag{8.8.1}$$

where $\theta$ is the angle from the array axis. Equation (8.8.1) has the form of a discrete-time Fourier transform of the array weights $w_n$ with

$$\psi = \frac{2\pi d}{\lambda}\cos\theta \tag{8.8.2}$$

playing the role of the frequency variable. The objective is to choose weights to approximate a desired beam pattern. When the magnitude of the desired beam pattern is symmetric about $\psi = 0$, it is typical to restrict the weights to be real and even-symmetric, in which case the problem is equivalent to linear-phase filter design. Such symmetry occurs when the beam is directed normal to the array (broadside) with no nulls required at specific angles. Moreover, beam patterns steered in other directions are frequently obtained by first designing a symmetric broadside beam pattern and then modulating the corresponding weights by an appropriate complex exponential. In this experiment, the beamformer is restricted to have an odd number of elements for simplicity, i.e., only Type I linear phase is considered.

The desired beam pattern chosen for this example has a mainlobe response that is equal to unity over a range of angles as opposed to a single angle. The specifications for the desired beam pattern (assumed to be symmetric) are listed in Table 8.1. In the case $d = \lambda/2$, the width of the mainlobe region is 5° at broadside. Beam patterns with a relatively wide and flat mainlobe find use in a number of contexts, which are sometimes grouped under the label *robust beamforming* [117]. The mainlobe shape is motivated by the presence of uncertainty in the direction of interest.

Table 8.1: Specifications for the beamformer example

| | |
|---|---|
| mainlobe region | $0 \leq \psi \leq \psi_p = 0.0436\pi$ |
| sidelobe region | $\psi_s = 0.0872\pi \leq \psi \leq \pi$ |
| mainlobe magnitude | within ±0.5 dB of unity |
| sidelobe magnitude | below $-20$, $-30$, $-40$ dB |

We consider sidelobe levels of $-20$, $-30$, and $-40$ dB. For each sidelobe level, array weights are designed using the successive thinning algorithms in Section 8.2 under both the minimum-increase and smallest-coefficient rules, the $p$-norm algorithm in Section 8.3, and the Parks-McClellan algorithm for comparison. For the sparse design algorithms, we allow up to 50% more length than that required by the Parks-McClellan design. With an optimal algorithm, sparsity is maximized by fixing the length $N$ to the maximum allowable value as mentioned in Section 2.1 since any solution of shorter length is feasible under the maximum length. With heuristic algorithms however, the sparsest solution is not necessarily attained at the maximum value of $N$ since optimality is not guaranteed. Hence we typically try all values of $N$ between the Parks-McClellan length and the maximum length. Note that the length of the final design is determined by the positions of the non-zero weights.

Table 8.2 lists the number of non-zero weights (corresponding to the number of required physical array elements) and the array length returned by the algorithms for each sidelobe level. For the sparse design algorithms, the decreases in the number of non-zero weights relative to the Parks-McClellan designs range from 15% to 33%, with the largest relative decreases at a sidelobe level of $-20$ dB. Thus the greatest gains in sparsity appear to occur at the least stringent sidelobe level. The amount of extra length used in the sparse designs is not more than 5% of the Parks-McClellan length and can actually be zero as in the $-30$ dB case.

Table 8.2: Numbers of non-zero weights and array lengths for different sidelobe levels

| sidelobe level [dB] | algorithm | non-zero weights | array length (in units of $d$) |
|---|---|---|---|
| $-20$ | Parks-McClellan | 43 | 42 |
| | minimum-increase | 29 | 44 |
| | smallest-coefficient | 29 | 44 |
| | $p$-norm | 29 | 44 |
| | feasible intervals | 7 | – |
| | linear relaxation | 17 | – |
| $-30$ | Parks-McClellan | 55 | 54 |
| | minimum-increase | 47 | 54 |
| | smallest-coefficient | 47 | 54 |
| | $p$-norm | 47 | 54 |
| | feasible intervals | 23 | – |
| | linear relaxation | 33 | – |
| $-40$ | Parks-McClellan | 79 | 78 |
| | minimum-increase | 65 | 80 |
| | smallest-coefficient | 65 | 82 |
| | $p$-norm | 67 | 78 |
| | feasible intervals | 35 | – |
| | linear relaxation | 45 | – |

Table 8.2 also shows the lower bound based on the feasible interval for each coefficient (see Section 8.5) as well as the lower bound resulting from the linear relaxation. These lower bounds are computed assuming the largest allowable value of $N$, i.e., 1.5 times the Parks-McClellan value. At the higher attenuation levels, the bound based on feasible intervals represents a significant fraction of the non-zero weights in the sparse solutions. The linear relaxation however falls well short of closing the remaining gap.

In a related experiment, we fix the number of non-zero weights at the value required by the Parks-McClellan algorithm (43, 55 and 79 for the different sidelobe levels) and determine how much additional sidelobe attenuation can be achieved using the sparse design methods, specifically by increasing the sidelobe attenuation in 0.1 dB increments until a design with more than the desired number of non-zeros is obtained. Here we also allow up to 50% more length than required for the Parks-McClellan design. Due to their approximate nature, it is possible that the sparse design algorithms may return a feasible design at attenuation levels beyond the point where an infeasible design is first encountered. Hence our results are conservative estimates of the potential improvement in attenuation. For the lower bounding methods, we continue to increase the sidelobe attenuation until the lower bounds also exceed the target number of non-zero weights. The attenuation levels at which this occurs are therefore upper bounds on the true maximum attenuation.

Table 8.3 lists the sidelobe levels and array lengths yielded by the algorithms for each number of non-zero weights. The sparse design algorithms increase the level of attenuation by 5.3–8.8 dB over the Parks-McClellan designs, with the greatest gain in the $-20$ dB case as before. Fig. 8-2 compares the beam patterns produced by the Parks-McClellan and minimum-increase algorithms using 79 non-zero weights. We also observe that the upper bounds in Table 8.3 from feasible intervals and linear relaxation are not particularly tight, especially in the case of 79 non-zero weights where the gap is over 10 dB.

Table 8.3: Sidelobe levels and array lengths for different numbers of non-zero weights

| non-zero weights | algorithm | sidelobe level [dB] | array length (in units of $d$) |
|---|---|---|---|
| 43 | Parks-McClellan | $-20.0$ | 42 |
| | minimum-increase | $-28.8$ | 54 |
| | smallest-coefficient | $-28.3$ | 52 |
| | $p$-norm | $-28.8$ | 54 |
| | feasible intervals | $-33.6$ | – |
| | linear relaxation | $-33.0$ | – |
| 55 | Parks-McClellan | $-30.0$ | 54 |
| | minimum-increase | $-35.3$ | 78 |
| | smallest-coefficient | $-35.3$ | 78 |
| | $p$-norm | $-35.3$ | 78 |
| | feasible intervals | $-41.4$ | – |
| | linear relaxation | $-39.3$ | – |
| 79 | Parks-McClellan | $-40.0$ | 78 |
| | minimum-increase | $-46.4$ | 88 |
| | smallest-coefficient | $-46.4$ | 88 |
| | $p$-norm | $-46.4$ | 88 |
| | feasible intervals | $-59.4$ | – |
| | linear relaxation | $-57.0$ | – |

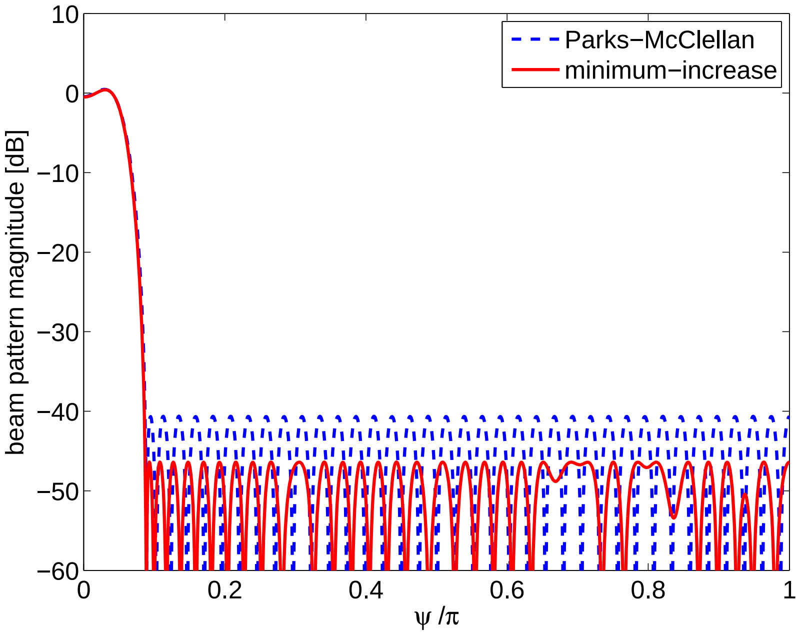


Figure 8-2: Beam patterns produced by the Parks-McClellan and minimum-increase algorithms given 79 non-zero weights.

### 8.8.2 Bandpass filters

The example in Section 8.8.1 featured a lowpass frequency response, assuming that $\psi$ is interpreted as a frequency variable. We now consider bandpass generalizations of this example obtained by shifting the passband center frequency to non-zero values $\psi_c$. The passband region is now defined by the interval $[\psi_c - \psi_p, \psi_c + \psi_p]$ and its symmetric counterpart, where $\psi_p$ is the same as in Table 8.1. The stopband region consists of the intervals $[0, \psi_c - \psi_s]$ and $[\psi_c + \psi_s, \pi]$ and their counterparts. It is assumed in this experiment that $\psi_c > \psi_s$. The magnitude tolerances for the passband and stopband are as given in Table 8.1. Note that these specifications do not correspond to modulating the array weights in Section 8.8.1 by a complex exponential, which would yield an asymmetric response, or to modulating by a cosine, which would likely increase the stopband error through addition.

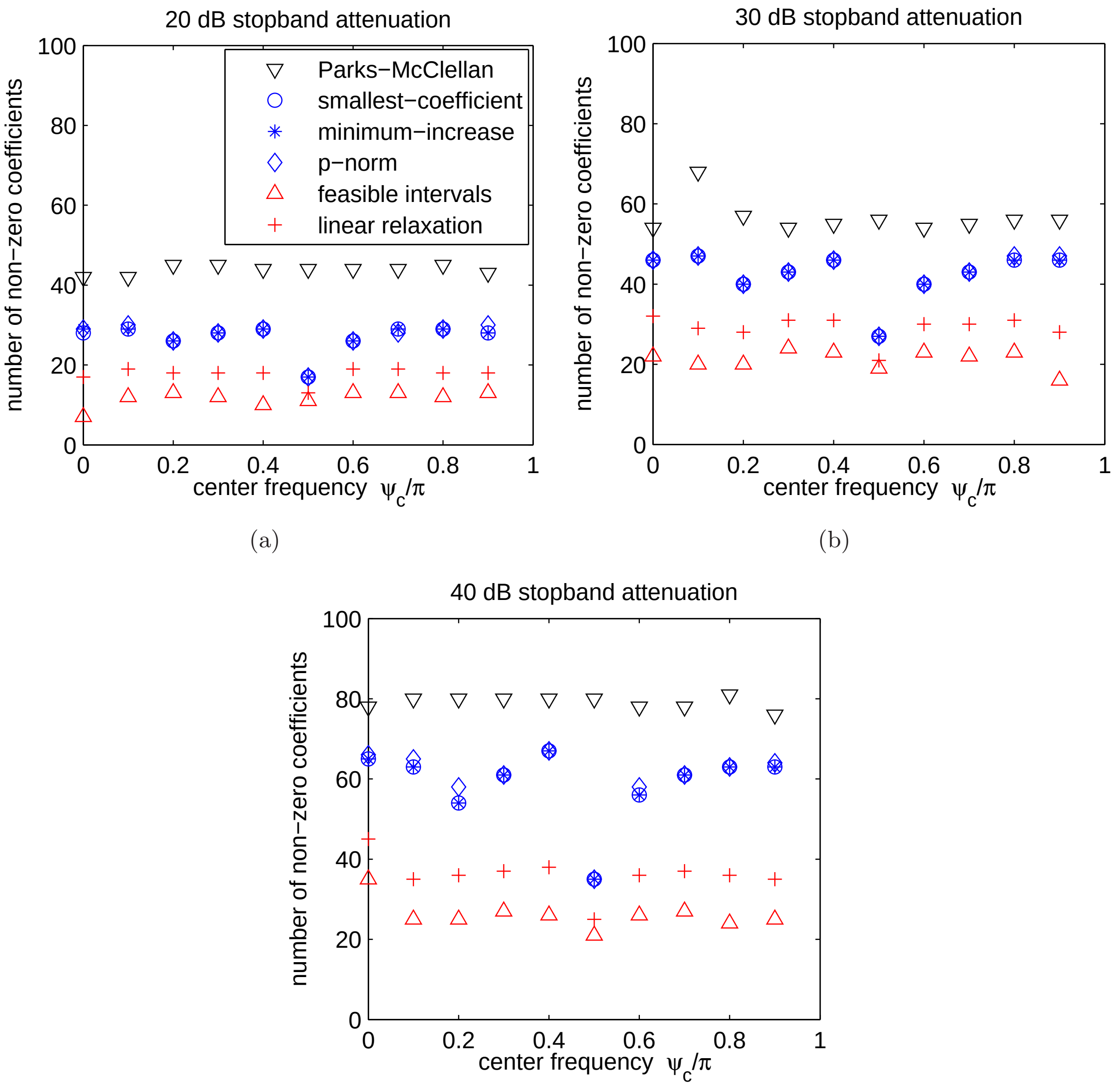
For each center frequency and stopband attenuation level, bandpass filters are designed using the sparse design algorithms of this chapter and the Parks-McClellan algorithm. As in Section 8.8.1, the sparse design algorithms have access to 50% more length than the Parks-McClellan algorithm. The lengths of the final designs however are much closer to the Parks-McClellan lengths. Both Type I and Type II linear phase are now considered.

We also compute for each instance the lower bounds based on feasible intervals and linear relaxation, again with $N$ equal to the largest allowable value.

Fig. 8-3 shows the number of non-zero filter coefficients resulting from each of the design and lower bounding methods. It appears that the center frequency has only a weak effect on the sparsity in most cases. The values $\psi_c = 0.5\pi$ and to a lesser extent $\psi_c = 0.2\pi$ are special cases and are discussed in greater detail below. The results of the two successive thinning algorithms are nearly identical, while those of the $p$-norm algorithm are slightly worse in a few instances. As in Table 8.2, the lower bound based on feasible intervals can account for a significant number of non-zero coefficients, in particular at higher attenuation levels. There is a still a sizable discrepancy however between the lower bound due to linear relaxation and the values obtained by the approximate algorithms.

In Fig. 8-4, we examine the filter impulse responses yielded by the Parks-McClellan and minimum-increase successive thinning algorithms for four contrasting center frequencies and a stopband attenuation of 40 dB. Only half of each impulse response is shown, re-indexed to allow easier comparison. We note also that the linear phase types (I or II) of the Parks-McClellan impulse responses in Fig. 8-4 are chosen to allow direct comparison with the sparse impulse responses and may not correspond to the number of non-zero coefficients in Fig. 8-3, which are chosen to be minimal. In Fig. 8-4(a), the center frequency is zero and the Parks-McClellan impulse response varies slowly. As a consequence, the zero-valued coefficients in the sparse impulse response are concentrated near the end. The center frequencies $\psi_c = 0.2\pi$ and $\psi_c = 0.5\pi$ in panels (b) and (c) are special cases. The impulse responses of the ideal infinite-length bandpass filters with these center frequencies have zero values at every fifth or second index respectively. Accordingly, the finite-length Parks-McClellan impulse responses have very small values in the same positions, which are then set to zero exactly in the sparse impulse responses. This structure accounts for the increased sparsity seen at these center frequencies in Fig. 8-3. In Fig. 8-4(d), we show impulse responses for a center frequency of $\psi_c = 0.7\pi$ that does not appear to have special properties. Compared to panel (a), the Parks-McClellan response varies more quickly and the sparse impulse response has zero values at smaller indices.

The experiment in this subsection is intended to be a preliminary exploration of the effect of the passband center frequency on the sparsity of bandpass filters. A finer grid of center frequency values may be considered in future work.

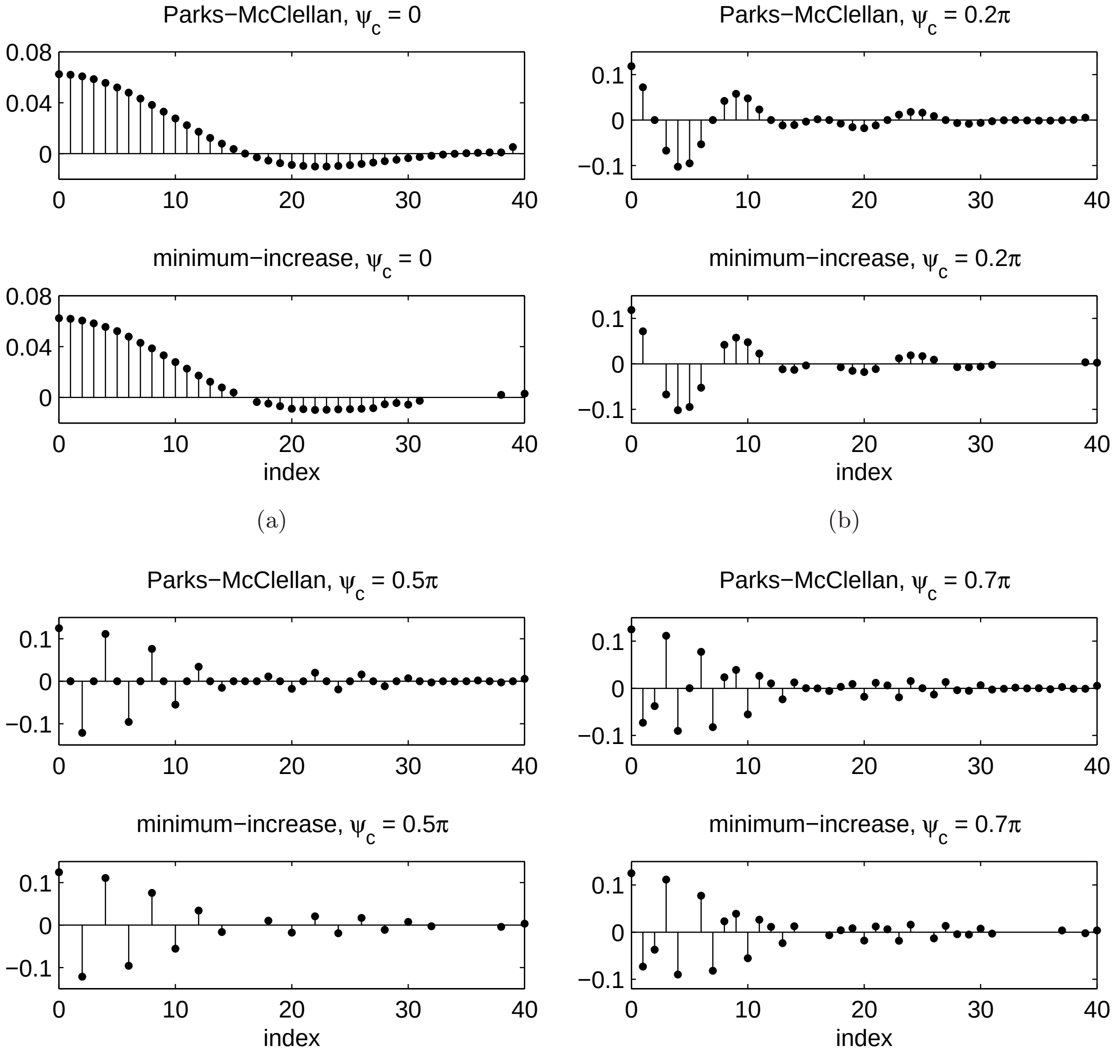


(a)

(b)

(c)

Figure 8-3: Number of non-zero coefficients returned by each design algorithm and lower bounding method for different center frequencies.



(a)

(b)

(c)

(d)

Figure 8-4: Impulse responses obtained using the Parks-McClellan and minimum-increase algorithms for different center frequencies and a stopband attenuation of 40 dB. Zero-valued coefficients are omitted.

### 8.8.3 Acoustic equalizer

As an example of a filter that is not strictly frequency-selective, we consider the design of an acoustic equalizer. In the equalization of acoustic systems such as loudspeakers and microphones, a linear-phase discrete-time filter may be used to attain a desired magnitude response while preserving the group delay of the original system to within a constant offset. Specifications for the equalizer are often given in terms of upper and lower bounds on the desired magnitude response of the overall system, which in turn specify bounds on the magnitude response of the equalizer.

In this example, we design a sparse equalizer for use in a low-frequency portion of a public address system for which the sampling rate is 400 Hz. Two sets of specifications are considered, corresponding to magnitude tolerances of ±0.25 dB and ±0.50 dB about an ideal response. Fig. 8-5 depicts the desired magnitude response and the allowable range of deviation. As in previous experiments, for the sparse design algorithms we allow 50% more length than the minimum feasible Parks-McClellan length. Only type I linear phase is used because the desired frequency response is non-zero at frequency $\pi$. Lower bounds on the optimal number of non-zero coefficients are also computed.

Table 8.4 lists the number of non-zero impulse response coefficients and the number of required delays returned by each algorithm. The reported number of delay elements assumes a causal direct-form FIR structure. In contrast to the earlier frequency-selective examples, a significantly larger gain in sparsity is obtained at the stricter tolerance level of ±0.25 dB. The impulse responses plotted in Fig. 8-6 suggest why this may be reasonable. On the other hand, the lower bounds for the ±0.25 dB case are much lower still. It seems that this is due more to the weakness of the bounds than to the sub-optimality of the solutions we have obtained.

The impulse responses given by the Parks-McClellan and minimum-increase algorithms under a ±0.25 dB tolerance are shown in Fig. 8-6. The Parks-McClellan impulse response has small values at many locations, which is exploited by the minimum-increase algorithm to increase sparsity. In this particular example, the minimum-increase algorithm has also introduced non-zero values at locations well beyond the support of the Parks-McClellan response.

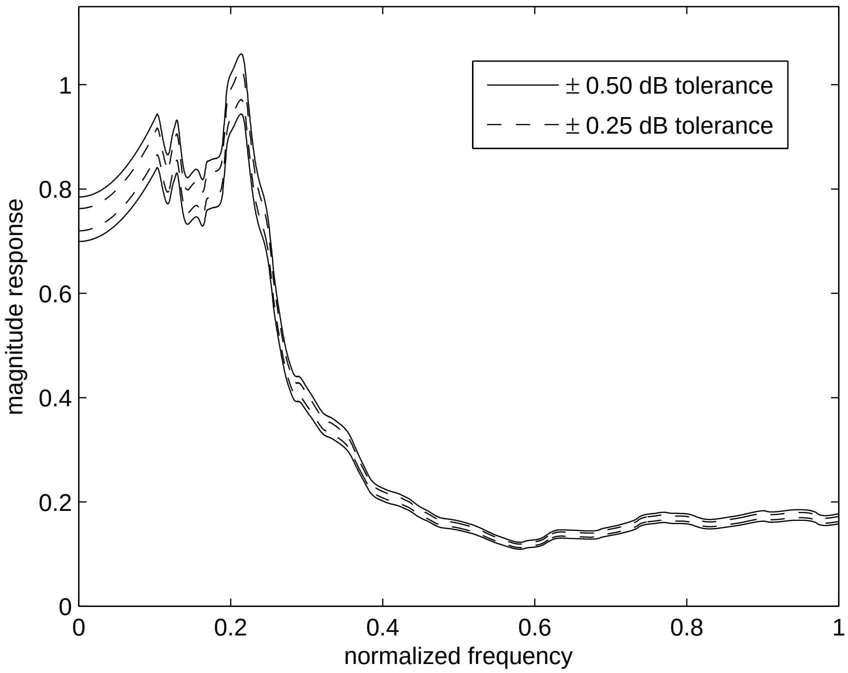


Figure 8-5: Desired equalizer magnitude response and tolerances.

Table 8.4: Numbers of non-zero impulse response values and delay elements for different equalization tolerances

| tolerance [dB] | algorithm | non-zero impulse response values | delay elements |
|---|---|---|---|
| ±0.50 | Parks-McClellan | 45 | 44 |
| | minimum-increase | 41 | 56 |
| | smallest-coefficient | 41 | 56 |
| | $p$-norm | 41 | 56 |
| | feasible intervals | 27 | – |
| | linear relaxation | 33 | – |
| ±0.25 | Parks-McClellan | 121 | 120 |
| | minimum-increase | 81 | 172 |
| | smallest-coefficient | 81 | 172 |
| | $p$-norm | 81 | 174 |
| | feasible intervals | 33 | – |
| | linear relaxation | 45 | – |

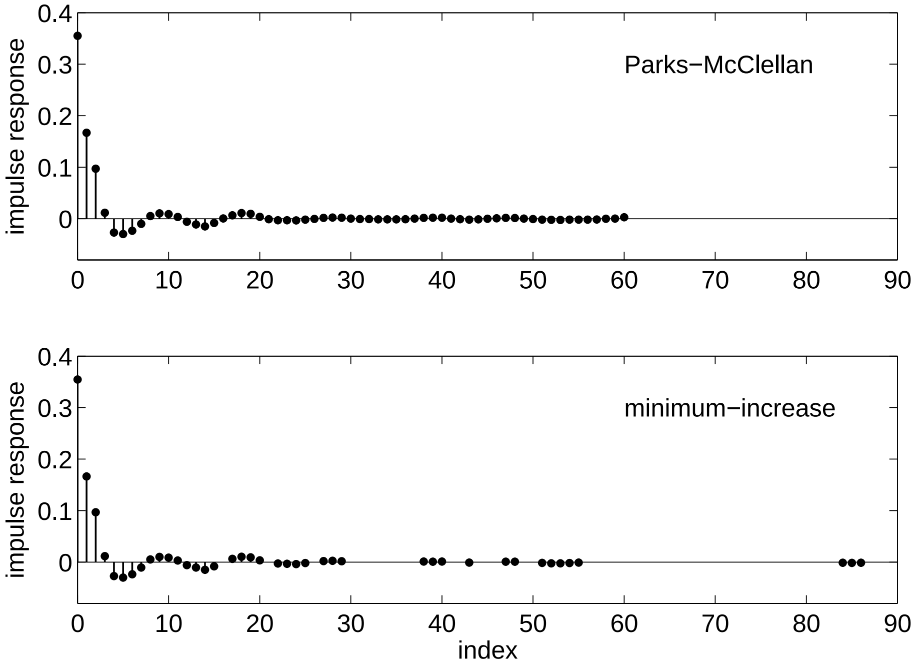


Figure 8-6: Equalizer impulse responses given by the Parks-McClellan and minimum-increase algorithms for a magnitude tolerance of ±0.25 dB. Only half of each impulse response is shown. Zero-valued coefficients are omitted.

### 8.8.4 Comparison with an optimal algorithm

In this subsection, it is shown that the approximate algorithms in this chapter can yield optimally sparse solutions. We consider Examples 2 and 3 from [3] and compare our algorithms to the commercial integer programming solver CPLEX that was used in [3]. Table 8.5 lists the specifications for the two examples. Example 2 corresponds to a wideband lowpass filter and Example 3 to the first filter in an interpolated FIR cascade from [17].

| | Example 2 | Example 3 |
|---|---|---|
| passband edge | $0.4\pi$ | $0.1616\pi$ |
| stopband edge | $0.5\pi$ | $0.2224\pi$ |
| passband ripple | 0.2 dB | 0.1612 dB |
| stopband attenuation | 60 dB | 34.548 dB |

Table 8.5: Specifications for Examples 2 and 3 from [3].

For the sparse design algorithms, we use 50% more length than required by the Parks-McClellan algorithm and consider both Type I and II linear phase. Table 8.6 displays the number of non-zero impulse response values and the number of delays returned by the algorithms. The results indicate that the approximate algorithms are capable of producing optimal designs with significantly less complexity compared to integer programming. The numbers of non-zero coefficients that we obtained using integer programming are slightly higher than those from [3]. The discrepancy is likely due to a larger number of frequency domain constraints used in the approximation of the semi-infinite constraint in (8.1.3). It is also seen that the lower bounds based on feasible intervals and linear relaxation are rather loose for these two examples.

| example | algorithm | non-zeros | delays |
|---|---|---|---|
| 2 | Parks-McClellan | 48 | 47 |
| | integer programming | 43 | 50 |
| | minimum increase | 43 | 50 |
| | smallest coefficient | 45 | 50 |
| | $p$-norm | 43 | 50 |
| | feasible intervals | 18 | – |
| | linear relaxation | 24 | – |
| 3 | Parks-McClellan | 56 | 55 |
| | integer programming | 46 | 57 |
| | minimum increase | 46 | 55 |
| | smallest coefficient | 46 | 55 |
| | $p$-norm | 46 | 55 |
| | feasible intervals | 24 | – |
| | linear relaxation | 32 | – |

Table 8.6: Results for Examples 2 and 3 from [3].

### 8.8.5 Comparison with the heuristic algorithm of [1]

We now present an example in which the approximate algorithms of this chapter perform significantly worse than an alternative heuristic algorithm proposed in [1], which implies that our algorithms can yield solutions that are far from optimal. The specific example from [1] is a lowpass filter with a passband edge of $0.26\pi$ and a stopband edge of $0.40\pi$. The number of non-zero impulse response coefficients is fixed to different values and the objective is to minimize the passband and stopband ripple assuming equal weighting.

The midpoint between the band edges in this example is $0.33\pi \approx \pi/3$, and hence the desired filter is close to a third-band filter. An ideal $n$th-band filter has the property that every $n$th coefficient in the impulse response is equal to zero except for the central coefficient. This property was exploited in [25] to pre-determine the positions of zero-valued coefficients for filters with approximately $n$th-band characteristics. In the case of the present example, the algorithm of [1], which is based on orthogonal matching pursuit (OMP), was able to automatically discover the approximate third-band structure and determine the positions of the given non-zero coefficients accordingly (two non-zero followed by one zero). Thus the resulting filters are effectively 1.5 times the length of a Parks-McClellan design with the same number of non-zero coefficients and the ripple levels are consequently much lower.

Motivated by the results in [1], we apply the algorithms developed in this chapter to the same example. For each number of non-zero coefficients, the ripple level is initialized at the Parks-McClellan value and is decreased in increments of 0.1 dB until the algorithms return a solution with more than the specified number of non-zeros. The final ripple levels are compared against those obtained by explicitly enforcing the third-band structure as done in [25], i.e., by constraining every third coefficient to zero and choosing the non-zero values in between to minimize the ripple. Since the third-band design requires 1.5 times the length of the Parks-McClellan design, we allow the same length for the successive thinning and $p$-norm algorithms. We restrict attention to Type I linear-phase filters to conform with [1].

In Table 8.7, we report the ripple levels achieved by each of the design methods. In cases where the approximate algorithms fail to improve upon the Parks-McClellan ripple level, we give either the Parks-McClellan value or the value for the same algorithm using fewer non-zero coefficients, whichever is lower. It is seen that the third-band constraint results in substantial improvements over the Parks-McClellan designs. Furthermore, the ripple values corresponding to the two lower bounding methods show that the third-band solutions are either optimal or close to optimal. For small numbers of non-zero coefficients, the approximate algorithms are able to keep pace with the third-band method. The minimum-increase algorithm, being the most complex, tracks the third-band values the longest. The two less complex approximate algorithms fall off sooner. All of our algorithms perform worse than the OMP algorithm of [1], which replicates all of the third-band values.

| non-zero coefficients | 11 | 21 | 31 | 41 | 51 | 61 | 71 | 81 |
|---|---|---|---|---|---|---|---|---|
| Parks-McClellan | −19.5 | −30.2 | −40.8 | −53.7 | −62.2 | −72.8 | −84.6 | −92.7 |
| third-band | −22.4 | −39.0 | −54.1 | −69.0 | −83.1 | −98.4 | −111.6 | −127.1 |
| minimum-increase | −22.4 | −39.0 | −54.1 | −69.0 | −83.1 | −98.4 | −109.9 | −109.9 |
| smallest-coefficient | −22.4 | −39.0 | −54.1 | −69.0 | −75.0 | −75.2 | −84.6 | −92.7 |
| $p$-norm | −22.4 | −39.0 | −54.1 | −54.1 | −62.2 | −73.1 | −84.9 | −95.0 |
| feasible intervals | −22.4 | −39.6 | −54.4 | −69.1 | −83.9 | −98.5 | −112.8 | −127.5 |
| linear relaxation | −22.4 | −39.0 | −54.1 | −69.1 | −83.9 | −98.4 | −112.7 | −127.4 |

Table 8.7: Ripple levels in dB for the first example from [1].

The preceding example suggests that the successive thinning and $p$-norm algorithms are less suited to identifying $n$th-band structure than the OMP algorithm. Conversely, our algorithms perform better on a less specialized example, also from [1]. In this second example, the passband edge is increased to $0.39\pi$ while the stopband edge remains at $0.40\pi$. The ripple level is now given, again with equal weighting, and the number of non-zero coefficients is to be minimized. Following [1], we focus on Type I linear phase and use only the lengths required by the OMP algorithm. Table 8.8 shows the number of non-zero coefficients given by the different algorithms. The successive thinning and $p$-norm algorithms are able to reduce the number of non-zero coefficients significantly relative to both the Parks-McClellan and OMP algorithms, in some cases more than halving the Parks-McClellan number.

| ripple level [dB] | 7 | 8 | 9 | 10 | 13 | 16 | 20 | 25 |
|---|---|---|---|---|---|---|---|---|
| Parks-McClellan | 13 | 21 | 29 | 39 | 69 | 101 | 145 | 205 |
| OMP | 11 | 15 | 23 | 27 | 49 | 73 | 115 | 173 |
| minimum-increase | 9 | 13 | 17 | 21 | 33 | 49 | 77 | 121 |
| smallest-increase | 9 | 13 | 17 | 21 | 35 | 49 | 79 | 121 |
| $p$-norm | 9 | 13 | 17 | 25 | 39 | 51 | 77 | 127 |

Table 8.8: Numbers of non-zero coefficients for the second example from [1].

# Chapter 9

# Conclusions and future work

In this thesis, we have considered the design of discrete-time filters according to measures of complexity that can be more closely aligned with the actual implementation cost as compared to a more conventional measure based on the total number of filter coefficients. A large part of the thesis focused on reducing the number of non-zero coefficients, motivated by the savings in computation, power consumption, hardware, or communication resulting from the elimination of operations involving zero-valued coefficients. Sparsity can be particularly important in the context of sensor arrays since the array elements can be expensive to manufacture or operate. The methods developed to increase coefficient sparsity were also extended to measures of complexity based on the number of bits in quantized representations of the filter coefficients. Specifically, we focused on the number of non-leading-zero bits in a sign-magnitude binary representation and the number of signed powers-of-two in a canonic signed digit representation.

The thesis also addressed a variety of basic filtering tasks. In particular, it was shown that the problems of weighted least-squares frequency response approximation, signal estimation, and signal detection could be unified under a single framework centered on a quadratic measure of performance. The approximation of frequency responses under a Chebyshev error criterion was also considered. Applications presented in the thesis included the design of efficient equalizers for multipath communication channels, a range of frequency-selective and frequency-shaping filters, and minimum-variance distortionless-response beamformers.

In nearly all cases, the design problems studied in the thesis are computationally difficult to solve. Several exceptions were identified in the case of a quadratic performance criterion. Specifically, it was seen that the diagonal, block-diagonal, and banded cases could be solved efficiently using greedy algorithms or dynamic programming. We focused in particular on the diagonal case, exploiting it to develop approximations.

The difficulty of the design problems in the general case motivated two basic approaches. In the first approach, the computational complexity of the design algorithms was constrained to be low. The various successive thinning algorithms, the sequential $p$-norm algorithm, and the heuristic bit reduction algorithms fall into this category. Many alternative low-complexity algorithms can be found in the literature. One of the contributions of this thesis is the attention paid to efficient implementations, particularly for the successive thinning and $p$-norm algorithms. The thesis also demonstrated through several experiments and examples that the successive thinning and $p$-norm algorithms often yield optimal or near-optimal designs. Hence these algorithms can be used with some confidence to design sparse filters when computation is limited, as for example in an adaptive setting. The evidence for near-optimality is less extensive for the heuristic bit reduction algorithms, largely because optimal solutions are very difficult to obtain.

The main weakness of most low-complexity algorithms is the lack of an estimate of the deviation from the true optimum. This weakness is addressed in the thesis by pursuing optimal algorithms, which are useful for determining fundamental limits in addition to ensuring optimal solutions. We focused in particular on branch-and-bound, a general procedure for combinatorial optimization that can often result in high computational complexity. This thesis emphasized techniques to reduce the complexity of branch-and-bound in the context of the filter design problems considered. More specifically, we concentrated on developing lower bounds on the optimal cost that can be leveraged effectively by the branch-and-bound algorithm.

The first class of bounds is based on determining the range of candidate values separately for each coefficient. We then check whether the range includes zero in the case of sparse design, or determine the value of minimum cost in the case of bit-efficient design. While the bounds based on candidate ranges are simple to compute, stronger bounds are usually desired. Toward this end, we developed two classes of relaxations, one based on linearization, the other on the diagonal case. The solutions to these relaxations yield lower bounds on the optimal value of the original problem. Of the two types of relaxations, linear relaxations are more standard and our experiments have shown that they can make branch-and-bound more efficient in the case of the non-leading-zero (NLZ) bit minimization problem. However, similar experiments indicate that linear relaxations appear to be only marginally beneficial for the signed power-of-two (SPT) minimization problem and not at all for sparsity maximization.

Diagonal relaxations on the other hand are more specific to the quadratic formulation that we considered and were shown to be more successful overall at improving efficiency. The use of diagonal relaxations can be viewed as an instance of a broader approach in which the solution to a special case is exploited to approximate and help solve the problem in the general case. A key to achieving an overall reduction in complexity is the availability of efficient methods for solving diagonal relaxations, and several techniques in this vein were discussed in the thesis. To characterize the approximation quality of diagonal relaxations, both analytical and numerical methods were employed and the dependence on problem parameters was investigated. The analysis and experiments showed that the approximation tends to be better when the matrix $\mathbf{Q}$ in the original problem is near-diagonal, well-conditioned, or has an eigenvalue distribution weighted toward small values. In the case of bit-efficient design, the lower bounds are also stronger in a relative sense for larger wordlengths. Within these general trends however, the quality of approximation can also depend on more detailed properties and further investigation may be necessary when specializing to a particular class of problem instances.

With the efficiency improvements made in this thesis to branch-and-bound, the design of optimally sparse filters under a quadratic performance constraint can be seen as a tractable problem even in moderately high dimensions, say up to 80 or 100. In situations that demand low-complexity algorithms such as with adaptive design, the value of the branch-and-bound algorithm lies in providing a computable benchmark against which the algorithms to be used in practice can be compared. For larger and more difficult problems, the branch-and-bound algorithm can be terminated early as done in some of the examples in the thesis, yielding a bound on the deviation from optimality in addition to what is in many cases a near-optimal solution. This bound is the main advantage of branch-and-bound under early termination compared to a heuristic algorithm, which provides solutions without guarantees. The typical behavior of branch-and-bound is such that the bound can often be made fairly tight in many fewer iterations than that required to certify optimality exactly.

Bit-efficient design on the other hand is a considerably more difficult problem and the development of design algorithms is consequently less mature. This thesis has demonstrated that lower bounds and relaxations can aid significantly in reducing computational complexity. However, with the current implementation in MATLAB, the solution of most problems of dimension greater than a few tens is not possible within a few hours of computation. This situation would undoubtedly be improved by implementing the algorithm in a more efficient (but less development-friendly) programming language such as C. Nevertheless, the intrinsic complexity of the problem remains very high, especially when the performance specification is loose enough to allow hundreds or even thousands of potential quantized values for each coefficient. It may be necessary therefore to reconsider the branch-and-bound framework that we have used.

Optimal algorithms for sparse filter design under a Chebyshev constraint have only been addressed at a preliminary level in the thesis. Possibilities in this direction are mentioned in the next section.

## 9.1 Future work

This thesis has focused on designing FIR filters and specifically those implemented in direct form. The extension of the complexity measures considered to other filter structures and to IIR filters could be of significant interest. Optimal design according to the measures of performance used in this work is likely to be difficult. That does not preclude however the development of design algorithms that are successful in practice. Furthermore, alternative performance measures may be defined to make the problem more tractable mathematically while still being relevant to applications.

Our presentation has focused mainly on real-valued filter coefficients. The quadratic framework of Chapters 2–7 can be modified as indicated in Section 2.1.2 to handle complex-valued coefficients provided that the real and imaginary parts are regarded as being independent. However, it is sometimes desirable to treat a complex value as a single unit; this appears to be necessary to reduce the complex-valued version of the detection problem in Section 2.1.3 to the canonical formulation. Moreover, applications such as channel equalization and beamforming are frequently formulated in terms of complex values. The complex-valued generalization of the algorithms of Chapters 2–7 is therefore deserving of further attention. Some of the techniques may be generalized with little effort. Other techniques such as linear relaxation and the use of candidate ranges in Section 6.2 are closely tied to real values and may need to be reformulated in terms of the complex modulus. The methods of Chapter 8 could also be extended to the complex case, which would encompass nonlinear-phase filters with real-valued coefficients in addition to filters with complex-valued coefficients. It is likely that the linear optimization framework used in Chapter 8 would be generalized to quadratic or second-order cone optimization.

From an optimization point of view, the development of relaxations and bounds is perhaps the main contribution of this work and is also a potentially rich area for future study. For the problem of quadratically constrained sparse filter design, an intriguing possibility is to consider relaxations based on the other special cases in Section 2.2, and especially the tridiagonal case which is known to be efficiently solvable. While it was tractable computationally to determine the best possible diagonal relaxation, it is unclear whether this is still the case for tridiagonal relaxations. Hence an alternative criterion such as minimum volume may need to be adopted. It may also be possible to devise more efficient algorithms for higher-bandwidth cases (pentadiagonal and so forth) and thereby obtain higher-bandwidth relaxations. Specifically, there may be a connection between the higher-bandwidth case and the junction tree algorithm [118], which aggregates nodes in a graphical model into supernodes before performing inference. Alternative relaxations could be of even greater importance for bit-efficient design given its higher difficulty compared to sparse filter design. Such relaxations may require the identification of additional special cases to exploit.

The existing relaxations are also a source of future work. One interesting idea is to combine the linear and diagonal relaxations in such a way that the new relaxation is stronger than either alone. A straightfoward but somewhat inefficient way is to solve both relaxations and then take the maximum of the resulting lower bounds. For the diagonal relaxation of (2.0.1), more analysis can be done to understand in particular the dependence on the eigenvalue distribution of $\mathbf{Q}$, possibly using a stochastic approach instead of the deterministic approach taken in Section 3.4. An alternative to the minimum volume criterion used in Section 6.4.1 may be considered for the diagonal relaxation of problems (5.1.1) and (5.1.2). Linear relaxations may benefit from having more specialized and efficient solvers such as the ones used for the diagonal relaxations.

It is clear from Chapter 8 that an optimal algorithm for sparse filter design under a Chebyshev constraint has yet to be fully developed, and specifically one that is tailored to the problem and not a general-purpose solver such as CPLEX. It appears that this will require relaxations that are more powerful than the linear relaxation in Section 8.6. Higher-order generalizations of linear relaxations have been developed in the integer optimization literature [citations] and these could be applied to the mixed-integer formulation (8.4.1). Another possibility is to rewrite the binary constraint on the indicator variables in (8.4.1) as the quadratic constraint $i_n^2 - i_n = 0$, and then apply semidefinite relaxations to the resulting non-convex quadratic program [119,120]. These relaxations could be applied to the quadratically constrained problem (3.1.1) as well. The currently high computational cost of evaluating the bound in Section 8.5 and determining the tightest possible linear relaxation could be addressed by developing an efficient specialized solver for linear programs in which the constraints are known to represent frequency domain specifications. Such a solver would also benefit the successive thinning algorithms in Section 8.2. As one example of what could be exploited, multiplication by the matrix $\mathbf{A}$ defined in (8.1.4a), which transforms a coefficient vector $\mathbf{b}$ into mostly uniformly-spaced samples of the frequency response, could be implemented more efficiently using the FFT.

From the perspective of design applications, more careful studies could be done to determine the effect of various specifications and problem characteristics on the expected level of sparsity or the expected number of bits. For example, the dependence on attenuation levels and passband center frequencies for frequency-selective filters was explored only at a preliminary level in Section 8.8. It would also be interesting to understand how sparse or bit-efficient an equalizer may be made for a channel that does not already have an approximately sparse response. As for the heuristic algorithms, it was seen in Section 8.8.5 that one situation in which they do not perform as well is the case of approximate $n$th-band filters. It would be desirable to identify additional cases in which they fail and to evaluate their performance more thoroughly.

More broadly, the methods developed in this thesis could have applications beyond signal processing that fall within a similar mathematical framework. Subset selection for linear regression [93] is one such example, portfolio optimization [121] may be another.

# Chapter 10

# Evolution of the thesis

This thesis has as its immediate ancestor Tom Baran's master's thesis [122], in particular his work on using 1-norm minimization to design sparse filters subject to a Chebyshev constraint on the frequency response. Inspired by Tom's work, the first part of this thesis to emerge was the successive thinning algorithm of Section 8.2, which was intended as an alternative heuristic. This was followed by the $p$-norm algorithm in Section 8.3 as a natural extension of the 1-norm approach. So Chapter 8 is really the first chapter chronologically. Early publications [111,112] were the source of many of the design examples in Section 8.8. A branch-and-bound algorithm and the lower bounding methods of Sections 8.5 and 8.6 suggested themselves fairly early on, but these are fairly standard techniques and it was unclear how the structure of the filter design problem could be exploited.

Also early in the Ph.D., Al Oppenheim suggested expanding the scope to include bit sparsity for finite-precision representations as well as coefficient sparsity. It was through an exploration of the literature in discrete-coefficient filter design that the idea of enclosing shapes, specifically boxes and ellipsoids, began to take hold. The literature review also led to a focus on the CSD representation. The thesis then started drifting toward quadratic performance criteria, initially still tied to frequency response approximation as before. This drift was fortunate since the quadratic version of the problem proved to be much more fruitful and accessible to analysis. A suggestion from Petros Boufounos in a DSPG meeting triggered an important generalization to filter design for signal detection and estimation. Thus the quadratic framework of Chapters 2–7 was born. A very productive period followed in which the diagonal relaxation was developed (the tractability of obtaining the tightest possible diagonal relaxation was a nice coup), and an efficient solver for diagonal relaxations as well as a branch-and-bound algorithm were implemented in MATLAB. The theoretical analysis of the diagonal relaxation took a longer time to mature and the process is still ongoing. More careful analysis of the linear relaxation and work on special cases were spurred by the presentation of preliminary results at ICASSP and by interaction with Charles Sestok.

Although bit complexity had been a theme in the thesis for a long time, the real work on bit-efficient design in Chapters 5–7 was only begun after the framework and results for quadratically-constrained sparse design were established. Because of this precedent, the work proceeded very quickly. It is apparent that each section in Chapters 5–7 has as its parallel and draws upon the corresponding section in Chapters 2–4.

The numerical experiments and design examples in Chapters 4 and 7 were the last part of the puzzle to fall into place. These turned out to be richer than first envisioned, at least for sparse design, and made the thesis more complete and satisfying. The design experiments were facilitated by collaboration and discussion with Xue Feng, Ballard Blair, and Jon Paul Kitchens. To anyone who has completed a Ph.D. thesis, it may not come as a surprise that the last experiments to make it into the thesis finished only a day before submission.

After thesis submission comes optimal relaxation.

# Appendix A

# Derivations and proofs for Chapter 2

## A.1 Enumeration of the subsets required in the algorithm of [2] for penta-diagonal Q

In this appendix, we determine the number of subsets that must be evaluated in the dynamic programming algorithm of [2] for the case of penta-diagonal $\mathbf{Q}$. In the penta-diagonal case ($W = 2$), the subsets to be enumerated have the property that when the indices in a subset are listed in order (increasing or decreasing), any two indices that are adjacent in the sequence differ by at most 2. For example, the subset $\{3, 5, 7\}$ is permissible while $\{3, 6, 7\}$ is not.

We use three numbers to parameterize the subsets: the cardinality $M$, the smallest index $i$, and the number $\ell$ of index values spanned by the subset, i.e., the difference between the largest and smallest indices plus one. We assume that (2.2.1) is to be evaluated for $N - K = 1, \ldots, M_0$, so that $M$ ranges from 1 to $M_0$. We also assume that $M_0$ grows proportionally to $N$. For a fixed value of $M$, $\ell$ can range from $M$ to $\min(2M - 1, N)$. The lowest value for $\ell$ corresponds to all indices being consecutive in value, while the highest value corresponds to each index differing by two from its neighbors, subject to not exceeding the maximum span of $N$. For a fixed value of $\ell$, $i$ can range from 1 to $N - \ell + 1$.

We first determine the number of subsets having cardinality $M$ and span $\ell$. To simplify the counting argument, we represent the indices belonging to a subset by ones and the indices not belonging to the subset but falling within its span by zeros. Thus the problem is equivalent to counting the number of ways of ordering $M$ ones and $\ell - M$ zeros such that no two zeros are adjacent. It follows that a one must be placed between every pair of zeros and also at either end, as otherwise the span would be less than $\ell$. This fixes the locations of $\ell - M + 1$ of the ones relative to the zeros and leaves $2M - \ell - 1$ ones remaining to be placed. There are $\ell - M + 1$ distinct positions for the remaining ones, defined relative to the positions of the zeros (e.g. before the first zero, between the first and second zeros, etc.). By a classical result from combinatorics (see e.g. [123]), the number of distinct orderings is given by

$$\binom{(2M - \ell - 1) + (\ell - M + 1) - 1}{(\ell - M + 1) - 1} = \binom{M - 1}{\ell - M}. \tag{A.1.1}$$

The total number of subsets, $T$, is equal to the sum of the quantity in (A.1.1) over all possible values of $M$, $\ell$, and $i$. Since there is no dependence on $i$,

$$T = \sum_{M=1}^{M_0} \sum_{\ell=M}^{\min(2M-1,N)} (N - \ell + 1)\binom{M - 1}{\ell - M}. \tag{A.1.2}$$

If $M_0 \leq \lceil N/2 \rceil$, then $\min(2M - 1, N) = 2M - 1$ for every value of $M$ in (A.1.2). For $M_0 > \lceil N/2 \rceil$, we decompose the sum over $M$ into two sums as follows,

$$T = \sum_{M=1}^{\lceil N/2 \rceil} \sum_{\ell=M}^{2M-1} (N - \ell + 1)\binom{M - 1}{\ell - M} + \sum_{M=\lceil N/2 \rceil + 1}^{M_0} \sum_{\ell=M}^{N} (N - \ell + 1)\binom{M - 1}{\ell - M},$$

and then discard the second sum in order to obtain a lower bound on $T$. In either case we have

$$\begin{aligned} T &\geq \sum_{M=1}^{M_0'} \sum_{\ell=M}^{2M-1} (N - \ell + 1)\binom{M - 1}{\ell - M} \\ &= \sum_{M=1}^{M_0'} \sum_{\ell=0}^{M-1} (N + 1 - M - \ell)\binom{M - 1}{\ell}, \end{aligned} \tag{A.1.3}$$

where $M_0' = \min(M_0, \lceil N/2 \rceil)$ and the second line is obtained from the first via the change of variables $\ell - M \to \ell$. The summations in (A.1.3) can now be evaluated using standard formulas such as

$$\sum_{\ell=0}^{M-1} \binom{M-1}{\ell} = 2^{M-1}, \qquad \sum_{\ell=0}^{M-1} \ell \binom{M-1}{\ell} = (M-1)2^{M-2},$$

yielding

$$T \geq \left(N + 3 - \frac{3}{2} M_0'\right) 2^{M_0'} - N - 3. \tag{A.1.4}$$

Since $M_0' = \min(M_0, \lceil N/2 \rceil)$ and we have assumed that $M_0$ scales linearly with $N$, the number of subsets $T$ grows at least as fast as $N \cdot 2^{\alpha N}$ for some constant fraction $\alpha$.

## A.2 Proof of Theorem 1

First, by interchanging $\mathbf{x}_1$ and $\mathbf{x}_2$ in (2.2.13) we infer that

$$\|\mathbf{x}_1\|_0 = \|\mathbf{x}_2\|_0 \quad \Longrightarrow \quad \left\|\mathbf{T}^{-1}\mathbf{x}_1\right\|_0 = \left\|\mathbf{T}^{-1}\mathbf{x}_2\right\|_0 \quad \forall\, \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^M. \tag{A.2.1}$$

Consider a vector $\mathbf{x}_1$ for which $\|\mathbf{x}_1\|_0 = 1$. If $\left\|\mathbf{T}^{-1}\mathbf{x}_1\right\|_0 = 0$, then (A.2.1) implies that $\mathbf{T}^{-1}\mathbf{e}_m = \mathbf{0}$ for all standard basis vectors $\mathbf{e}_m$, $m = 1, \ldots, M$, and therefore $\mathbf{T}^{-1} = \mathbf{0}$, contradicting the fact that $\mathbf{T}^{-1}$ is a left-inverse. If $\left\|\mathbf{T}^{-1}\mathbf{x}_1\right\|_0 > 1$, then according to (2.2.13), $\left\|\mathbf{T}^{-1}\mathbf{x}\right\|_0 > 1$ for all non-zero $\mathbf{x} \in \mathbb{R}^M$ and $\mathbf{T}^{-1}$ is not surjective, again contradicting the left-inverse property. Hence we must have $\left\|\mathbf{T}^{-1}\mathbf{x}\right\|_0 = 1$ for all $\mathbf{x}$ with a single non-zero component, and in particular for $\mathbf{x} = \mathbf{e}_m$, $m = 1, \ldots, M$. This implies that each column of $\mathbf{T}^{-1}$ is a non-zero multiple of a standard basis vector $\mathbf{e}_n$ in $\mathbb{R}^N$. In the case $M = N$, no two columns of $\mathbf{T}^{-1}$ can be multiples of the same standard basis vector as otherwise $\mathbf{T}^{-1}$ would not be a surjection. Since each column of $\mathbf{T}^{-1}$ is a multiple of a different standard basis vector, $\mathbf{T}^{-1}$ can be transformed into a diagonal matrix by a permutation, and $\mathbf{T}$ is therefore a composition of a permutation and a diagonal scaling. In the case $M > N$, there must be two columns of $\mathbf{T}^{-1}$, say columns $m_1$ and $m_2$, that are multiples of each other. Then by choosing a multiplier $a$ to produce cancellation, we have

$$\|\mathbf{e}_{m_1} + a\mathbf{e}_{m_2}\|_0 = 2 \quad \text{and} \quad \left\|\mathbf{T}^{-1}\left(\mathbf{e}_{m_1} + a\mathbf{e}_{m_2}\right)\right\|_0 = 0,$$

which together with $\|\mathbf{e}_{m_1}\|_0 = \left\|\mathbf{T}^{-1}\mathbf{e}_{m_1}\right\|_0 = 1$ violates (2.2.13). Thus it is not possible to meet the desired criteria with $M > N$.

## A.3 Derivation of subproblem parameters

In this appendix, we show that an arbitrary subproblem defined by subsets ($\mathcal{Z}$,$\mathcal{U}$, $\mathcal{F}$) can be reduced to the form in (2.3.1), which is a lower-dimensional version of the original problem (2.0.1). Recall that the subset $\mathcal{Z}$ represents the coefficients that have been constrained to a value of zero, $\mathcal{U}$ represents the coefficients designated as being non-zero in the cost function, and $\mathcal{F}$ represents the remaining coefficients. The assignment of coefficients to $\mathcal{U}$ can either be by necessity because a zero value is no longer feasible, or by choice in the context of the branch-and-bound procedure. We derive expressions for the parameters of the subproblem in terms of the original parameters $\mathbf{Q}$, $\mathbf{c}$ and $\gamma$.

We first consider the two special cases $\mathcal{U} = \emptyset$ and $\mathcal{Z} = \emptyset$ and then combine the results to arrive at the general case. For the case $\mathcal{U} = \emptyset$, the subset $\mathcal{F}$ is equal to $\mathcal{Y}$, the complement of $\mathcal{Z}$, and the quadratic constraint with $b_n = 0$ for $n \in \mathcal{Z}$ is given by (2.2.2). By completing the square, (2.2.2) can be rewritten as

$$\left(\mathbf{b}_{\mathcal{Y}} - \mathbf{c}_{\mathcal{Y}} - (\mathbf{Q}_{\mathcal{Y}\mathcal{Y}})^{-1}\mathbf{Q}_{\mathcal{Y}\mathcal{Z}}\mathbf{c}_{\mathcal{Z}}\right)^T \mathbf{Q}_{\mathcal{Y}\mathcal{Y}} \left(\mathbf{b}_{\mathcal{Y}} - \mathbf{c}_{\mathcal{Y}} - (\mathbf{Q}_{\mathcal{Y}\mathcal{Y}})^{-1}\mathbf{Q}_{\mathcal{Y}\mathcal{Z}}\mathbf{c}_{\mathcal{Z}}\right) \le \gamma - \mathbf{c}_{\mathcal{Z}}^T(\mathbf{Q}/\mathbf{Q}_{\mathcal{Y}\mathcal{Y}})\mathbf{c}_{\mathcal{Z}}.$$

Comparing this with (2.1.1), we see that the subproblem defined by ($\mathcal{Z}$, $\emptyset$, $\mathcal{Y}$) can be formulated as

$$\min_{\mathbf{b}_{\mathcal{Y}}} \quad \|\mathbf{b}_{\mathcal{Y}}\|_0 \qquad \text{s.t.} \quad \left(\mathbf{b}_{\mathcal{Y}} - \mathbf{c}'_{\mathcal{Y}}\right)^T \mathbf{Q}_{\mathcal{Y}\mathcal{Y}} \left(\mathbf{b}_{\mathcal{Y}} - \mathbf{c}'_{\mathcal{Y}}\right) \le \gamma_{\text{eff}}, \tag{A.3.1}$$

with $\mathbf{c}'_{\mathcal{Y}} = \mathbf{c}_{\mathcal{Y}} + (\mathbf{Q}_{\mathcal{Y}\mathcal{Y}})^{-1}\mathbf{Q}_{\mathcal{Y}\mathcal{Z}}\mathbf{c}_{\mathcal{Z}}$ and $\gamma_{\text{eff}} = \gamma - \mathbf{c}_{\mathcal{Z}}^T(\mathbf{Q}/\mathbf{Q}_{\mathcal{Y}\mathcal{Y}})\mathbf{c}_{\mathcal{Z}}$. Problem (A.3.1) is a lower-dimensional instance of (2.0.1) with $\mathbf{Q}$ replaced by $\mathbf{Q}_{\mathcal{Y}\mathcal{Y}}$, $\mathbf{c}$ by $\mathbf{c}'_{\mathcal{Y}}$, and $\gamma$ by $\gamma_{\text{eff}}$.

Next we consider the case $\mathcal{Z} = \emptyset$. Since the variables $b_n$ for $n \in \mathcal{U}$ have been designated as being non-zero, the zero-norm $\|\mathbf{b}\|_0$ may be rewritten as $|\mathcal{U}| + \|\mathbf{b}_{\mathcal{F}}\|_0$. Problem (2.0.1) becomes

$$|\mathcal{U}| + \min_{\mathbf{b}_{\mathcal{U}},\mathbf{b}_{\mathcal{F}}} \|\mathbf{b}_{\mathcal{F}}\|_0 \qquad \text{s.t.} \qquad \begin{bmatrix} (\mathbf{b}_{\mathcal{U}} - \mathbf{c}_{\mathcal{U}})^T & (\mathbf{b}_{\mathcal{F}} - \mathbf{c}_{\mathcal{F}})^T \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{\mathcal{U}\mathcal{U}} & \mathbf{Q}_{\mathcal{U}\mathcal{F}} \\ \mathbf{Q}_{\mathcal{F}\mathcal{U}} & \mathbf{Q}_{\mathcal{F}\mathcal{F}} \end{bmatrix} \begin{bmatrix} \mathbf{b}_{\mathcal{U}} - \mathbf{c}_{\mathcal{U}} \\ \mathbf{b}_{\mathcal{F}} - \mathbf{c}_{\mathcal{F}} \end{bmatrix} \le \gamma, \tag{A.3.2}$$

where we have partitioned $\mathbf{Q}$, $\mathbf{b}$, and $\mathbf{c}$ according to the subsets $\mathcal{F}$ and $\mathcal{U}$. The variables $b_n$, $n \in \mathcal{U}$, appear in the constraint in (A.3.2) but not in the objective. Thus we are free to choose a value for $\mathbf{b}_{\mathcal{U}}$ without regard to its cost since it is already accounted for by the term $|\mathcal{U}|$. In the interest of minimizing $\|\mathbf{b}_{\mathcal{F}}\|_0$, it is best to choose $\mathbf{b}_{\mathcal{U}}$ as a function of $\mathbf{b}_{\mathcal{F}}$

to maximize the margin in the constraint, thereby making the set of feasible $\mathbf{b}_\mathcal{F}$ as large as possible. This is equivalent to minimizing the left-hand side of the constraint with respect to $\mathbf{b}_\mathcal{U}$ while holding $\mathbf{b}_\mathcal{F}$ constant. By a calculation similar to the one made in deriving (2.2.3) from (2.2.2), we obtain

$$\mathbf{b}_\mathcal{U}^* = \mathbf{c}_\mathcal{U} - (\mathbf{Q}_{\mathcal{U}\mathcal{U}})^{-1}\,\mathbf{Q}_{\mathcal{U}\mathcal{F}}(\mathbf{b}_\mathcal{F} - \mathbf{c}_\mathcal{F}) \tag{A.3.3}$$

as the minimizer. Substituting (A.3.3) into (A.3.2) results in

$$|\mathcal{U}| + \min_{\mathbf{b}_\mathcal{U},\mathbf{b}_\mathcal{F}} \|\mathbf{b}_\mathcal{F}\|_0 \qquad \text{s.t.} \qquad (\mathbf{b}_\mathcal{F} - \mathbf{c}_\mathcal{F})^T(\mathbf{Q}/\mathbf{Q}_{\mathcal{U}\mathcal{U}})(\mathbf{b}_\mathcal{F} - \mathbf{c}_\mathcal{F}) \le \gamma, \tag{A.3.4}$$

where the Schur complement $\mathbf{Q}/\mathbf{Q}_{\mathcal{U}\mathcal{U}} = \mathbf{Q}_{\mathcal{F}\mathcal{F}} - \mathbf{Q}_{\mathcal{F}\mathcal{U}}\,(\mathbf{Q}_{\mathcal{U}\mathcal{U}})^{-1}\,\mathbf{Q}_{\mathcal{U}\mathcal{F}}$. Problem (A.3.4) is of the same form as (2.0.1) with $|\mathcal{F}|$ variables instead of $N$, $\mathbf{Q}/\mathbf{Q}_{\mathcal{U}\mathcal{U}}$ in place of $\mathbf{Q}$, $\mathbf{c}_\mathcal{F}$ in place of $\mathbf{c}$, and $\gamma$ unchanged.

For a general subproblem defined by $(\mathcal{Z},\mathcal{U},\mathcal{F})$ with $\mathcal{U} \neq \emptyset$ and $\mathcal{Z} \neq \emptyset$, we may start from (A.3.1), which corresponds to the subproblem $(\mathcal{Z},\emptyset,\mathcal{F})$, and then apply the same reductions as in the case $\mathcal{Z} = \emptyset$, treating (A.3.1) as the original problem. Thus $\mathbf{Q}_{\mathcal{Y}\mathcal{Y}}$, $\mathbf{c}'_\mathcal{Y}$, and $\gamma_\text{eff}$ in (A.3.1) play the roles of $\mathbf{Q}$, $\mathbf{c}$, and $\gamma$ in (2.0.1). Transforming (A.3.1) in the same way as (2.0.1) was transformed into (A.3.4) and noting that $\mathcal{Y} = \mathcal{U} \cup \mathcal{F}$, we arrive at (2.3.1) with effective parameters given by

$$\mathbf{Q}_\text{eff} = \mathbf{Q}_{\mathcal{Y}\mathcal{Y}}/\mathbf{Q}_{\mathcal{U}\mathcal{U}} = \mathbf{Q}_{\mathcal{F}\mathcal{F}} - \mathbf{Q}_{\mathcal{F}\mathcal{U}}\,(\mathbf{Q}_{\mathcal{U}\mathcal{U}})^{-1}\,\mathbf{Q}_{\mathcal{U}\mathcal{F}}, \tag{A.3.5a}$$

$$\mathbf{c}_\text{eff} = \mathbf{c}_\mathcal{F} + \big((\mathbf{Q}_{\mathcal{Y}\mathcal{Y}})^{-1}\mathbf{Q}_{\mathcal{Y}\mathcal{Z}}\mathbf{c}_\mathcal{Z}\big)_\mathcal{F} = \mathbf{c}_\mathcal{F} + (\mathbf{Q}_\text{eff})^{-1}\big(\mathbf{Q}_{\mathcal{F}\mathcal{Z}} - \mathbf{Q}_{\mathcal{F}\mathcal{U}}(\mathbf{Q}_{\mathcal{U}\mathcal{U}})^{-1}\mathbf{Q}_{\mathcal{U}\mathcal{Z}}\big)\mathbf{c}_\mathcal{Z}, \tag{A.3.5b}$$

$$\gamma_\text{eff} = \gamma - \mathbf{c}_\mathcal{Z}^T(\mathbf{Q}/\mathbf{Q}_{\mathcal{Y}\mathcal{Y}})\mathbf{c}_\mathcal{Z}. \tag{A.3.5c}$$

The second equality in (A.3.5b) is obtained by expressing $\mathbf{Q}_{\mathcal{Y}\mathcal{Y}}$ as a $2 \times 2$ block matrix corresponding to the partition $\mathcal{Y} = \mathcal{U} \cup \mathcal{F}$ and applying the formula for the inverse of a $2 \times 2$ block matrix to $(\mathbf{Q}_{\mathcal{Y}\mathcal{Y}})^{-1}$. Equations (A.3.5) with the second form of (A.3.5b) are identical to (2.3.2). We may also define $\mathbf{f}_\text{eff} = \mathbf{Q}_\text{eff}\mathbf{c}_\text{eff}$ in analogy with $\mathbf{f} = \mathbf{Q}\mathbf{c}$. It can be shown that

$$\mathbf{f}_\text{eff} = \mathbf{f}_\mathcal{F} - \mathbf{Q}_{\mathcal{F}\mathcal{U}}(\mathbf{Q}_{\mathcal{U}\mathcal{U}})^{-1}\mathbf{f}_\mathcal{U}. \tag{A.3.6}$$

# Appendix B

# Derivations and proofs for Chapter 3

## B.1 Derivation of (3.3.3)

This appendix shows how (3.3.3a) is derived; the derivation of (3.3.3b) is similar. Let $\mathbf{b}^*$ denote an optimal solution to the maximization in (3.3.3a). The Karush-Kuhn-Tucker optimality condition (see [98]) for (3.3.3a) is given by

$$\mathbf{e}_n = \lambda \mathbf{Q}(\mathbf{b}^* - \mathbf{c}), \tag{B.1.1}$$

where $\lambda$ is a non-negative Lagrange multiplier. Since (B.1.1) implies that $\lambda$ cannot be zero, (B.1.1) can be inverted to yield

$$\mathbf{b}^* - \mathbf{c} = \frac{1}{\lambda}\mathbf{Q}^{-1}\mathbf{e}_n. \tag{B.1.2}$$

A non-zero value for $\lambda$ also implies that the constraint in (3.3.3a), i.e., constraint (2.1.1), must be met with equality. Substituting (B.1.2) into (2.1.1) to solve for $\lambda$,

$$\frac{1}{\lambda^2}\mathbf{e}_n^T\mathbf{Q}^{-1}\mathbf{e}_n = \frac{1}{\lambda^2}\left(\mathbf{Q}^{-1}\right)_{nn} = \gamma,$$
$$\lambda = \sqrt{\frac{\left(\mathbf{Q}^{-1}\right)_{nn}}{\gamma}}.$$

Hence

$$\mathbf{b}^* = \mathbf{c} + \sqrt{\frac{\gamma}{\left(\mathbf{Q}^{-1}\right)_{nn}}}\mathbf{Q}^{-1}\mathbf{e}_n$$

and the maximum value is given by the $n$th element of $\mathbf{b}^*$, i.e.,

$$\max\left\{b_n : (\mathbf{b}-\mathbf{c})^T\mathbf{Q}(\mathbf{b}-\mathbf{c}) \leq \gamma\right\} = \mathbf{e}_n^T\mathbf{b}^* = c_n + \sqrt{\frac{\gamma}{\left(\mathbf{Q}^{-1}\right)_{nn}}}\mathbf{e}_n^T\mathbf{Q}^{-1}\mathbf{e}_n = c_n + \sqrt{\gamma\left(\mathbf{Q}^{-1}\right)_{nn}}.$$

## B.2 Derivation of the dual of the linear relaxation (3.3.8)

We refer the reader to [98] for background in duality theory in optimization. The Lagrangian for problem (3.3.8) can be written as

$$L = (\mathbf{g}^+ - \boldsymbol{\mu}^+)^T\mathbf{b}^+ + (\mathbf{g}^- - \boldsymbol{\mu}^-)^T\mathbf{b}^- + \frac{\lambda}{2}\left((\mathbf{b}^+ - \mathbf{b}^- - \mathbf{c})^T\mathbf{Q}(\mathbf{b}^+ - \mathbf{b}^- - \mathbf{c}) - \gamma\right), \quad \text{(B.2.1)}$$

where $g_n^+ = 1/B_n^+$ and $g_n^- = 1/B_n^-$ for all $n$, and $\lambda$, $\boldsymbol{\mu}^+$, and $\boldsymbol{\mu}^-$ are non-negative Lagrange multipliers. The general form of the dual problem is

$$\max_{\lambda,\boldsymbol{\mu}^+,\boldsymbol{\mu}^-}\ \min_{\mathbf{b}^+,\mathbf{b}^-}\ L. \quad \text{(B.2.2)}$$

We show that it is only necessary to consider $\boldsymbol{\mu}^+$, $\boldsymbol{\mu}^-$ satisfying $\mathbf{g}^+ - \boldsymbol{\mu}^+ + \mathbf{g}^- - \boldsymbol{\mu}^- = \mathbf{0}$ in the outer maximization. Suppose that there is an index $n$ for which $g_n^+ - \mu_n^+ + g_n^- - \mu_n^- \neq 0$. Let $\mathbf{b}^+ = \alpha\mathbf{e}_n + \mathbf{c}/2$ and $\mathbf{b}^- = \alpha\mathbf{e}_n - \mathbf{c}/2$ with $\alpha$ an arbitrary real number, so that $\mathbf{b}^+ - \mathbf{b}^- - \mathbf{c} = \mathbf{0}$. Then (B.2.1) becomes

$$L = \alpha(g_n^+ - \mu_n^+ + g_n^- - \mu_n^-) + \frac{1}{2}\left((\mathbf{g}^+ - \boldsymbol{\mu}^+ - \mathbf{g}^- + \boldsymbol{\mu}^-)^T\mathbf{c} - \lambda\gamma\right),$$

which is affine in $\alpha$. By taking $\alpha$ to $+\infty$ or $-\infty$, the value of the inner minimization in (B.2.2) approaches $-\infty$ and therefore does not need to be considered in the outer maximization. Henceforth we assume that $\mathbf{g}^+ - \boldsymbol{\mu}^+ + \mathbf{g}^- - \boldsymbol{\mu}^- = \mathbf{0}$.

The case $\lambda = 0$ can also be excluded from the maximization in (B.2.2). If $\lambda = 0$, the value of the minimization over $\mathbf{b}^+$, $\mathbf{b}^-$ is $-\infty$ unless $\mathbf{g}^+ - \boldsymbol{\mu}^+ = \mathbf{g}^- - \boldsymbol{\mu}^- = \mathbf{0}$, in which case the value is zero. As will be seen, the value of the outer maximization is at least equal to zero, so the case $\lambda = 0$ does not have to be considered further.

The first-order optimality condition for the minimization in (B.2.2) reads

$$\begin{bmatrix} \mathbf{g}^+ - \boldsymbol{\mu}^+ \\ \mathbf{g}^- - \boldsymbol{\mu}^- \end{bmatrix} + \lambda \begin{bmatrix} \mathbf{Q} & -\mathbf{Q} \\ -\mathbf{Q} & \mathbf{Q} \end{bmatrix} \begin{bmatrix} \mathbf{b}^{+*} - \mathbf{c} \\ \mathbf{b}^{-*} \end{bmatrix} = \mathbf{0}.$$

Given that $\mathbf{g}^+ - \boldsymbol{\mu}^+ + \mathbf{g}^- - \boldsymbol{\mu}^- = \mathbf{0}$, the second row is the negative of the first and hence is redundant. For $\lambda > 0$, the first row can be solved to yield

$$\mathbf{b}^{+*} - \mathbf{b}^{-*} - \mathbf{c} = -\frac{1}{\lambda}\mathbf{Q}^{-1}(\mathbf{g}^+ - \boldsymbol{\mu}^+). \tag{B.2.3}$$

Substituting (B.2.3) into (B.2.1),

$$\begin{aligned} L &= (\mathbf{g}^+ - \boldsymbol{\mu}^+)^T(\mathbf{b}^{+*} - \mathbf{b}^{-*}) + \frac{1}{2\lambda}(\mathbf{g}^+ - \boldsymbol{\mu}^+)^T\mathbf{Q}^{-1}(\mathbf{g}^+ - \boldsymbol{\mu}^+) - \frac{\lambda\gamma}{2} \\ &= (\mathbf{g}^+ - \boldsymbol{\mu}^+)^T\left(\mathbf{c} - \frac{1}{\lambda}\mathbf{Q}^{-1}(\mathbf{g}^+ - \boldsymbol{\mu}^+)\right) + \frac{1}{2\lambda}(\mathbf{g}^+ - \boldsymbol{\mu}^+)^T\mathbf{Q}^{-1}(\mathbf{g}^+ - \boldsymbol{\mu}^+) - \frac{\lambda\gamma}{2} \\ &= \mathbf{c}^T(\mathbf{g}^+ - \boldsymbol{\mu}^+) - \frac{1}{2\lambda}(\mathbf{g}^+ - \boldsymbol{\mu}^+)^T\mathbf{Q}^{-1}(\mathbf{g}^+ - \boldsymbol{\mu}^+) - \frac{\lambda\gamma}{2}. \end{aligned}$$

We have now reduced (B.2.2) to

$$\begin{aligned} \max_{\lambda,\boldsymbol{\mu}^+,\boldsymbol{\mu}^-} \quad & \mathbf{c}^T(\mathbf{g}^+ - \boldsymbol{\mu}^+) - \frac{1}{2\lambda}(\mathbf{g}^+ - \boldsymbol{\mu}^+)^T\mathbf{Q}^{-1}(\mathbf{g}^+ - \boldsymbol{\mu}^+) - \frac{\lambda\gamma}{2} \\ \text{s.t.} \quad & \mathbf{g}^+ - \boldsymbol{\mu}^+ + \mathbf{g}^- - \boldsymbol{\mu}^- = \mathbf{0}, \\ & \lambda > 0, \quad \boldsymbol{\mu}^+ \geq \mathbf{0}, \quad \boldsymbol{\mu}^- \geq \mathbf{0}. \end{aligned}$$

Making the change of variables $\boldsymbol{\mu} = \mathbf{g}^+ - \boldsymbol{\mu}^+$, this can be rewritten as

$$\begin{aligned} \max_{\lambda,\boldsymbol{\mu}} \quad & \mathbf{c}^T\boldsymbol{\mu} - \frac{1}{2\lambda}\boldsymbol{\mu}^T\mathbf{Q}^{-1}\boldsymbol{\mu} - \frac{\lambda\gamma}{2} \\ \text{s.t.} \quad & -\mathbf{g}^- \leq \boldsymbol{\mu} \leq \mathbf{g}^+, \\ & \lambda > 0. \end{aligned} \tag{B.2.4}$$

The maximization over $\lambda$ can be solved independently while holding $\boldsymbol{\mu}$ fixed, resulting in

$$\lambda^* = \sqrt{\gamma^{-1}\boldsymbol{\mu}^T\mathbf{Q}^{-1}\boldsymbol{\mu}}.$$

Substituting $\lambda = \lambda^*$ in (B.2.4) yields the final form in (3.3.9).

## B.3 Proof of correctness for the best-case linear relaxation examples

We wish to show that in the best-case examples constructed in Section 3.3.2, the lower bound on (2.0.1) resulting from the linear relaxation is equal to $N/2$ for $N$ even and $(N-1)/2$ for $N$ odd. First we calculate $B_n^+$ and $B_n^-$ using (3.3.7). The inverse of $\mathbf{Q}$ is given by

$$\mathbf{Q}^{-1} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{V}_\perp \end{bmatrix} \begin{bmatrix} \frac{1}{\lambda_1} & \\ & \frac{1}{\lambda_2}\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{v}_1^T \\ \mathbf{V}_\perp^T \end{bmatrix}. \tag{B.3.1}$$

Considering just the diagonal elements,

$$\left(\mathbf{Q}^{-1}\right)_{nn} = \frac{1}{\lambda_1}(\mathbf{v}_1)_n^2 + \frac{1}{\lambda_2}\left\|\mathbf{w}_n\right\|_2^2 = 1 + \frac{1}{\lambda_2}\left\|\mathbf{w}_n\right\|_2^2,$$

where $\mathbf{w}_n$ represents the $n$th row of $\mathbf{V}_\perp$. The equality of diagonal entries in (3.3.11) implies

$$\|\mathbf{w}_n\|_2^2 = 1 - (\mathbf{v}_1)_n^2 = \frac{N-1}{N}.$$

Hence

$$\left(\mathbf{Q}^{-1}\right)_{nn} = 1 + \frac{N-1}{N\lambda_2}$$

and given that $\mathbf{c} = \mathbf{e}$ and $\gamma = 1$,

$$B_n^+ = \sqrt{1 + \frac{N-1}{N\lambda_2}} + 1 = B^+, \tag{B.3.2a}$$

$$B_n^- = \sqrt{1 + \frac{N-1}{N\lambda_2}} - 1 = B^-. \tag{B.3.2b}$$

For the case of even $N$, we determine the optimal value of the linear relaxation directly, specifically by exhibiting feasible solutions to the primal (3.3.8) and to the dual (3.3.9) that have the same objective value. For the primal, let $\mathbf{b}_\ell = \mathbf{c} - (1/\sqrt{N\lambda_2})\mathbf{e}$. Since $\mathbf{v}_1$ and $\mathbf{e}$ are orthogonal, the eigendecomposition in (3.3.10) implies that $\mathbf{e}$ is an eigenvector of $\mathbf{Q}$ with eigenvalue $\lambda_2$. It follows from substituting $\mathbf{b} = \mathbf{b}_\ell$ into (2.1.1) that $\mathbf{b}_\ell$ is a feasible solution to the primal. The corresponding objective value is

$$\sum_{n=0}^{N-1} \frac{c_n - 1/\sqrt{N\lambda_2}}{B^+} = \frac{1}{B^+}\left(N - \sqrt{\frac{N}{\lambda_2}}\right), \tag{B.3.3}$$

assuming that $\lambda_2$ is large enough for all components of $\mathbf{b}_\ell$ to be positive. For the dual (3.3.9), let $\boldsymbol{\mu}_\ell = (1/B^+)\mathbf{e}$, which is equal to $\mathbf{g}^+$ by definition and hence feasible. Using the fact that $\mathbf{e}$ is also an eigenvector of $\mathbf{Q}^{-1}$ with eigenvalue $1/\lambda_2$, the dual objective value corresponding to $\boldsymbol{\mu}_\ell$ is

$$\frac{1}{B^+}\mathbf{c}^T\mathbf{e} - \sqrt{\frac{1}{(B^+)^2}\mathbf{e}^T\mathbf{Q}^{-1}\mathbf{e}} = \frac{1}{B^+}\left(N - \sqrt{\frac{N}{\lambda_2}}\right),$$

which is equal to the primal objective value in (B.3.3). We conclude that the optimal value of the linear relaxation is given by (B.3.3). As $\lambda_2$ increases to infinity, $B^+$ in (B.3.2a) converges to 2 and the optimal value of the linear relaxation approaches $N/2$ from below. Thus for sufficiently large $\lambda_2$, the optimal value of the linear relaxation is strictly greater than $N/2 - 1$, yielding a lower bound on (2.0.1) equal to $N/2$ after rounding up to the next integer.

When $N$ is odd, the proof above does not apply directly because the vectors $\mathbf{v}_1$ and $\mathbf{e}$ are no longer orthogonal and $\mathbf{e}$ is no longer an eigenvector of $\mathbf{Q}$ and $\mathbf{Q}^{-1}$. Thus the solutions $\mathbf{b}_\ell = \mathbf{c} - (1/\sqrt{N\lambda_2})\mathbf{e}$ and $\boldsymbol{\mu}_\ell = (1/B^+)\mathbf{e}$ are not necessarily optimal for (3.3.8) and (3.3.9) respectively. Instead, given that $\mathbf{v}_1^T\mathbf{e} = 1/\sqrt{N}$, we may express $\mathbf{e}$ as

$$\mathbf{e} = \frac{1}{\sqrt{N}}\mathbf{v}_1 + \sqrt{N - \frac{1}{N}}\mathbf{v}_\perp,$$

where $\mathbf{v}_\perp$ is a unit-norm vector orthogonal to $\mathbf{v}_1$, i.e., $\mathbf{v}_\perp$ is in the span of $\mathbf{V}_\perp$. Hence

$$\begin{aligned}
\mathbf{e}^T\mathbf{Q}^{-1}\mathbf{e} &= \left(\frac{1}{\sqrt{N}}\mathbf{v}_1 + \sqrt{N - \frac{1}{N}}\mathbf{v}_\perp\right)^T \begin{bmatrix}\mathbf{v}_1 & \mathbf{V}_\perp\end{bmatrix} \begin{bmatrix}\frac{1}{\lambda_1} & \\ & \frac{1}{\lambda_2}\mathbf{I}\end{bmatrix} \begin{bmatrix}\mathbf{v}_1^T \\ \mathbf{V}_\perp^T\end{bmatrix} \left(\frac{1}{\sqrt{N}}\mathbf{v}_1 + \sqrt{N - \frac{1}{N}}\mathbf{v}_\perp\right) \\
&= \begin{bmatrix}\frac{1}{\sqrt{N}} & \sqrt{N - \frac{1}{N}}\mathbf{v}_\perp^T\mathbf{V}_\perp\end{bmatrix} \begin{bmatrix}\frac{1}{\lambda_1} & \\ & \frac{1}{\lambda_2}\mathbf{I}\end{bmatrix} \begin{bmatrix}\frac{1}{\sqrt{N}} \\ \sqrt{N - \frac{1}{N}}\mathbf{V}_\perp^T\mathbf{v}_\perp\end{bmatrix} \\
&= \frac{1}{N\lambda_1} + \left(N - \frac{1}{N}\right)\frac{1}{\lambda_2}\left\|\mathbf{V}_\perp^T\mathbf{v}_\perp\right\|_2^2 \\
&= 1 + \left(N - \frac{1}{N}\right)\frac{1}{\lambda_2},
\end{aligned}$$

where the equality $\left\|\mathbf{V}_\perp^T\mathbf{v}_\perp\right\|_2^2 = 1$ can be deduced from (3.3.11), the orthogonality of $\mathbf{v}_\perp$ and $\mathbf{v}_1$, and the assumption that $\mathbf{v}_\perp$ has unit 2-norm. The dual objective value corresponding to $\boldsymbol{\mu}_\ell = (1/B^+)\mathbf{e}$ is therefore

$$\frac{1}{B^+}\left(N - \sqrt{1 + \left(N - \frac{1}{N}\right)\frac{1}{\lambda_2}}\right). \tag{B.3.4}$$

Although $\boldsymbol{\mu}_\ell$ may not be an optimal solution to (3.3.9), it is still a feasible solution and consequently the quantity in (B.3.4) is a lower bound on the optimal value of the linear relaxation. As $\lambda_2$ increases to infinity, this lower bound approaches $(N-1)/2$ from below. Thus for sufficiently large $\lambda_2$, the optimal value of the linear relaxation must be strictly larger than $(N-1)/2-1$ and the lower bound on (2.0.1) that it yields is equal to $(N-1)/2$ after rounding up.

## B.4 Proof of correctness for the worst-case linear relaxation examples

For the worst-case examples constructed in Section 3.3.2, we wish to show that the optimal value of (2.0.1) is equal to $N-1$ and the optimal value of the linear relaxation is less than 1. To prove the first statement, we make use of the eigendecomposition of $\mathbf{Q}^{-1}$ in (B.3.1). A calculation identical to that in Appendix B.3 shows that

$$\left(\mathbf{Q}^{-1}\right)_{nn} = \frac{1}{N\lambda_1} + \frac{N-1}{N\lambda_2}.$$

Substituting in $\lambda_1 = 1/(N-1)$ and $\lambda_2 = (N-1)/2$ gives $\left(\mathbf{Q}^{-1}\right)_{nn} = (N+1)/N > 1$. Given that $\mathbf{c} = \mathbf{e}$ and $\gamma = 1$, this implies that (2.3.3) is satisfied for all $n$ and the minimum zero-norm in (2.0.1) is no greater than $N-1$. To show that the minimum zero-norm is no less than $N-1$, we verify that (2.2.3) is violated for every subset $\mathcal{Z}$ of size 2, i.e., that no feasible combination of two zero-valued coefficients exists. Specializing (2.2.3) to the case at hand, we obtain

$$\begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} \left(\mathbf{Q}^{-1}\right)_{mm} & \left(\mathbf{Q}^{-1}\right)_{mn} \\ \left(\mathbf{Q}^{-1}\right)_{nm} & \left(\mathbf{Q}^{-1}\right)_{nn} \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \leq 1 \tag{B.4.1}$$

for arbitrary indices $m$ and $n$. By symmetry, $\left(\mathbf{Q}^{-1}\right)_{mn} = \left(\mathbf{Q}^{-1}\right)_{nm}$, and from (B.3.1),

$$\left(\mathbf{Q}^{-1}\right)_{mn} = \frac{1}{\lambda_1}(\mathbf{v}_1)_m(\mathbf{v}_1)_n + \frac{1}{\lambda_2}\mathbf{w}_m^T\mathbf{w}_n = \frac{N-1}{N} + \frac{2}{N-1}\mathbf{w}_m^T\mathbf{w}_n.$$

From the off-diagonal entries of (3.3.11) we have

$$\mathbf{w}_m^T \mathbf{w}_n = -(\mathbf{v}_1)_m (\mathbf{v}_1)_n = -\frac{1}{N}.$$

Hence (B.4.1) becomes

$$\begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} \frac{N+1}{N} & \frac{1}{N}\left(N-1-\frac{2}{N-1}\right) \\ \frac{1}{N}\left(N-1-\frac{2}{N-1}\right) & \frac{N+1}{N} \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \leq 1.$$

The left-hand side reduces to

$$\frac{N}{N-\frac{1}{N-1}},$$

which is strictly greater than 1, thus violating (B.4.1). We conclude that the minimum zero-norm in (2.0.1) is equal to $N-1$.

It remains to show that the optimal value of the linear relaxation is less than 1. Using a proof identical to the one in Appendix B.3, it can be seen that $\mathbf{b}_\ell = \mathbf{c} - (1/\sqrt{N\lambda_1})\mathbf{e}$ and $\boldsymbol{\mu}_\ell = (1/B^+)\mathbf{e}$ are optimal solutions to the primal (3.3.8) and the dual (3.3.9), where $B^+$ is now given by

$$B^+ = \sqrt{\left(\mathbf{Q}^{-1}\right)_{nn}} + 1 = \sqrt{\frac{N+1}{N}} + 1.$$

The corresponding optimal value is

$$\frac{1}{B^+}\left(N - \sqrt{\frac{N}{\lambda_1}}\right) = \frac{N-\sqrt{N(N-1)}}{\sqrt{\frac{N+1}{N}}+1}. \tag{B.4.2}$$

The denominator is clearly greater than 2 and it is straightforward to verify that the numerator is less than 1 for any $N$. Thus the lower bound on (2.0.1) that results from taking the ceiling of (B.4.2) is equal to 1.

## B.5 Proof of Lemma 2

To derive the desired bounds on $K^*$, we first obtain bounds on $E_0(K)$, beginning with a lower bound. Using the second definition of the Schur complement in (2.2.4), we observe that for each subset $\mathcal{Z}$ of size $K$,

$$\mathbf{c}_{\mathcal{Z}}^T(\mathbf{Q}/\mathbf{Q}_{\mathcal{YY}})\mathbf{c}_{\mathcal{Z}} = \mathbf{c}_{\mathcal{Z}}^T\left((\mathbf{Q}^{-1})_{\mathcal{ZZ}}\right)^{-1}\mathbf{c}_{\mathcal{Z}} \geq \lambda_{\min}\left(\left((\mathbf{Q}^{-1})_{\mathcal{ZZ}}\right)^{-1}\right)\|\mathbf{c}_{\mathcal{Z}}\|_2^2 = \lambda_{\max}^{-1}\left((\mathbf{Q}^{-1})_{\mathcal{ZZ}}\right)\|\mathbf{c}_{\mathcal{Z}}\|_2^2.$$

Given that

$$\lambda_{\max}\left((\mathbf{Q}^{-1})_{\mathcal{ZZ}}\right) \leq \lambda_{\max}(\mathbf{Q}^{-1}),$$

it follows that

$$\mathbf{c}_{\mathcal{Z}}^T(\mathbf{Q}/\mathbf{Q}_{\mathcal{YY}})\mathbf{c}_{\mathcal{Z}} \geq \lambda_{\max}^{-1}(\mathbf{Q}^{-1})\|\mathbf{c}_{\mathcal{Z}}\|_2^2 = \lambda_{\min}(\mathbf{Q})\|\mathbf{c}_{\mathcal{Z}}\|_2^2$$

for all $\mathcal{Z}$ of size $K$. Hence

$$E_0(K) \geq \min_{|\mathcal{Z}|=K}\lambda_{\min}(\mathbf{Q})\|\mathbf{c}_{\mathcal{Z}}\|_2^2 = \lambda_{\min}(\mathbf{Q})\Sigma_K\big(\{c_n^2\}\big). \tag{B.5.1}$$

An upper bound on $E_0(K)$ can be derived in a similar manner. For each $\mathcal{Z}$ of size $K$ we have

$$\mathbf{c}_{\mathcal{Z}}^T(\mathbf{Q}/\mathbf{Q}_{\mathcal{YY}})\mathbf{c}_{\mathcal{Z}} \leq \lambda_{\max}\left(\left((\mathbf{Q}^{-1})_{\mathcal{ZZ}}\right)^{-1}\right)\|\mathbf{c}_{\mathcal{Z}}\|_2^2 \leq \lambda_{\max}(\mathbf{Q})\|\mathbf{c}_{\mathcal{Z}}\|_2^2,$$

and therefore

$$E_0(K) \leq \min_{|\mathcal{Z}|=K}\lambda_{\max}(\mathbf{Q})\|\mathbf{c}_{\mathcal{Z}}\|_2^2 = \lambda_{\max}(\mathbf{Q})\Sigma_K\big(\{c_n^2\}\big). \tag{B.5.2}$$

The lower bound on $E_0(K)$ in (B.5.1) implies that $\overline{K}$ defined in (3.4.9) is an upper bound on $K^*$. Likewise from (B.5.2), $\underline{K}$ in (3.4.10) is a lower bound on $K^*$. To obtain the bound on the ratio $\overline{K}/\underline{K}$, we infer from the definition of $\underline{K}$ in (3.4.10) that $\lambda_{\max}(\mathbf{Q})\Sigma_{\underline{K}+1}\big(\{c_n^2\}\big) > \gamma$. This can be rewritten in terms of $\lambda_{\min}(\mathbf{Q})$ as

$$\lambda_{\min}(\mathbf{Q})\kappa(\mathbf{Q})\Sigma_{\underline{K}+1}\big(\{c_n^2\}\big) > \gamma.$$

Furthermore,

$$\kappa(\mathbf{Q})\Sigma_{\underline{K}+1}\big(\{c_n^2\}\big) \leq \frac{\lceil(\underline{K}+1)\kappa(\mathbf{Q})\rceil}{\underline{K}+1}\Sigma_{\underline{K}+1}\big(\{c_n^2\}\big) \leq \Sigma_{\lceil(\underline{K}+1)\kappa(\mathbf{Q})\rceil}\big(\{c_n^2\}\big),$$

where the second inequality is due to the average of the $(\underline{K}+1)$ smallest elements in a sequence being smaller than the average of the $\lceil(\underline{K}+1)\kappa(\mathbf{Q})\rceil$ smallest elements, given

$\kappa(\mathbf{Q}) \geq 1$. Combining the last two lines of inequalities,

$$\lambda_{\min}(\mathbf{Q})\Sigma_{\lceil(\underline{K}+1)\kappa(\mathbf{Q})\rceil}\big(\{c_n^2\}\big) > \gamma.$$

It follows from the definition of $\overline{K}$ that $\overline{K} \leq \lceil(\underline{K}+1)\kappa(\mathbf{Q})\rceil - 1$.

## B.6 Proof of Lemma 4

We expand $\mathbf{\Lambda}^{-1/2}\mathbf{Q}\mathbf{\Lambda}^{-1/2}$ as

$$\begin{aligned}\mathbf{\Lambda}^{-1/2}\mathbf{Q}\mathbf{\Lambda}^{-1/2} &= \mathbf{\Lambda}^{-1/2}(\mathbf{I}+\mathbf{\Delta})\mathbf{\Lambda}(\mathbf{I}+\mathbf{\Delta})^T\mathbf{\Lambda}^{-1/2} \\ &= \mathbf{I} + \underbrace{\mathbf{\Lambda}^{-1/2}\mathbf{\Delta}\mathbf{\Lambda}^{1/2}}_{\widetilde{\mathbf{\Delta}}} + \underbrace{\mathbf{\Lambda}^{1/2}\mathbf{\Delta}^T\mathbf{\Lambda}^{-1/2}}_{\widetilde{\mathbf{\Delta}}^T} + \underbrace{\mathbf{\Lambda}^{-1/2}\mathbf{\Delta}\mathbf{\Lambda}^{1/2}\mathbf{\Lambda}^{1/2}\mathbf{\Delta}^T\mathbf{\Lambda}^{-1/2}}_{\widetilde{\mathbf{\Delta}}\widetilde{\mathbf{\Delta}}^T}.\end{aligned}$$

Then

$$\begin{aligned}\lambda_{\min}\big(\mathbf{\Lambda}^{-1/2}\mathbf{Q}\mathbf{\Lambda}^{-1/2}\big) &= 1 + \lambda_{\min}\big(\widetilde{\mathbf{\Delta}} + \widetilde{\mathbf{\Delta}}^T + \widetilde{\mathbf{\Delta}}\widetilde{\mathbf{\Delta}}^T\big) \\ &\geq 1 + \lambda_{\min}\big(\widetilde{\mathbf{\Delta}} + \widetilde{\mathbf{\Delta}}^T\big) \qquad \text{(B.6.1)}\end{aligned}$$

since $\widetilde{\mathbf{\Delta}}\widetilde{\mathbf{\Delta}}^T$ is positive semidefinite. Similarly

$$\begin{aligned}\lambda_{\max}\big(\mathbf{\Lambda}^{-1/2}\mathbf{Q}\mathbf{\Lambda}^{-1/2}\big) &= 1 + \lambda_{\max}\big(\widetilde{\mathbf{\Delta}} + \widetilde{\mathbf{\Delta}}^T + \widetilde{\mathbf{\Delta}}\widetilde{\mathbf{\Delta}}^T\big) \\ &\leq 1 + \lambda_{\max}\big(\widetilde{\mathbf{\Delta}} + \widetilde{\mathbf{\Delta}}^T\big) + \lambda_{\max}\big(\widetilde{\mathbf{\Delta}}\widetilde{\mathbf{\Delta}}^T\big). \qquad \text{(B.6.2)}\end{aligned}$$

We proceed to show that $\lambda_{\min}\big(\widetilde{\mathbf{\Delta}} + \widetilde{\mathbf{\Delta}}^T\big)$ is bounded from below by $-\kappa(\mathbf{Q})\rho(\mathbf{\Delta})$ and $\lambda_{\max}\big(\widetilde{\mathbf{\Delta}} + \widetilde{\mathbf{\Delta}}^T\big)$ is bounded from above by $\kappa(\mathbf{Q})\rho(\mathbf{\Delta})$. First we determine the set of possible locations in the complex plane for the eigenvalues of $\widetilde{\mathbf{\Delta}}$. Because $\widetilde{\mathbf{\Delta}}$ and $\mathbf{\Delta}$ are related by a similarity transformation, they share the same set of eigenvalues. Since $\mathbf{V}$ is an orthogonal matrix, its eigenvalues are located on the unit circle centered at the origin, and thus the eigenvalues of $\mathbf{\Delta}$ are located on the unit circle centered at $-1$. The assumption that $\rho(\mathbf{\Delta})$ is small compared to 1 further restricts the eigenvalues of $\mathbf{\Delta}$ to lie on a small arc near the origin as depicted in Fig. B-1.

To relate the eigenvalues of $\widetilde{\mathbf{\Delta}} + \widetilde{\mathbf{\Delta}}^T$ to those of $\widetilde{\mathbf{\Delta}}$, we make use of a diagonalization of $\widetilde{\mathbf{\Delta}}$. Given that $\mathbf{V}$ is orthogonal, both $\mathbf{V}$ and $\mathbf{\Delta}$ are normal matrices and $\mathbf{\Delta}$ can be

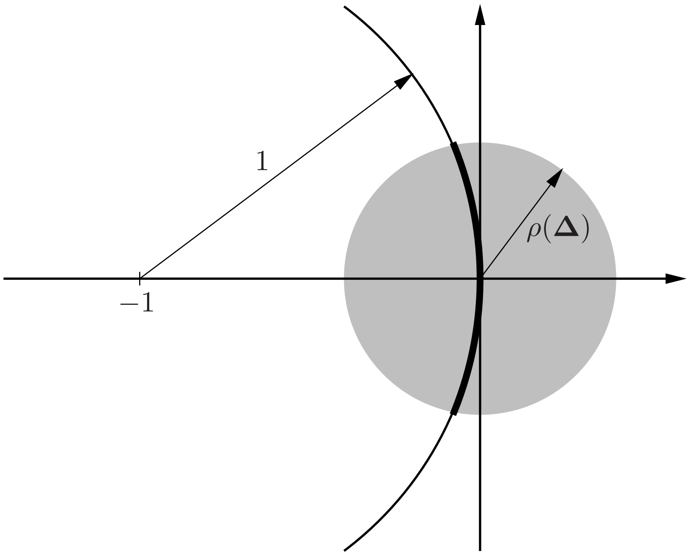


Figure B-1: The set of possible locations (dark segment of arc) in the complex plane for the eigenvalues of $\mathbf{\Delta}$ and $\widetilde{\mathbf{\Delta}}$.

diagonalized as $\mathbf{\Delta} = \mathbf{U}\mathbf{\Psi}\mathbf{U}^H$, where $\mathbf{U}$ is a unitary matrix and $\mathbf{\Psi}$ is a complex diagonal matrix. Then

$$\widetilde{\mathbf{\Delta}} = \underbrace{\mathbf{\Lambda}^{-1/2}\mathbf{U}}_{\widetilde{\mathbf{U}}}\,\mathbf{\Psi}\,\underbrace{\mathbf{U}^H\mathbf{\Lambda}^{1/2}}_{\widetilde{\mathbf{U}}^{-1}}. \tag{B.6.3}$$

Using (B.6.3) and a theorem from [92], it follows that for any eigenvalue of $\widetilde{\mathbf{\Delta}} + \widetilde{\mathbf{\Delta}}^T$, there exists an eigenvalue $\lambda_0$ of $\widetilde{\mathbf{\Delta}}$ such that

$$\left|\lambda(\widetilde{\mathbf{\Delta}} + \widetilde{\mathbf{\Delta}}^T) - \lambda_0\right| \leq \left\|\widetilde{\mathbf{U}}^{-1}\widetilde{\mathbf{\Delta}}^T\widetilde{\mathbf{U}}\right\|_2, \tag{B.6.4}$$

where $\|\mathbf{A}\|_2$ denotes the spectral norm of the matrix $\mathbf{A}$. Expanding the right-hand side of (B.6.4) and using the sub-multiplicative property of matrix norms,

$$\begin{aligned}\left|\lambda(\widetilde{\mathbf{\Delta}} + \widetilde{\mathbf{\Delta}}^T) - \lambda_0\right| &\leq \left\|\mathbf{U}^H\mathbf{\Lambda}^{1/2}\mathbf{\Lambda}^{1/2}\mathbf{U}\mathbf{\Psi}^H\mathbf{U}^H\mathbf{\Lambda}^{-1/2}\mathbf{\Lambda}^{-1/2}\mathbf{U}\right\|_2 \\ &\leq \left\|\mathbf{U}^H\mathbf{\Lambda}\mathbf{U}\right\|_2 \left\|\mathbf{\Psi}^H\right\|_2 \left\|\mathbf{U}^H\mathbf{\Lambda}^{-1}\mathbf{U}\right\|_2 \\ &= \lambda_{\max}(\mathbf{Q})\rho(\mathbf{\Delta})\lambda_{\min}^{-1}(\mathbf{Q}) \\ &= \kappa(\mathbf{Q})\rho(\mathbf{\Delta}), \end{aligned} \tag{B.6.5}$$

where the second-to-last equality is due to the equivalence between the spectral norm and the spectral radius for normal matrices.

Equation (B.6.5) implies that $\lambda_{\text{min}}\big(\widetilde{\mathbf{\Delta}}+\widetilde{\mathbf{\Delta}}^T\big)$ and $\lambda_{\text{max}}\big(\widetilde{\mathbf{\Delta}}+\widetilde{\mathbf{\Delta}}^T\big)$ lie within a Euclidean distance of $\kappa(\mathbf{Q})\rho(\mathbf{\Delta})$ from the arc in Fig. B-1. Furthermore, $\lambda_{\text{min}}\big(\widetilde{\mathbf{\Delta}}+\widetilde{\mathbf{\Delta}}^T\big)$ and $\lambda_{\text{max}}\big(\widetilde{\mathbf{\Delta}}+\widetilde{\mathbf{\Delta}}^T\big)$ must be real because $\widetilde{\mathbf{\Delta}}+\widetilde{\mathbf{\Delta}}^T$ is symmetric. It is clear then that $\lambda_{\text{max}}\big(\widetilde{\mathbf{\Delta}}+\widetilde{\mathbf{\Delta}}^T\big)$ can be no greater than $+\kappa(\mathbf{Q})\rho(\mathbf{\Delta})$. Given the assumption that $\kappa(\mathbf{Q})\rho(\mathbf{\Delta}) < 1$, it is also apparent from Fig. B-1 that $-\kappa(\mathbf{Q})\rho(\mathbf{\Delta})$ is the minimum possible value for $\lambda_{\text{min}}\big(\widetilde{\mathbf{\Delta}}+\widetilde{\mathbf{\Delta}}^T\big)$, corresponding to setting $\lambda_0 = 0$ in (B.6.5). All other points on the arc of possible locations for eigenvalues of $\widetilde{\mathbf{\Delta}}$ are farther away from the point $-\kappa(\mathbf{Q})\rho(\mathbf{\Delta})$ than $\lambda_0 = 0$. Combining (B.6.1) and the lower bound of $-\kappa(\mathbf{Q})\rho(\mathbf{\Delta})$ on $\lambda_{\text{min}}\big(\widetilde{\mathbf{\Delta}}+\widetilde{\mathbf{\Delta}}^T\big)$ proves the bound on $\lambda_{\text{min}}\big(\mathbf{\Lambda}^{-1/2}\mathbf{Q}\mathbf{\Lambda}^{-1/2}\big)$.

To complete the proof of the bound on $\lambda_{\text{max}}\big(\mathbf{\Lambda}^{-1/2}\mathbf{Q}\mathbf{\Lambda}^{-1/2}\big)$, it remains to bound the last term in (B.6.2) as the middle term is now bounded by $\kappa(\mathbf{Q})\rho(\mathbf{\Delta})$. Using the definition of the spectral norm and the sub-multiplicative property,

$$\begin{aligned}\lambda_{\text{max}}\big(\widetilde{\mathbf{\Delta}}\widetilde{\mathbf{\Delta}}^T\big) = \left\|\widetilde{\mathbf{\Delta}}^T\right\|_2^2 &= \left\|\mathbf{\Lambda}^{1/2}\mathbf{\Delta}^T\mathbf{\Lambda}^{-1/2}\right\|_2^2 \\ &\leq \left\|\mathbf{\Lambda}^{1/2}\right\|_2^2 \left\|\mathbf{\Delta}^T\right\|_2^2 \left\|\mathbf{\Lambda}^{-1/2}\right\|_2^2 \\ &= \kappa(\mathbf{Q})\rho^2(\mathbf{\Delta}).\end{aligned}$$

# Appendix C

# Derivations for Chapter 5

## C.1 Scalar CSD quantization given a fixed number of SPTs

This appendix discusses the CSD quantization of scalars under a constraint on the number of SPTs (problem (5.2.7)). Our objective is to find the closest integer-valued approximation to a real number $c$ where the approximation is of the form

$$b = \sum_{p=0}^{P-1} s_p 2^p, \quad s_p \in \{-1, 0, +1\}$$

with at most $B$ of the digits $s_p$ being non-zero. Furthermore, in a CSD representation, no two non-zero digits can be adjacent. This non-adjacency property restricts $B$ to be no greater than $\lceil P/2 \rceil$.

The quantization can be performed recursively, first by determining the position $p_1$ of the most significant non-zero digit, subtracting the corresponding SPT from $c$, and then repeating with the residual. More specifically, once $p_1$ is determined, we set $s_{p_1} = \operatorname{sgn}(c)$, update the parameters as follows:

$$\hat{c} = c - \operatorname{sgn}(c) 2^{p_1}, \quad \hat{B} = B - 1, \quad \hat{P} = p_1 - 1,$$

and then proceed in the same way with the next most significant non-zero digit. The update to $P$ reflects the fact that the next highest power of two can be at most $p_1 - 2$ due to the selection of $p_1$. The process continues until either $\hat{B} = 0$ or $\hat{P} \leq 0$, for a maximum of $\lceil P/2 \rceil$ iterations.

To determine $p_1$ as a function of $c$, $B$, and $P$, we can equivalently specify the interval of real numbers for which choosing $p_1$ as the highest power of two will result in the best approximation. We may restrict attention to the case $c \geq 0$ since the sign of $c$ affects only the sign of $s_{p_1}$. Then $p_1$ is determined by the interval in which $c$ falls. Note that in addition to the values $0, 1, \ldots, P-1$, $p_1$ can also equal $-\infty$, i.e., the best approximation to $c$ is $b = 0$.

First we determine the range of integers that can be represented exactly when $p_1$ is the highest power of two. With $s_{p_1} = +1$, the largest possible integer $b_{\max}(p_1, B)$ is attained by alternating 0's and $+1$'s after $s_{p_1}$, while the smallest integer $b_{\min}(p_1, B)$ is attained by alternating 0's and $-1$'s. The number of non-zero digits occurring after $s_{p_1}$ is at most $B' = \min\{B-1, \lfloor p_1/2 \rfloor\}$. Hence

$$b_{\max}(p_1, B) = 2^{p_1}\left(1 + \sum_{p=1}^{B'} \left(\frac{1}{4}\right)^p\right) = 2^{p_1}\left(\frac{4}{3} - \frac{1}{3}\left(\frac{1}{4}\right)^{B'}\right),$$
$$b_{\min}(p_1, B) = 2^{p_1}\left(1 - \sum_{p=1}^{B'} \left(\frac{1}{4}\right)^p\right) = 2^{p_1}\left(\frac{2}{3} + \frac{1}{3}\left(\frac{1}{4}\right)^{B'}\right).$$

Since

$$b_{\min}(p_1 + 1, B) = 2^{p_1}\left(\frac{4}{3} + \frac{2}{3}\left(\frac{1}{4}\right)^{B''}\right) > b_{\max}(p_1, B)$$

where $B'' = \min\{B-1, \lfloor (p_1+1)/2 \rfloor\}$, the range of integers with $p_1$ as the highest power does not overlap with the corresponding range for $p_1+1$. The boundary between the quantization intervals for $p_1$ and $p_1+1$ is given by the midpoint between $b_{\max}(p_1, B)$ and $b_{\min}(p_1+1, B)$. After some straightforward calculations, we obtain

$$\frac{b_{\max}(p_1, B) + b_{\min}(p_1 + 1, B)}{2} = \begin{cases} 2^{p_1}\left(\frac{4}{3} + \frac{1}{6}\left(\frac{1}{4}\right)^{B-1}\right), & B - 1 \leq \left\lfloor \frac{p_1}{2} \right\rfloor, \\ 2^{p_1}\left(\frac{4}{3} + \frac{1}{6}\left(-\frac{1}{2}\right)^{p_1}\right), & B - 1 > \left\lfloor \frac{p_1}{2} \right\rfloor. \end{cases}$$

Similarly, the boundary between the quantization intervals for $p_1$ and $p_1 - 1$ is the midpoint between $b_{\max}(p_1 - 1, B)$ and $b_{\min}(p_1, B)$. In addition, there are two special cases: The boundary between $p_1 = -\infty$ and $p_1 = 0$ is $1/2$, corresponding to the choice between $b = 0$ and $b = 1$. For $p_1 = P-1$, there is no upper boundary and the quantization interval extends to $\infty$. This completes the specification of the quantization intervals for a single iteration.

## C.2 Derivation of subproblem parameters

In this appendix, we consider the subproblem that is created when some of the coefficients in either problem (5.1.1) or (5.1.2) are fixed. We show that the subproblem is also quadratically constrained and we relate the parameters $\mathbf{Q}_{\text{eff}}$, $\mathbf{c}_{\text{eff}}$, and $\gamma_{\text{eff}}$ for the subproblem to the original parameters $\mathbf{Q}$, $\mathbf{c}$, and $\gamma$.

Denote by $\mathcal{K}$ the subset of coefficients whose values have been fixed and by $\mathcal{F}$ the complementary subset. In terms of these two subsets, (2.1.1) can be rewritten as

$$(\mathbf{b}_{\mathcal{F}}-\mathbf{c}_{\mathcal{F}})^T\mathbf{Q}_{\mathcal{F}\mathcal{F}}(\mathbf{b}_{\mathcal{F}}-\mathbf{c}_{\mathcal{F}})+2(\mathbf{b}_{\mathcal{K}}-\mathbf{c}_{\mathcal{K}})\mathbf{Q}_{\mathcal{K}\mathcal{F}}(\mathbf{b}_{\mathcal{F}}-\mathbf{c}_{\mathcal{F}})+(\mathbf{b}_{\mathcal{K}}-\mathbf{c}_{\mathcal{K}})^T\mathbf{Q}_{\mathcal{K}\mathcal{K}}(\mathbf{b}_{\mathcal{K}}-\mathbf{c}_{\mathcal{K}})\leq\gamma, \tag{C.2.1}$$

which is quadratic in the variables $\mathbf{b}_{\mathcal{F}}$ when $\mathbf{b}_{\mathcal{K}}$ is held constant. From the term in (C.2.1) that is quadratic in $\mathbf{b}_{\mathcal{F}}$, we see that the subproblem parameter $\mathbf{Q}_{\text{eff}}$ is given by

$$\mathbf{Q}_{\text{eff}}=\mathbf{Q}_{\mathcal{F}\mathcal{F}}. \tag{C.2.2}$$

By partitioning $\mathbf{Q}^{-1}$ according to $\mathcal{K}$ and $\mathcal{F}$ and inverting the resulting $2\times 2$ block matrix, it can be shown that the corresponding relationship between $(\mathbf{Q}_{\text{eff}})^{-1}$ and $\mathbf{Q}^{-1}$ is

$$(\mathbf{Q}_{\text{eff}})^{-1}=\left(\mathbf{Q}^{-1}\right)_{\mathcal{F}\mathcal{F}}-\left(\mathbf{Q}^{-1}\right)_{\mathcal{F}\mathcal{K}}\left(\left(\mathbf{Q}^{-1}\right)_{\mathcal{K}\mathcal{K}}\right)^{-1}\left(\mathbf{Q}^{-1}\right)_{\mathcal{K}\mathcal{F}}. \tag{C.2.3}$$

The parameter $\mathbf{c}_{\text{eff}}$ is equal to the value of $\mathbf{b}_{\mathcal{F}}$ that minimizes the left-hand side of (C.2.1), just as $\mathbf{b}=\mathbf{c}$ minimizes the left-hand side of (2.1.1). A straightforward calculation results in

$$\mathbf{c}_{\text{eff}}=\mathbf{c}_{\mathcal{F}}-(\mathbf{Q}_{\mathcal{F}\mathcal{F}})^{-1}\mathbf{Q}_{\mathcal{F}\mathcal{K}}(\mathbf{b}_{\mathcal{K}}-\mathbf{c}_{\mathcal{K}}). \tag{C.2.4}$$

An alternative form for (C.2.4) is

$$\mathbf{c}_{\text{eff}}=\mathbf{c}_{\mathcal{F}}+\left(\mathbf{Q}^{-1}\right)_{\mathcal{F}\mathcal{K}}\left(\left(\mathbf{Q}^{-1}\right)_{\mathcal{K}\mathcal{K}}\right)^{-1}(\mathbf{b}_{\mathcal{K}}-\mathbf{c}_{\mathcal{K}}), \tag{C.2.5}$$

which can be derived in the same way as (C.2.3). Using (C.2.4), (C.2.1) can be rewritten as

$$(\mathbf{b}_{\mathcal{F}}-\mathbf{c}_{\text{eff}})^T\mathbf{Q}_{\text{eff}}(\mathbf{b}_{\mathcal{F}}-\mathbf{c}_{\text{eff}})\leq\gamma-(\mathbf{b}_{\mathcal{K}}-\mathbf{c}_{\mathcal{K}})^T(\mathbf{Q}/\mathbf{Q}_{\mathcal{F}\mathcal{F}})(\mathbf{b}_{\mathcal{K}}-\mathbf{c}_{\mathcal{K}}),$$

which shows that

$$\gamma_{\text{eff}} = \gamma - (\mathbf{b}_{\mathcal{K}} - \mathbf{c}_{\mathcal{K}})^T (\mathbf{Q}/\mathbf{Q}_{\mathcal{F}\mathcal{F}})(\mathbf{b}_{\mathcal{K}} - \mathbf{c}_{\mathcal{K}}). \tag{C.2.6}$$

Equations (C.2.2)–(C.2.6) give the desired expressions for the subproblem parameters.

In Section 5.3.2, we make use of the special case in which a single coefficient $b_m$ is fixed. With $\mathcal{K} = \{m\}$, (C.2.3), (C.2.5) and (C.2.6) become

$$(\mathbf{Q}_{\text{eff}})^{-1} = \left(\mathbf{Q}^{-1}\right)_{\mathcal{F}\mathcal{F}} - \frac{1}{\left(\mathbf{Q}^{-1}\right)_{mm}} \left(\mathbf{Q}^{-1}\right)_{\mathcal{F}m} \left(\mathbf{Q}^{-1}\right)_{m\mathcal{F}}, \tag{C.2.7a}$$

$$\mathbf{c}_{\text{eff}} = \mathbf{c}_{\mathcal{F}} + \frac{b_m - c_m}{\left(\mathbf{Q}^{-1}\right)_{mm}} \left(\mathbf{Q}^{-1}\right)_{\mathcal{F}m}, \tag{C.2.7b}$$

$$\gamma_{\text{eff}} = \gamma - \frac{(b_m - c_m)^2}{\left(\mathbf{Q}^{-1}\right)_{mm}}. \tag{C.2.7c}$$

Note that no matrix inversions are required in (C.2.7).

# Appendix D

# Derivations for Chapter 6

## D.1 Derivation of the dual of problem (6.3.6)

In this section, we derive the dual of problem (6.3.6), the linear relaxation of problem (5.1.1). The derivation is broadly similar to that in Appendix B.2.

We first introduce some definitions to facilitate the use of matrix notation. To express the objective function more compactly, we replace $b_n$ by $\widetilde{b}_n = b_n - \underline{B}_n$ for $n \in \mathcal{P}$ and by $\widetilde{b}_n = b_n - \overline{B}_n$ for $n \in \mathcal{N}$. The parameters $c_n$ are changed accordingly as shown in (6.3.8a). We also define the vectors $\mathbf{p}^+_{\mathcal{D}}$, $\mathbf{p}^-_{\mathcal{D}}$, $\mathbf{p}_{\mathcal{P}}$, and $\mathbf{p}_{\mathcal{N}}$ as in (6.3.8b) and (6.3.8c). With these definitions and neglecting the constant terms in the objective, (6.3.6) can be rewritten as

$$\min_{\mathbf{b}^+_{\mathcal{D}},\mathbf{b}^-_{\mathcal{D}},\widetilde{\mathbf{b}}_{\mathcal{P}},\widetilde{\mathbf{b}}_{\mathcal{N}}} \quad \begin{bmatrix} \mathbf{w}^+_{\mathcal{D}} \\ \mathbf{w}^-_{\mathcal{D}} \\ \mathbf{w}_{\mathcal{P}} \\ -\mathbf{w}_{\mathcal{N}} \end{bmatrix}^T \begin{bmatrix} \mathbf{b}^+_{\mathcal{D}} \\ \mathbf{b}^-_{\mathcal{D}} \\ \widetilde{\mathbf{b}}_{\mathcal{P}} \\ \widetilde{\mathbf{b}}_{\mathcal{N}} \end{bmatrix} \tag{D.1.1}$$

$$\text{s.t.} \quad \begin{bmatrix} \mathbf{b}^+_{\mathcal{D}} - \mathbf{c}_{\mathcal{D}} \\ \mathbf{b}^-_{\mathcal{D}} \\ \widetilde{\mathbf{b}}_{\mathcal{P}} - \widetilde{\mathbf{c}}_{\mathcal{P}} \\ \widetilde{\mathbf{b}}_{\mathcal{N}} - \widetilde{\mathbf{c}}_{\mathcal{N}} \end{bmatrix}^T \begin{bmatrix} \mathbf{Q}_{\mathcal{D}\mathcal{D}} & -\mathbf{Q}_{\mathcal{D}\mathcal{D}} & \mathbf{Q}_{\mathcal{D}\mathcal{P}} & \mathbf{Q}_{\mathcal{D}\mathcal{N}} \\ -\mathbf{Q}_{\mathcal{D}\mathcal{D}} & \mathbf{Q}_{\mathcal{D}\mathcal{D}} & -\mathbf{Q}_{\mathcal{D}\mathcal{P}} & -\mathbf{Q}_{\mathcal{D}\mathcal{N}} \\ \mathbf{Q}_{\mathcal{P}\mathcal{D}} & -\mathbf{Q}_{\mathcal{P}\mathcal{D}} & \mathbf{Q}_{\mathcal{P}\mathcal{P}} & \mathbf{Q}_{\mathcal{P}\mathcal{N}} \\ \mathbf{Q}_{\mathcal{N}\mathcal{D}} & -\mathbf{Q}_{\mathcal{N}\mathcal{D}} & \mathbf{Q}_{\mathcal{N}\mathcal{P}} & \mathbf{Q}_{\mathcal{N}\mathcal{N}} \end{bmatrix} \begin{bmatrix} \mathbf{b}^+_{\mathcal{D}} - \mathbf{c}_{\mathcal{D}} \\ \mathbf{b}^-_{\mathcal{D}} \\ \widetilde{\mathbf{b}}_{\mathcal{P}} - \widetilde{\mathbf{c}}_{\mathcal{P}} \\ \widetilde{\mathbf{b}}_{\mathcal{N}} - \widetilde{\mathbf{c}}_{\mathcal{N}} \end{bmatrix} \leq \gamma,$$

$$\mathbf{0} \leq \mathbf{b}^+_{\mathcal{D}} \leq \mathbf{p}^+_{\mathcal{D}}, \qquad \mathbf{0} \leq \mathbf{b}^-_{\mathcal{D}} \leq \mathbf{p}^-_{\mathcal{D}},$$

$$\mathbf{0} \leq \widetilde{\mathbf{b}}_{\mathcal{P}} \leq \mathbf{p}_{\mathcal{P}}, \qquad \mathbf{0} \leq -\widetilde{\mathbf{b}}_{\mathcal{N}} \leq \mathbf{p}_{\mathcal{N}}.$$

We assign the non-negative Lagrange multiplier $\lambda/2$ to the quadratic constraint in (D.1.1), $\boldsymbol{\mu}_{\mathcal{D}}^{+}$, $\boldsymbol{\mu}_{\mathcal{D}}^{-}$, $\boldsymbol{\mu}_{\mathcal{P}}$, and $\boldsymbol{\mu}_{\mathcal{N}}$ to the corresponding non-negativity and non-positivity constraints, and $\boldsymbol{\nu}_{\mathcal{D}}^{+}$, $\boldsymbol{\nu}_{\mathcal{D}}^{-}$, $\boldsymbol{\nu}_{\mathcal{P}}$, and $\boldsymbol{\nu}_{\mathcal{N}}$ to the constraints involving $\mathbf{p}_{\mathcal{D}}^{+}$, $\mathbf{p}_{\mathcal{D}}^{-}$, $\mathbf{p}_{\mathcal{P}}$, and $\mathbf{p}_{\mathcal{N}}$ respectively. The Lagrangian for (D.1.1) is then given by

$$
\begin{aligned}
L = \frac{\lambda}{2}
\begin{bmatrix} \mathbf{b}_{\mathcal{D}}^{+} - \mathbf{c}_{\mathcal{D}} \\ \mathbf{b}_{\mathcal{D}}^{-} \\ \widetilde{\mathbf{b}}_{\mathcal{P}} - \widetilde{\mathbf{c}}_{\mathcal{P}} \\ \widetilde{\mathbf{b}}_{\mathcal{N}} - \widetilde{\mathbf{c}}_{\mathcal{N}} \end{bmatrix}^{T}
&\begin{bmatrix} \mathbf{Q}_{\mathcal{D}\mathcal{D}} & -\mathbf{Q}_{\mathcal{D}\mathcal{D}} & \mathbf{Q}_{\mathcal{D}\mathcal{P}} & \mathbf{Q}_{\mathcal{D}\mathcal{N}} \\ -\mathbf{Q}_{\mathcal{D}\mathcal{D}} & \mathbf{Q}_{\mathcal{D}\mathcal{D}} & -\mathbf{Q}_{\mathcal{D}\mathcal{P}} & -\mathbf{Q}_{\mathcal{D}\mathcal{N}} \\ \mathbf{Q}_{\mathcal{P}\mathcal{D}} & -\mathbf{Q}_{\mathcal{P}\mathcal{D}} & \mathbf{Q}_{\mathcal{P}\mathcal{P}} & \mathbf{Q}_{\mathcal{P}\mathcal{N}} \\ \mathbf{Q}_{\mathcal{N}\mathcal{D}} & -\mathbf{Q}_{\mathcal{N}\mathcal{D}} & \mathbf{Q}_{\mathcal{N}\mathcal{P}} & \mathbf{Q}_{\mathcal{N}\mathcal{N}} \end{bmatrix}
\begin{bmatrix} \mathbf{b}_{\mathcal{D}}^{+} - \mathbf{c}_{\mathcal{D}} \\ \mathbf{b}_{\mathcal{D}}^{-} \\ \widetilde{\mathbf{b}}_{\mathcal{P}} - \widetilde{\mathbf{c}}_{\mathcal{P}} \\ \widetilde{\mathbf{b}}_{\mathcal{N}} - \widetilde{\mathbf{c}}_{\mathcal{N}} \end{bmatrix} \\
&+ \begin{bmatrix} \mathbf{w}_{\mathcal{D}}^{+} + \boldsymbol{\nu}_{\mathcal{D}}^{+} - \boldsymbol{\mu}_{\mathcal{D}}^{+} \\ \mathbf{w}_{\mathcal{D}}^{-} + \boldsymbol{\nu}_{\mathcal{D}}^{-} - \boldsymbol{\mu}_{\mathcal{D}}^{-} \\ \mathbf{w}_{\mathcal{P}} + \boldsymbol{\nu}_{\mathcal{P}} - \boldsymbol{\mu}_{\mathcal{P}} \\ -(\mathbf{w}_{\mathcal{N}} + \boldsymbol{\nu}_{\mathcal{N}} - \boldsymbol{\mu}_{\mathcal{N}}) \end{bmatrix}^{T}
\begin{bmatrix} \mathbf{b}_{\mathcal{D}}^{+} \\ \mathbf{b}_{\mathcal{D}}^{-} \\ \widetilde{\mathbf{b}}_{\mathcal{P}} \\ \widetilde{\mathbf{b}}_{\mathcal{N}} \end{bmatrix}
- \frac{\lambda\gamma}{2} -
\begin{bmatrix} \mathbf{p}_{\mathcal{D}}^{+} \\ \mathbf{p}_{\mathcal{D}}^{-} \\ \mathbf{p}_{\mathcal{P}} \\ \mathbf{p}_{\mathcal{N}} \end{bmatrix}^{T}
\begin{bmatrix} \boldsymbol{\nu}_{\mathcal{D}}^{+} \\ \boldsymbol{\nu}_{\mathcal{D}}^{-} \\ \boldsymbol{\nu}_{\mathcal{P}} \\ \boldsymbol{\nu}_{\mathcal{N}} \end{bmatrix}.
\end{aligned}
\tag{D.1.2}
$$

The objective function for the dual is obtained by minimizing $L$ with respect to $\mathbf{b}_{\mathcal{D}}^{+}$, $\mathbf{b}_{\mathcal{D}}^{-}$, $\widetilde{\mathbf{b}}_{\mathcal{P}}$, and $\widetilde{\mathbf{b}}_{\mathcal{N}}$. Given that the dual is a maximization problem, we show that it is only necessary to consider $\boldsymbol{\mu}_{\mathcal{D}}^{\pm}$ and $\boldsymbol{\nu}_{\mathcal{D}}^{\pm}$ satisfying $\mathbf{w}_{\mathcal{D}}^{+} + \boldsymbol{\nu}_{\mathcal{D}}^{+} - \boldsymbol{\mu}_{\mathcal{D}}^{+} + \mathbf{w}_{\mathcal{D}}^{-} + \boldsymbol{\nu}_{\mathcal{D}}^{-} - \boldsymbol{\mu}_{\mathcal{D}}^{-} = \mathbf{0}$. This restriction is similar to the constraint $\mathbf{g}^{+} - \boldsymbol{\mu}^{+} + \mathbf{g}^{-} - \boldsymbol{\mu}^{-} = \mathbf{0}$ imposed in Appendix B.2. Supposing to the contrary that there is an index $n \in \mathcal{D}$ such that $w_n^{+} + \nu_n^{+} - \mu_n^{+} + w_n^{-} + \nu_n^{-} - \mu_n^{-} \neq 0$, consider the solution $\mathbf{b}_{\mathcal{D}}^{+} = \alpha\mathbf{e}_n + \mathbf{c}_{\mathcal{D}}/2$, $\mathbf{b}_{\mathcal{D}}^{-} = \alpha\mathbf{e}_n - \mathbf{c}_{\mathcal{D}}/2$, $\widetilde{\mathbf{b}}_{\mathcal{P}} = \widetilde{\mathbf{c}}_{\mathcal{P}}$, and $\widetilde{\mathbf{b}}_{\mathcal{N}} = \widetilde{\mathbf{c}}_{\mathcal{N}}$. The quadratic term in (D.1.2) vanishes and the remainder becomes

$$
L = \alpha(w_n^{+} + \nu_n^{+} - \mu_n^{+} + w_n^{-} + \nu_n^{-} - \mu_n^{-}) +
\begin{bmatrix} \mathbf{w}_{\mathcal{D}}^{+} + \boldsymbol{\nu}_{\mathcal{D}}^{+} - \boldsymbol{\mu}_{\mathcal{D}}^{+} \\ \mathbf{w}_{\mathcal{D}}^{-} + \boldsymbol{\nu}_{\mathcal{D}}^{-} - \boldsymbol{\mu}_{\mathcal{D}}^{-} \\ \mathbf{w}_{\mathcal{P}} + \boldsymbol{\nu}_{\mathcal{P}} - \boldsymbol{\mu}_{\mathcal{P}} \\ -(\mathbf{w}_{\mathcal{N}} + \boldsymbol{\nu}_{\mathcal{N}} - \boldsymbol{\mu}_{\mathcal{N}}) \end{bmatrix}^{T}
\begin{bmatrix} \mathbf{c}_{\mathcal{D}}^{+}/2 \\ -\mathbf{c}_{\mathcal{D}}^{-}/2 \\ \widetilde{\mathbf{c}}_{\mathcal{P}} \\ \widetilde{\mathbf{c}}_{\mathcal{N}} \end{bmatrix}
- \frac{\lambda\gamma}{2} -
\begin{bmatrix} \mathbf{p}_{\mathcal{D}}^{+} \\ \mathbf{p}_{\mathcal{D}}^{-} \\ \mathbf{p}_{\mathcal{P}} \\ \mathbf{p}_{\mathcal{N}} \end{bmatrix}^{T}
\begin{bmatrix} \boldsymbol{\nu}_{\mathcal{D}}^{+} \\ \boldsymbol{\nu}_{\mathcal{D}}^{-} \\ \boldsymbol{\nu}_{\mathcal{P}} \\ \boldsymbol{\nu}_{\mathcal{N}} \end{bmatrix}.
$$

By taking $\alpha$ to $+\infty$ or $-\infty$, we can drive $L$ to $-\infty$. This shows that the dual objective function is unbounded from below in the absence of the constraint $\mathbf{w}_{\mathcal{D}}^{+} + \boldsymbol{\nu}_{\mathcal{D}}^{+} - \boldsymbol{\mu}_{\mathcal{D}}^{+} + \mathbf{w}_{\mathcal{D}}^{-} + \boldsymbol{\nu}_{\mathcal{D}}^{-} - \boldsymbol{\mu}_{\mathcal{D}}^{-} = \mathbf{0}$.

The Lagrangian is minimized with respect to $\mathbf{b}_{\mathcal{D}}^{+}$, $\mathbf{b}_{\mathcal{D}}^{-}$, $\widetilde{\mathbf{b}}_{\mathcal{P}}$, and $\widetilde{\mathbf{b}}_{\mathcal{N}}$ when the following optimality condition is satisfied:

$$\lambda \begin{bmatrix} \mathbf{Q}_{\mathcal{D}\mathcal{D}} & -\mathbf{Q}_{\mathcal{D}\mathcal{D}} & \mathbf{Q}_{\mathcal{D}\mathcal{P}} & \mathbf{Q}_{\mathcal{D}\mathcal{N}} \\ -\mathbf{Q}_{\mathcal{D}\mathcal{D}} & \mathbf{Q}_{\mathcal{D}\mathcal{D}} & -\mathbf{Q}_{\mathcal{D}\mathcal{P}} & -\mathbf{Q}_{\mathcal{D}\mathcal{N}} \\ \mathbf{Q}_{\mathcal{P}\mathcal{D}} & -\mathbf{Q}_{\mathcal{P}\mathcal{D}} & \mathbf{Q}_{\mathcal{P}\mathcal{P}} & \mathbf{Q}_{\mathcal{P}\mathcal{N}} \\ \mathbf{Q}_{\mathcal{N}\mathcal{D}} & -\mathbf{Q}_{\mathcal{N}\mathcal{D}} & \mathbf{Q}_{\mathcal{N}\mathcal{P}} & \mathbf{Q}_{\mathcal{N}\mathcal{N}} \end{bmatrix} \begin{bmatrix} \mathbf{b}_{\mathcal{D}}^{+} - \mathbf{c}_{\mathcal{D}} \\ \mathbf{b}_{\mathcal{D}}^{-} \\ \widetilde{\mathbf{b}}_{\mathcal{P}} - \widetilde{\mathbf{c}}_{\mathcal{P}} \\ \widetilde{\mathbf{b}}_{\mathcal{N}} - \widetilde{\mathbf{c}}_{\mathcal{N}} \end{bmatrix} + \begin{bmatrix} \mathbf{w}_{\mathcal{D}}^{+} + \boldsymbol{\nu}_{\mathcal{D}}^{+} - \boldsymbol{\mu}_{\mathcal{D}}^{+} \\ \mathbf{w}_{\mathcal{D}}^{-} + \boldsymbol{\nu}_{\mathcal{D}}^{-} - \boldsymbol{\mu}_{\mathcal{D}}^{-} \\ \mathbf{w}_{\mathcal{P}} + \boldsymbol{\nu}_{\mathcal{P}} - \boldsymbol{\mu}_{\mathcal{P}} \\ -(\mathbf{w}_{\mathcal{N}} + \boldsymbol{\nu}_{\mathcal{N}} - \boldsymbol{\mu}_{\mathcal{N}}) \end{bmatrix} = \mathbf{0}.$$

Given the constraint $\mathbf{w}_{\mathcal{D}}^{+} + \boldsymbol{\nu}_{\mathcal{D}}^{+} - \boldsymbol{\mu}_{\mathcal{D}}^{+} + \mathbf{w}_{\mathcal{D}}^{-} + \boldsymbol{\nu}_{\mathcal{D}}^{-} - \boldsymbol{\mu}_{\mathcal{D}}^{-} = \mathbf{0}$, the second row is the negative of the first row and is therefore redundant. The remaining system of equations can be solved to yield the minimizer

$$\begin{bmatrix} \mathbf{b}_{\mathcal{D}}^{+} - \mathbf{b}_{\mathcal{D}}^{-} - \mathbf{c}_{\mathcal{D}} \\ \widetilde{\mathbf{b}}_{\mathcal{P}} - \widetilde{\mathbf{c}}_{\mathcal{P}} \\ \widetilde{\mathbf{b}}_{\mathcal{N}} - \widetilde{\mathbf{c}}_{\mathcal{N}} \end{bmatrix} = -\frac{1}{\lambda}\mathbf{Q}^{-1} \begin{bmatrix} \mathbf{w}_{\mathcal{D}}^{+} + \boldsymbol{\nu}_{\mathcal{D}}^{+} - \boldsymbol{\mu}_{\mathcal{D}}^{+} \\ \mathbf{w}_{\mathcal{P}} + \boldsymbol{\nu}_{\mathcal{P}} - \boldsymbol{\mu}_{\mathcal{P}} \\ -(\mathbf{w}_{\mathcal{N}} + \boldsymbol{\nu}_{\mathcal{N}} - \boldsymbol{\mu}_{\mathcal{N}}) \end{bmatrix}, \tag{D.1.3}$$

where $\lambda$ is assumed to be strictly positive as in Appendix B.2 and the partitioning of $\mathbf{Q}^{-1}$ is not shown for brevity. Substituting (D.1.3) into (D.1.2) and simplifying, the dual problem can now be stated as

$$\begin{aligned} \max_{\lambda,\boldsymbol{\mu},\boldsymbol{\nu}} \quad & -\frac{1}{2\lambda} \begin{bmatrix} \mathbf{w}_{\mathcal{D}}^{+} + \boldsymbol{\nu}_{\mathcal{D}}^{+} - \boldsymbol{\mu}_{\mathcal{D}}^{+} \\ \mathbf{w}_{\mathcal{P}} + \boldsymbol{\nu}_{\mathcal{P}} - \boldsymbol{\mu}_{\mathcal{P}} \\ -(\mathbf{w}_{\mathcal{N}} + \boldsymbol{\nu}_{\mathcal{N}} - \boldsymbol{\mu}_{\mathcal{N}}) \end{bmatrix}^{T} \mathbf{Q}^{-1} \begin{bmatrix} \mathbf{w}_{\mathcal{D}}^{+} + \boldsymbol{\nu}_{\mathcal{D}}^{+} - \boldsymbol{\mu}_{\mathcal{D}}^{+} \\ \mathbf{w}_{\mathcal{P}} + \boldsymbol{\nu}_{\mathcal{P}} - \boldsymbol{\mu}_{\mathcal{P}} \\ -(\mathbf{w}_{\mathcal{N}} + \boldsymbol{\nu}_{\mathcal{N}} - \boldsymbol{\mu}_{\mathcal{N}}) \end{bmatrix} - \frac{\lambda\gamma}{2} \\ & + \begin{bmatrix} \mathbf{c}_{\mathcal{D}} \\ \widetilde{\mathbf{c}}_{\mathcal{P}} \\ \widetilde{\mathbf{c}}_{\mathcal{N}} \end{bmatrix}^{T} \begin{bmatrix} \mathbf{w}_{\mathcal{D}}^{+} + \boldsymbol{\nu}_{\mathcal{D}}^{+} - \boldsymbol{\mu}_{\mathcal{D}}^{+} \\ \mathbf{w}_{\mathcal{P}} + \boldsymbol{\nu}_{\mathcal{P}} - \boldsymbol{\mu}_{\mathcal{P}} \\ -(\mathbf{w}_{\mathcal{N}} + \boldsymbol{\nu}_{\mathcal{N}} - \boldsymbol{\mu}_{\mathcal{N}}) \end{bmatrix} - \begin{bmatrix} \mathbf{p}_{\mathcal{D}}^{+} \\ \mathbf{p}_{\mathcal{D}}^{-} \\ \mathbf{p}_{\mathcal{P}} \\ \mathbf{p}_{\mathcal{N}} \end{bmatrix}^{T} \begin{bmatrix} \boldsymbol{\nu}_{\mathcal{D}}^{+} \\ \boldsymbol{\nu}_{\mathcal{D}}^{-} \\ \boldsymbol{\nu}_{\mathcal{P}} \\ \boldsymbol{\nu}_{\mathcal{N}} \end{bmatrix} \\ \text{s.t.} \quad & \mathbf{w}_{\mathcal{D}}^{+} + \boldsymbol{\nu}_{\mathcal{D}}^{+} - \boldsymbol{\mu}_{\mathcal{D}}^{+} + \mathbf{w}_{\mathcal{D}}^{-} + \boldsymbol{\nu}_{\mathcal{D}}^{-} - \boldsymbol{\mu}_{\mathcal{D}}^{-} = \mathbf{0}, \\ & \lambda > 0, \\ & \boldsymbol{\mu}_{\mathcal{D}}^{+} \geq \mathbf{0}, \quad \boldsymbol{\mu}_{\mathcal{D}}^{-} \geq \mathbf{0}, \quad \boldsymbol{\mu}_{\mathcal{P}} \geq \mathbf{0}, \quad \boldsymbol{\mu}_{\mathcal{N}} \geq \mathbf{0}, \\ & \boldsymbol{\nu}_{\mathcal{D}}^{+} \geq \mathbf{0}, \quad \boldsymbol{\nu}_{\mathcal{D}}^{-} \geq \mathbf{0}, \quad \boldsymbol{\nu}_{\mathcal{P}} \geq \mathbf{0}, \quad \boldsymbol{\nu}_{\mathcal{N}} \geq \mathbf{0}. \end{aligned}$$

As in Appendix B.2, the maximization over $\lambda$ can be solved independently, yielding

$$\begin{aligned}\max_{\boldsymbol{\mu},\boldsymbol{\nu}} \quad & -\left(\gamma \begin{bmatrix} \mathbf{w}_{\mathcal{D}}^{+} + \boldsymbol{\nu}_{\mathcal{D}}^{+} - \boldsymbol{\mu}_{\mathcal{D}}^{+} \\ \mathbf{w}_{\mathcal{P}} + \boldsymbol{\nu}_{\mathcal{P}} - \boldsymbol{\mu}_{\mathcal{P}} \\ -(\mathbf{w}_{\mathcal{N}} + \boldsymbol{\nu}_{\mathcal{N}} - \boldsymbol{\mu}_{\mathcal{N}}) \end{bmatrix}^{T} \mathbf{Q}^{-1} \begin{bmatrix} \mathbf{w}_{\mathcal{D}}^{+} + \boldsymbol{\nu}_{\mathcal{D}}^{+} - \boldsymbol{\mu}_{\mathcal{D}}^{+} \\ \mathbf{w}_{\mathcal{P}} + \boldsymbol{\nu}_{\mathcal{P}} - \boldsymbol{\mu}_{\mathcal{P}} \\ -(\mathbf{w}_{\mathcal{N}} + \boldsymbol{\nu}_{\mathcal{N}} - \boldsymbol{\mu}_{\mathcal{N}}) \end{bmatrix}\right)^{1/2} \\ & + \begin{bmatrix} \mathbf{c}_{\mathcal{D}} \\ \widetilde{\mathbf{c}}_{\mathcal{P}} \\ \widetilde{\mathbf{c}}_{\mathcal{N}} \end{bmatrix}^{T} \begin{bmatrix} \mathbf{w}_{\mathcal{D}}^{+} + \boldsymbol{\nu}_{\mathcal{D}}^{+} - \boldsymbol{\mu}_{\mathcal{D}}^{+} \\ \mathbf{w}_{\mathcal{P}} + \boldsymbol{\nu}_{\mathcal{P}} - \boldsymbol{\mu}_{\mathcal{P}} \\ -(\mathbf{w}_{\mathcal{N}} + \boldsymbol{\nu}_{\mathcal{N}} - \boldsymbol{\mu}_{\mathcal{N}}) \end{bmatrix} - \begin{bmatrix} \mathbf{p}_{\mathcal{D}}^{+} \\ \mathbf{p}_{\mathcal{D}}^{-} \\ \mathbf{p}_{\mathcal{P}} \\ \mathbf{p}_{\mathcal{N}} \end{bmatrix}^{T} \begin{bmatrix} \boldsymbol{\nu}_{\mathcal{D}}^{+} \\ \boldsymbol{\nu}_{\mathcal{D}}^{-} \\ \boldsymbol{\nu}_{\mathcal{P}} \\ \boldsymbol{\nu}_{\mathcal{N}} \end{bmatrix} \qquad \text{(D.1.4)} \\ \text{s.t.} \quad & \mathbf{w}_{\mathcal{D}}^{+} + \boldsymbol{\nu}_{\mathcal{D}}^{+} - \boldsymbol{\mu}_{\mathcal{D}}^{+} + \mathbf{w}_{\mathcal{D}}^{-} + \boldsymbol{\nu}_{\mathcal{D}}^{-} - \boldsymbol{\mu}_{\mathcal{D}}^{-} = \mathbf{0}, \\ & \boldsymbol{\mu}_{\mathcal{D}}^{+} \geq \mathbf{0}, \quad \boldsymbol{\mu}_{\mathcal{D}}^{-} \geq \mathbf{0}, \quad \boldsymbol{\mu}_{\mathcal{P}} \geq \mathbf{0}, \quad \boldsymbol{\mu}_{\mathcal{N}} \geq \mathbf{0}, \\ & \boldsymbol{\nu}_{\mathcal{D}}^{+} \geq \mathbf{0}, \quad \boldsymbol{\nu}_{\mathcal{D}}^{-} \geq \mathbf{0}, \quad \boldsymbol{\nu}_{\mathcal{P}} \geq \mathbf{0}, \quad \boldsymbol{\nu}_{\mathcal{N}} \geq \mathbf{0}. \end{aligned}$$

To convert (D.1.4) into its final form (6.3.7), we examine the quantity $\boldsymbol{\rho} = \mathbf{w}_{\mathcal{D}}^{+} + \boldsymbol{\nu}_{\mathcal{D}}^{+} - \boldsymbol{\mu}_{\mathcal{D}}^{+} = -\mathbf{w}_{\mathcal{D}}^{-} - \boldsymbol{\nu}_{\mathcal{D}}^{-} + \boldsymbol{\mu}_{\mathcal{D}}^{-}$ that appears in several places in (D.1.4). It can be seen that $\boldsymbol{\rho}$ ranges over all of $\mathbb{R}^{|\mathcal{D}|}$, but if any component $\rho_n$ exceeds $w_n^{+}$, $\nu_n^{+}$ is forced to be positive, whereas if $\rho_n < -w_n^{-}$, $\nu_n^{-}$ is forced to be positive. Since the vectors $\mathbf{p}_{\mathcal{D}}^{+}$, $\mathbf{p}_{\mathcal{D}}^{-}$, $\mathbf{p}_{\mathcal{P}}$, and $\mathbf{p}_{\mathcal{N}}$ are all strictly positive, the last term in the objective function in (6.3.7) penalizes positive components of $\boldsymbol{\nu}$. We may equivalently express $\boldsymbol{\rho}$ as $\boldsymbol{\rho} = \boldsymbol{\pi}_{\mathcal{D}} + \boldsymbol{\nu}_{\mathcal{D}}^{+} - \boldsymbol{\nu}_{\mathcal{D}}^{-}$, where $-\mathbf{w}_{\mathcal{D}}^{-} \leq \boldsymbol{\pi}_{\mathcal{D}} \leq \mathbf{w}_{\mathcal{D}}^{+}$. The range of values for $\boldsymbol{\pi}_{\mathcal{D}}$ corresponds to the range of values for $\boldsymbol{\rho}$ that are not penalized. In a similar vein, we also substitute $\boldsymbol{\pi}_{\mathcal{P}}$ for $\mathbf{w}_{\mathcal{P}} - \boldsymbol{\mu}_{\mathcal{P}}$ and $\boldsymbol{\pi}_{\mathcal{N}}$ for $-\mathbf{w}_{\mathcal{N}} + \boldsymbol{\mu}_{\mathcal{N}}$. These substitutions result in the final form given in (6.3.7).

## D.2 Derivation of the dual of problem (6.3.15)

This appendix presents a derivation of the dual of problem (6.3.15), the linear relaxation of problem (5.1.2). The derivation is somewhat similar to those in Appendices B.2 and D.1.

In Section 6.3.3, we defined a vector $\mathbf{s}$ that collects together all variables $s_{np}^{\pm}$ that have not been fixed. We also defined a power-of-two matrix $\mathbf{P}$ such that $\mathbf{b} = \mathbf{Ps}$, and matrices $\mathbf{J}$, $\mathbf{F}$, and a vector $\boldsymbol{\ell}$ to represent constraints (6.3.13) and (6.3.14). With these definitions and neglecting the constant term in the cost function, problem (6.3.15) can be stated in a more compact form:

$$\begin{aligned}
\min_{\mathbf{s}} \quad & \mathbf{e}^T\mathbf{s} \\
\text{s.t.} \quad & (\mathbf{Ps} - \widetilde{\mathbf{c}})^T\mathbf{Q}(\mathbf{Ps} - \widetilde{\mathbf{c}}) \le \gamma, \\
& \mathbf{Js} \le \mathbf{e}, \\
& \mathbf{Fs} \ge \boldsymbol{\ell}, \\
& \mathbf{0} \le \mathbf{s} \le \mathbf{e},
\end{aligned} \tag{D.2.1}$$

recalling that $\mathbf{e}$ denotes a vector of ones with dimensions that depend on context. We associate a non-negative Lagrange multiplier $\lambda/2$ with the quadratic constraint in (D.2.1), and vectors of Lagrange multipliers $\boldsymbol{\mu}$, $\boldsymbol{\nu}$, $\boldsymbol{\pi}^-$ and $\boldsymbol{\pi}^+$ with the constraints $\mathbf{Js} \le \mathbf{e}$, $\mathbf{Fs} \ge \boldsymbol{\ell}$, $\mathbf{s} \ge \mathbf{0}$, and $\mathbf{s} \le \mathbf{e}$ respectively. The resulting Lagrangian is

$$L = (\mathbf{e}+\mathbf{J}^T\boldsymbol{\mu}-\mathbf{F}^T\boldsymbol{\nu}+\boldsymbol{\pi}^+-\boldsymbol{\pi}^-)^T\mathbf{s}+\frac{\lambda}{2}(\mathbf{Ps}-\widetilde{\mathbf{c}})^T\mathbf{Q}(\mathbf{Ps}-\widetilde{\mathbf{c}})-\frac{\lambda\gamma}{2}-\mathbf{e}^T\boldsymbol{\mu}+\boldsymbol{\ell}^T\boldsymbol{\nu}-\mathbf{e}^T\boldsymbol{\pi}^+. \tag{D.2.2}$$

As in Appendices B.2 and D.1, the dual objective function is obtained by minimizing the Lagrangian with respect to $\mathbf{s}$. Also similar to before, we show that it is not necessary to consider certain combinations of Lagrange multipliers (i.e., dual variables) since they lead to a dual objective value of $-\infty$. To make this more explicit, we consider vectors $\mathbf{s}$ of the form $\mathbf{s} = \mathbf{s}_0 + \alpha\mathbf{s}_{\text{ker}}$, where $\mathbf{s}_0$ is an arbitrary fixed vector and $\mathbf{s}_{\text{ker}}$ belongs to the nullspace of $\mathbf{P}$. The Lagrangian then takes the form

$$L = (\mathbf{e}+\mathbf{J}^T\boldsymbol{\mu}-\mathbf{F}^T\boldsymbol{\nu}+\boldsymbol{\pi}^+-\boldsymbol{\pi}^-)^T(\mathbf{s}_0+\alpha\mathbf{s}_{\text{ker}})+\frac{\lambda}{2}(\mathbf{Ps}_0-\widetilde{\mathbf{c}})^T\mathbf{Q}(\mathbf{Ps}_0-\widetilde{\mathbf{c}})-\frac{\lambda\gamma}{2}-\mathbf{e}^T\boldsymbol{\mu}+\boldsymbol{\ell}^T\boldsymbol{\nu}-\mathbf{e}^T\boldsymbol{\pi}^+,$$

which is affine in $\alpha$. Unless $(\mathbf{e} + \mathbf{J}^T\boldsymbol{\mu} - \mathbf{F}^T\boldsymbol{\nu} + \boldsymbol{\pi}^+ - \boldsymbol{\pi}^-)^T\mathbf{s}_{\text{ker}} = 0$, the value of the

Lagrangian approaches $-\infty$ as $\alpha \to \infty$ or $\alpha \to -\infty$. Hence in the dual problem, we can restrict attention to dual variables $\boldsymbol{\mu}$, $\boldsymbol{\nu}$, $\boldsymbol{\pi}^+$ and $\boldsymbol{\pi}^-$ for which $\mathbf{e}+\mathbf{J}^T\boldsymbol{\mu}-\mathbf{F}^T\boldsymbol{\nu}+\boldsymbol{\pi}^+-\boldsymbol{\pi}^-$ is orthogonal to every vector in the nullspace of $\mathbf{P}$. In other words, $\mathbf{e}+\mathbf{J}^T\boldsymbol{\mu}-\mathbf{F}^T\boldsymbol{\nu}+\boldsymbol{\pi}^+-\boldsymbol{\pi}^-$ belongs to the row space of $\mathbf{P}$, which we represent by the constraint

$$\mathbf{e}+\mathbf{J}^T\boldsymbol{\mu}-\mathbf{F}^T\boldsymbol{\nu}+\boldsymbol{\pi}^+-\boldsymbol{\pi}^- = \mathbf{P}^T\boldsymbol{\rho} \tag{D.2.3}$$

for some vector $\boldsymbol{\rho}$ in $\mathbb{R}^N$. Substituting (D.2.3) into (D.2.2) results in

$$L = \boldsymbol{\rho}^T\mathbf{P}\mathbf{s} + \frac{\lambda}{2}(\mathbf{P}\mathbf{s}-\widetilde{\mathbf{c}})^T\mathbf{Q}(\mathbf{P}\mathbf{s}-\widetilde{\mathbf{c}}) - \frac{\lambda\gamma}{2} - \mathbf{e}^T\boldsymbol{\mu} + \boldsymbol{\ell}^T\boldsymbol{\nu} - \mathbf{e}^T\boldsymbol{\pi}^+, \tag{D.2.4}$$

which depends on $\mathbf{s}$ only through the quantity $\mathbf{P}\mathbf{s}$. Minimizing (D.2.4) with respect to $\mathbf{P}\mathbf{s}$ by setting the gradient to zero, we obtain the condition

$$\mathbf{P}\mathbf{s}-\widetilde{\mathbf{c}} = -\frac{1}{\lambda}\mathbf{Q}^{-1}\boldsymbol{\rho},$$

which yields the dual objective function upon substitution into (D.2.4).

The dual of (D.2.1) can now be formulated as follows:

$$\begin{aligned}
\max_{\boldsymbol{\mu},\boldsymbol{\nu},\boldsymbol{\pi}^{\pm},\boldsymbol{\rho}} \quad & \widetilde{\mathbf{c}}^T\boldsymbol{\rho} - \frac{1}{2\lambda}\boldsymbol{\rho}^T\mathbf{Q}^{-1}\boldsymbol{\rho} - \frac{\lambda\gamma}{2} - \mathbf{e}^T\boldsymbol{\mu} + \boldsymbol{\ell}^T\boldsymbol{\nu} - \mathbf{e}^T\boldsymbol{\pi}^+, \\
\text{s.t.} \quad & \mathbf{e}+\mathbf{J}^T\boldsymbol{\mu}-\mathbf{F}^T\boldsymbol{\nu}+\boldsymbol{\pi}^+-\boldsymbol{\pi}^- = \mathbf{P}^T\boldsymbol{\rho}, \\
& \lambda > 0, \quad \boldsymbol{\mu} \geq \mathbf{0}, \quad \boldsymbol{\nu} \geq \mathbf{0}, \quad \boldsymbol{\pi}^{\pm} \geq \mathbf{0}.
\end{aligned}$$

As in Appendices B.2 and D.1, the maximization over $\lambda$ can be done independently. The resulting maximization problem is the one shown in (6.3.18).

# Appendix E

# Proofs for Chapter 8

## E.1 Proof of Theorem 8

To facilitate the proof, we first show that the inequality

$$\left|\mathbf{A}_n^T(\mathbf{p}^+ - \mathbf{p}^-)\right| < C \tag{E.1.1}$$

holds for any feasible solution $(\mathbf{p}^+, \mathbf{p}^-)$ to the dual problem (8.2.2), any column $\mathbf{A}_n$ of the matrix $\mathbf{A}$ defined in (8.1.4a) ($n$ is not necessarily in $\mathcal{Y}$), and $C$ satisfying (8.2.6). Using the fact that the magnitude of a sum is bounded by the sum of the magnitudes of individual terms,

$$\begin{aligned}\left|\mathbf{A}_n^T(\mathbf{p}^+ - \mathbf{p}^-)\right| &= \left|\sum_{k=1}^{K} W(\omega_k)\cos(n\omega_k)\left(p_k^+ - p_k^-\right)\right| \\ &\leq \sum_{k=1}^{K} W(\omega_k)\left|\cos(n\omega_k)\right|\left|p_k^+ - p_k^-\right| \\ &\leq \sum_{k=1}^{K} W(\omega_k)\left|\cos(n\omega_k)\right|\left(p_k^+ + p_k^-\right),\end{aligned}$$

noting that $W(\omega_k)$, $p_k^+$ and $p_k^-$ are non-negative. Bounding $W(\omega_k)$ by its maximum value and $|\cos(n\omega_k)|$ by 1, we obtain

$$\left|\mathbf{A}_n^T(\mathbf{p}^+ - \mathbf{p}^-)\right| \leq \max_{k=1,\ldots,K} W(\omega_k)\sum_{k=1}^{K}\left(p_k^+ + p_k^-\right) < C$$

as desired, where the second inequality follows from the first constraint in (8.2.2) and the assumption (8.2.6) on $C$.

Proceeding with the proof of the main result, suppose $\left(\hat{\delta}, \hat{\mathbf{b}}_{\mathcal{Y}_2}, \hat{b}_m^+, \hat{b}_m^-\right)$ is an optimal solution to (8.2.5). Then $\hat{b}_m^+$ and $\hat{b}_m^-$ cannot both be non-zero, as otherwise both could be decreased by $\min\left\{\hat{b}_m^+, \hat{b}_m^-\right\}$, reducing the objective value without affecting feasibility. Assume first that $\hat{b}_m^- = 0$ and $\hat{b}_m^+ > 0$. Let $\left(\delta^*, \mathbf{b}_{\mathcal{Y}_2}^*\right)$ be an optimal solution to (8.2.1) with $\mathcal{Y} = \mathcal{Y}_2$, and $(\mathbf{p}^{+*}, \mathbf{p}^{-*})$ be an optimal solution to the corresponding dual (8.2.2). We wish to show that $\left(\delta^*, \mathbf{b}_{\mathcal{Y}_2}^*, 0, 0\right)$, which is a feasible solution to (8.2.5), has a strictly lower objective value than the assumed optimal solution $\left(\hat{\delta}, \hat{\mathbf{b}}_{\mathcal{Y}_2}, \hat{b}_m^+, \hat{b}_m^-\right)$, thus establishing a contradiction.

First, we use strong duality to equate the optimal values for (8.2.1) and (8.2.2) under $\mathcal{Y} = \mathcal{Y}_2$:

$$\delta^* = \mathbf{d}^T\left(\mathbf{p}^{+*} - \mathbf{p}^{-*}\right). \tag{E.1.2}$$

Since $\mathbf{p}^{+*}$ and $\mathbf{p}^{-*}$ are non-negative and $\left(\hat{\delta}, \hat{\mathbf{b}}_{\mathcal{Y}_2}, \hat{b}_m^+, \hat{b}_m^-\right)$ satisfies the constraints for (8.2.5),

$$\mathbf{d}^T\mathbf{p}^{+*} \leq \left(\begin{bmatrix}\mathbf{e} & \mathbf{A}_{\mathcal{Y}_2}\end{bmatrix}\begin{bmatrix}\hat{\delta} \\ \hat{\mathbf{b}}_{\mathcal{Y}_2}\end{bmatrix} + \mathbf{A}_m\hat{b}_m^+\right)^T \mathbf{p}^{+*}, \tag{E.1.3}$$

$$-\mathbf{d}^T\mathbf{p}^{-*} \leq \left(\begin{bmatrix}\mathbf{e} & -\mathbf{A}_{\mathcal{Y}_2}\end{bmatrix}\begin{bmatrix}\hat{\delta} \\ \hat{\mathbf{b}}_{\mathcal{Y}_2}\end{bmatrix} - \mathbf{A}_m\hat{b}_m^+\right)^T \mathbf{p}^{-*}. \tag{E.1.4}$$

Combining (E.1.2)–(E.1.4),

$$\begin{aligned}\delta^* &\leq \hat{\delta}\mathbf{e}^T(\mathbf{p}^{+*} + \mathbf{p}^{-*}) + \hat{\mathbf{b}}_{\mathcal{Y}_2}^T\mathbf{A}_{\mathcal{Y}_2}^T(\mathbf{p}^{+*} - \mathbf{p}^{-*}) + \hat{b}_m^+\mathbf{A}_m^T(\mathbf{p}^{+*} - \mathbf{p}^{-*}) \\ &= \hat{\delta} + \hat{b}_m^+\mathbf{A}_m^T(\mathbf{p}^{+*} - \mathbf{p}^{-*}),\end{aligned} \tag{E.1.5}$$

where the simplifications result from the feasibility of $(\mathbf{p}^{+*}, \mathbf{p}^{-*})$ for the dual (8.2.2). Applying the bound in (E.1.1) to (E.1.5),

$$\delta^* < \hat{\delta} + C\hat{b}_m^+.$$

The left-hand side represents the objective value of the feasible solution $\left(\delta^*, \mathbf{b}_{\mathcal{Y}_2}^*, 0, 0\right)$, while the right-hand side represents the value of the assumed optimal solution $\left(\hat{\delta}, \hat{\mathbf{b}}_{\mathcal{Y}_2}, \hat{b}_m^+, 0\right)$.

This contradicts the optimality of $\left(\hat{\delta}, \hat{\mathbf{b}}_{\mathcal{Y}_2}, \hat{b}_m^+, 0\right)$, and hence $\hat{b}_m^+$ must be zero. The case $\hat{b}_m^+ = 0$, $\hat{b}_m^- > 0$ is similarly excluded.

The conclusion that $\hat{b}_m^+ = \hat{b}_m^- = 0$ has two consequences: First, the pair $\left(\hat{\delta}, \hat{\mathbf{b}}_{\mathcal{Y}_2}\right)$ becomes a feasible solution to (8.2.1) with $\mathcal{Y} = \mathcal{Y}_2$. Secondly, the inequality in (E.1.5) becomes $\delta^* \leq \hat{\delta}$, and in fact equality must hold in order for $\left(\hat{\delta}, \hat{\mathbf{b}}_{\mathcal{Y}_2}, 0, 0\right)$ to be an optimal solution to (8.2.5). Therefore $\left(\hat{\delta}, \hat{\mathbf{b}}_{\mathcal{Y}_2}\right)$ is also an optimal solution to (8.2.1) for $\mathcal{Y} = \mathcal{Y}_2$, completing the proof of the forward direction.

To prove the converse, suppose that $\left(\delta^*, \mathbf{b}^*_{\mathcal{Y}_2}\right)$ is an optimal solution to (8.2.1) with $\mathcal{Y} = \mathcal{Y}_2$. It was shown in the proof of the forward direction, specifically in (E.1.5), that the optimal objective value in (8.2.5) can be no less than $\delta^*$. Furthermore, $\left(\delta^*, \mathbf{b}^*_{\mathcal{Y}_2}, 0, 0\right)$ is a feasible solution to (8.2.5) and achieves a value of $\delta^*$. We conclude that $\left(\delta^*, \mathbf{b}^*_{\mathcal{Y}_2}, 0, 0\right)$ is an optimal solution to (8.2.5).

## E.2 Proof of Theorem 9

For ease of notation, we collect all of the variables in (8.3.6) into a single vector $\mathbf{x} = (b_0^+, b_0^-, b_1^+, b_1^-, \ldots, b_{N-1}^+, b_{N-1}^-)$ ($b_n^+$ and $b_n^-$ are interleaved for later convenience). The vector $\mathbf{x}_{\mathcal{Y}}$ is defined similarly except that only the indices in $\mathcal{Y}$ are included. We prove the theorem for the case in which $\mathcal{Y} = \{0, \ldots, N-1\}$ so that $\mathbf{x}_{\mathcal{Y}} = \mathbf{x}$ and $\mathcal{P}_{\mathcal{Y}} = \mathcal{P}$. For cases in which $\mathcal{Y} \neq \{0, \ldots, N-1\}$, it suffices to observe that a local minimum $\mathbf{x}^*$ for (8.3.6) is also a local minimum for the problem

$$\begin{aligned} \min_{\mathbf{x}} \quad & F(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{P}, \\ & x_{2n} = x_{2n+1} = 0, \quad n \in \mathcal{Z}, \end{aligned}$$

i.e., the restriction of (8.3.6) to a smaller feasible set. Equivalently, the vector $\mathbf{x}^*_{\mathcal{Y}}$ is a local minimum of

$$\min_{\mathbf{x}_{\mathcal{Y}}} \quad F(\mathbf{x}_{\mathcal{Y}}) \quad \text{s.t.} \quad \mathbf{x}_{\mathcal{Y}} \in \mathcal{P}_{\mathcal{Y}},$$

and the proof follows with $\mathbf{x}$ and $\mathcal{P}$ replaced by $\mathbf{x}_{\mathcal{Y}}$ and $\mathcal{P}_{\mathcal{Y}}$.

We prove by contradiction that (8.3.8) holds, following ideas from [98]. Suppose that there exists a vector $\overline{\mathbf{x}} \in \mathcal{P}$, $\overline{\mathbf{x}} \neq \mathbf{x}^*$, such that $\nabla F(\mathbf{x}^*)^T(\overline{\mathbf{x}} - \mathbf{x}^*) < 0$. Define $\Delta\mathbf{x} = \overline{\mathbf{x}} - \mathbf{x}^*$

and

$$g(\epsilon) = F\left(\mathbf{x}^* + \epsilon\Delta\mathbf{x}\right), \quad \epsilon \in [0, \overline{\epsilon}]$$

to be the restriction of $F(\mathbf{x})$ to a line segment in the direction of $\Delta\mathbf{x}$. Since $x^*_{2n} + x^*_{2n+1} > 0$ for all $n$, it is possible to choose $\overline{\epsilon}$ sufficiently small so that

$$x^*_{2n} + x^*_{2n+1} + \epsilon\left(\Delta x_{2n} + \Delta x_{2n+1}\right) > 0, \quad n = 0, \ldots, N-1, \tag{E.2.1}$$

for all $\epsilon \in [0, \overline{\epsilon}]$. The function $F$ is continuously differentiable wherever (E.2.1) holds, and as a result,

$$g'(\epsilon) = \nabla F\left(\mathbf{x}^* + \epsilon\Delta\mathbf{x}\right)^T \Delta\mathbf{x}, \quad \epsilon \in [0, \overline{\epsilon}],$$

exists and is continuous. Applying the mean value theorem to $g(\epsilon)$, there exists a number $s \in [0, 1]$ such that

$$F\left(\mathbf{x}^* + \epsilon\Delta\mathbf{x}\right) = F(\mathbf{x}^*) + \epsilon\nabla F\left(\mathbf{x}^* + s\epsilon\Delta\mathbf{x}\right)^T \Delta\mathbf{x} \quad \forall\, \epsilon \in [0, \overline{\epsilon}].$$

As $\epsilon$ converges to zero, $\nabla F\left(\mathbf{x}^* + s\epsilon\Delta\mathbf{x}\right)$ converges to $\nabla F(\mathbf{x}^*)$, and consequently the second term on the right-hand side becomes negative according to the assumption. Therefore, for all $\epsilon$ sufficiently small, $F\left(\mathbf{x}^* + \epsilon\Delta\mathbf{x}\right) < F(\mathbf{x}^*)$, which contradicts the fact that $\mathbf{x}^*$ is a local minimum.

Next, suppose that $\overline{\mathbf{x}} \in \mathcal{P}$, $\overline{\mathbf{x}} \neq \mathbf{x}^*$, is such that $\nabla F(\mathbf{x}^*)^T\Delta\mathbf{x} = 0$. As before, when $\overline{\epsilon}$ is sufficiently small, $\mathbf{x}^* + \epsilon\Delta\mathbf{x}$ satisfies condition (E.2.1) for all $\epsilon \in [0, \overline{\epsilon}]$. Since $F$ is also twice continuously differentiable wherever (E.2.1) holds, the second-order version of the mean value theorem guarantees the existence of $s \in [0, 1]$ such that

$$\begin{aligned} F\left(\mathbf{x}^* + \epsilon\Delta\mathbf{x}\right) &= F(\mathbf{x}^*) + \epsilon\nabla F(\mathbf{x}^*)^T\Delta\mathbf{x} + \frac{1}{2}\epsilon^2\Delta\mathbf{x}^T\nabla^2 F\left(\mathbf{x}^* + s\epsilon\Delta\mathbf{x}\right)\Delta\mathbf{x}, \\ &= F(\mathbf{x}^*) + \frac{1}{2}\epsilon^2\Delta\mathbf{x}^T\nabla^2 F\left(\mathbf{x}^* + s\epsilon\Delta\mathbf{x}\right)\Delta\mathbf{x} \quad \forall\, \epsilon \in [0, \overline{\epsilon}], \end{aligned}$$

where the first-order term is zero by assumption. The Hessian $\nabla^2 F(\mathbf{x})$ is a block-diagonal matrix with $2 \times 2$ diagonal blocks as follows:

$$\nabla^2 F(\mathbf{x}) = p(p-1)\times$$

$$\operatorname{Diag}\left(|x_0+x_1|^{p-2}\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix}, |x_2+x_3|^{p-2}\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix}, \ldots, |x_{2N-2}+x_{2N-1}|^{p-2}\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix}\right).$$

It can be seen that for $p < 1$, each $2 \times 2$ block is negative semidefinite and the product $\Delta\mathbf{x}^T \nabla^2 F\left(\mathbf{x}^* + s\epsilon\Delta\mathbf{x}\right)\Delta\mathbf{x}$ is strictly negative unless $\Delta\mathbf{x}$ is of the form

$$\Delta\mathbf{x} = (-k_0, +k_0, -k_1, +k_1, \ldots, -k_{N-1}, +k_{N-1}), \tag{E.2.2}$$

in which case the product is zero. Hence if $\Delta\mathbf{x}$ does not conform to (E.2.2), then $F\left(\mathbf{x}^* + \epsilon\Delta\mathbf{x}\right) < F(\mathbf{x}^*)$ for all $\epsilon \in (0, \bar{\epsilon}]$, again contradicting the local minimality of $\mathbf{x}^*$.

It remains to consider the case in which $\Delta\mathbf{x}$ is of the form in (E.2.2) and the $k_n$ are not all zero. First, note that the signs of $k_n$ in (E.2.2) cannot be arbitrary. Recalling that $\mathbf{x}^*$ must have the property that $x^*_{2n}x^*_{2n+1} = 0$, it follows that $k_n$ must be non-negative if $x^*_{2n+1} = 0$ and must be non-positive if $x^*_{2n} = 0$. Assume without loss of generality that $x^*_{2n+1} = 0$ for all $n$ so that $k_n \geq 0$. Given that $\overline{\mathbf{x}}$ is feasible, the points

$$\mathbf{x}^* + \epsilon\Delta\mathbf{x} = (x^*_0 - \epsilon k_0, \epsilon k_0, x^*_2 - \epsilon k_1, \epsilon k_1, \ldots, x^*_{2N-2} - \epsilon k_{N-1}, \epsilon k_{N-1}), \quad \epsilon \in (0, 1],$$

are also feasible by the convexity of $\mathcal{P}$ and have the same cost value as $\mathbf{x}^*$. Since feasibility depends only on the differences $x_{2n} - x_{2n+1}$, the points

$$(x^*_0 - 2\epsilon k_0, 0, x^*_2 - 2\epsilon k_1, 0, \ldots, x^*_{2N-2} - 2\epsilon k_{N-1}, 0), \quad \epsilon \in (0, 1],$$

are also feasible, but have a strictly lower cost than $\mathbf{x}^*$. This is again a contradiction, and thus we have proven that condition (8.3.8) holds for any local minimum $\mathbf{x}^*_{\mathcal{Y}}$. The vertex property follows by definition because (8.3.8) implies that $\mathbf{x}^*_{\mathcal{Y}}$ can be strictly separated from the rest of $\mathcal{P}_{\mathcal{Y}}$ by a hyperplane, specifically one normal to $\nabla F(\mathbf{x}^*_{\mathcal{Y}})$.

# Bibliography

[1] D. Mattera, F. Palmieri, and S. Haykin, "Efficient sparse FIR filter design," in *Proc. ICASSP*, vol. 2, May 2002, pp. 1537–1540.

[2] C. K. Sestok, "Data selection in binary hypothesis testing," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, Dec. 2003.

[3] J. T. Kim, W. J. Oh, and Y. H. Lee, "Design of nonuniformly spaced linear-phase FIR filters using mixed integer linear programming," *IEEE Trans. Signal Process.*, vol. 44, pp. 123–126, Jan. 1996.

[4] L. Vandenberghe, S. Boyd, and S.-P. Wu, "Determinant maximization with linear matrix inequality constraints," *SIAM J. Matrix Anal. Appl.*, vol. 19, no. 2, pp. 499–533, 1998.

[5] T. W. Parks and J. H. McClellan, "Chebyshev approximation for nonrecursive digital filters with linear phase," *IEEE Trans. Circuit Theory*, vol. 19, pp. 189–194, March 1972.

[6] N. Sankarayya, K. Roy, and D. Bhattacharya, "Algorithms for low power and high speed FIR filter realization using differential coefficients," *IEEE Trans. Circuits Syst. II*, vol. 44, pp. 488–497, November 1997.

[7] S. Ramprasad, N. R. Shanbhag, and I. N. Hajj, "Decorrelating (DECOR) transformations for low-power digital filters," *IEEE Trans. Circuits Syst. II*, vol. 46, no. 6, pp. 776–788, 1999.

[8] H. Samueli, "An improved search algorithm for the design of multiplierless FIR filters with powers-of-two coefficients," *IEEE Trans. Circuits Syst.*, vol. 36, no. 7, pp. 1044–1047, 1989.

[9] C.-L. Chen and A. N. Willson, Jr., "A trellis search algorithm for the design of FIR filters with signed-powers-of-two coefficients," *IEEE Trans. Circuits Syst. II*, vol. 46, pp. 29–39, Jan. 1999.

[10] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing.* Upper Saddle River, NJ: Prentice-Hall, Inc., 1999.

[11] E. B. Hogenauer, "An economical class of digital filters for decimation and interpolation," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 29, no. 2, pp. 155–162, 1981.

[12] A. Y. Kwentus, Z. Jiang, and A. N. Willson, Jr., "Application of filter sharpening to cascaded integrator-comb decimation filters," *IEEE Trans. Signal Process.*, vol. 45, no. 2, 1997.

[13] J. Adams and A. Willson, Jr., "A new approach to FIR digital filters with fewer multipliers and reduced sensitivity," *IEEE Trans. Circuits Syst.*, vol. 30, pp. 277–283, May 1983.

[14] ——, "Some efficient digital prefilter structures," *IEEE Trans. Circuits Syst.*, vol. 31, pp. 260–266, Mar. 1984.

[15] G. Boudreaux and T. Parks, "Thinning digital filters: A piecewise-exponential approximation approach," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 31, pp. 105–113, Feb. 1983.

[16] Z. Jing and A. Fam, "A new structure for narrow transition band, lowpass digital filter design," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 32, no. 2, pp. 362–370, 1984.

[17] Y. Neuvo, C.-Y. Dong, and S. Mitra, "Interpolated finite impulse response filters," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 32, pp. 563–570, Jun. 1984.

[18] T. Saramaki, T. Neuvo, and S. K. Mitra, "Design of computationally efficient interpolated FIR filters," *IEEE Trans. Circuits Syst.*, vol. 35, pp. 70–88, Jan. 1988.

[19] J. L. H. Webb and D. C. Munson, Jr., "A new approach to designing computationally efficient interpolated FIR filters," *IEEE Trans. Signal Process.*, vol. 44, no. 8, pp. 1923–1931, 1996.

[20] Y. C. Lim, "Frequency-response masking approach for the synthesis of sharp linear phase digital filters," *IEEE Trans. Circuits Syst.*, vol. 33, pp. 357–364, Apr. 1986.

[21] Y. C. Lim and Y. Lian, "Frequency-response masking approach for digital filter design: complexity reduction via masking filter factorization," *IEEE Trans. Circuits Syst. II*, vol. 41, pp. 518–525, Aug. 1994.

[22] R. J. Hartnett and G. F. Boudreaux-Bartels, "On the use of cyclotomic polynomial prefilters for efficient FIR filter design," *IEEE Trans. Signal Process.*, vol. 41, pp. 1766–1779, May 1993.

[23] J. T. Kim, W. J. Oh, and Y. H. Lee, "Design of nonuniformly spaced linear-phase FIR filters using mixed integer linear programming," *IEEE Trans. Signal Process.*, vol. 44, no. 1, pp. 123–126, 1996.

[24] Y.-S. Song and Y. H. Lee, "Design of sparse FIR filters based on branch-and-bound algorithm," in *Proc. 40th Midwest Symp. Circuits and Systems*, vol. 2, 1997, pp. 1445–1448.

[25] J. L. H. Webb and D. C. Munson, "Chebyshev optimization of sparse FIR filters using linear programming with an application to beamforming," *IEEE Trans. Signal Process.*, vol. 44, pp. 1912–1922, Aug. 1996.

[26] Y.-S. Song and Y. H. Lee, "Design of sparse FIR filters based on branch-and-bound algorithm," in *Proc. MWSCAS*, vol. 2, Aug. 1997, pp. 1445–1448.

[27] J.-K. Liang, R. de Figueiredo, and F. Lu, "Design of optimal Nyquist, partial response, Nth band, and nonuniform tap spacing FIR digital filters using linear programming techniques," *IEEE Trans. Circuits Syst.*, vol. 32, pp. 386–392, Apr. 1985.

[28] T. A. Baran and A. V. Oppenheim, "Design and implementation of discrete-time filters for efficient rate-conversion systems," in *Proc. Asilomar Conf. Signals Syst. Comp.*, Nov. 2007.

[29] R. M. Leahy and B. D. Jeffs, "On the design of maximally sparse beamforming arrays," *IEEE Trans. Antennas Propag.*, vol. 39, pp. 1178–1187, Aug. 1991.

[30] J. W. Adams, "FIR digital filters with least-squares stopbands subject to peak-gain constraints," *IEEE Trans. Circuits Syst.*, vol. 39, pp. 376–388, Apr. 1991.

[31] M. Smith and D. Farden, "Thinning the impulse response of FIR digital filters," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 6, 1981, pp. 240–242.

[32] S. F. Cotter and B. D. Rao, "Sparse channel estimation via matching pursuit with application to equalization," *IEEE Trans. Commun.*, vol. 50, no. 3, pp. 374–377, Mar. 2002.

[33] S. A. Raghavan, J. K. Wolf, L. B. Milstein, and L. C. Barbosa, "Non-uniformly spaced tapped-delay-line equalizers," *IEEE Trans. Commun.*, vol. 41, no. 9, pp. 1290–1295, Sep. 1993.

[34] I. Lee, "Optimization of tap spacings for the tapped delay line decision feedback equalizer," *IEEE Commun. Lett.*, vol. 5, no. 10, pp. 429–431, Oct. 2001.

[35] M. Kocic, D. Brady, and M. Stojanovic, "Sparse equalization for real-time digital underwater acoustic communications," in *IEEE OCEANS*, vol. 3, Oct. 1995, pp. 1417–1422.

[36] A. A. Rontogiannis and K. Berberidis, "Efficient decision feedback equalization for sparse wireless channels," *IEEE Trans. Wireless Commun.*, vol. 2, no. 3, pp. 570–581, May 2003.

[37] F. K. H. Lee and P. J. McLane, "Design of nonuniformly spaced tapped-delay-line equalizers for sparse multipath channels," *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 530–535, Apr. 2004.

[38] I. J. Fevrier, S. B. Gelfand, and M. P. Fitz, "Reduced complexity decision feedback equalization for multipath channels with large delay spreads," *IEEE Trans. Commun.*, vol. 47, no. 6, pp. 927–937, Jun. 1999.

[39] S. Ariyavisitakul, N. R. Sollenberger, and L. J. Greenstein, "Tap-selectable decision-feedback equalization," *IEEE Trans. Commun.*, vol. 45, no. 12, pp. 1497–1500, Dec. 1997.

[40] M. J. Lopez and A. C. Singer, "A DFE coefficient placement algorithm for sparse reverberant channels," *IEEE Trans. Commun.*, vol. 49, no. 8, pp. 1334–1338, Aug. 2001.

[41] H. Sui, E. Masry, and B. D. Rao, "Chip-level DS-CDMA downlink interference suppression with optimized finger placement," *IEEE Trans. Signal Process.*, vol. 54, no. 10, pp. 3908–3921, Oct. 2006.

[42] G. Kutz and D. Raphaeli, "Determination of tap positions for sparse equalizers," *IEEE Trans. Commun.*, vol. 55, no. 9, pp. 1712–1724, Sep. 2007.

[43] D. Giacobello, M. N. Murthi, M. G. Christensen, S. H. Jensen, and M. Moonen, "Enhancing sparsity in linear prediction of speech by iteratively reweighted 1-norm minimization," in *Proc. ICASSP*, Mar. 2010, pp. 4650–4653.

[44] H. L. V. Trees, *Detection, Estimation, and Modulation Theory.* New York: John Wiley & Sons, 2004, vol. 1.

[45] C. K. Sestok, "Data selection for detection of known signals: The restricted-length matched filter," in *Proc. ICASSP*, vol. 2, May 2004, pp. 1085–1088.

[46] E. Avenhaus, "On the design of digital filters with coefficients of limited word length," *IEEE Trans. Audio Electroacoust.*, vol. 20, no. 3, pp. 206–212, Aug. 1972.

[47] F. Brglez, "Digital filter design with short word-length coefficients," *IEEE Trans. Circuits Syst.*, vol. 25, no. 12, pp. 1044–1050, Dec. 1978.

[48] R. E. Crochiere, "A new statistical approach to the coefficient word length problem for digital filters," *IEEE Trans. Circuits Syst.*, vol. 22, no. 3, pp. 190–196, Mar. 1975.

[49] D. Kodek, "Design of optimal finite wordlength FIR digital filters using integer programming techniques," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 28, no. 3, pp. 304–308, 1980.

[50] J. M. de Sa, "A new design method of optimal finite wordlength linear phase FIR digital filters," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 31, no. 4, pp. 1032–1034, 1983.

[51] D. Kodek and K. Steiglitz, "Comparison of optimal and local search methods for designing finite wordlength FIR digital filters," *IEEE Trans. Circuits Syst.*, vol. 28, no. 1, pp. 28–32, 1981.

[52] Y. Lim and S. Parker, "FIR filter design over a discrete powers-of-two coefficient space," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 31, pp. 583–591, Jun. 1983.

[53] Y. C. Lim, "Design of discrete-coefficient-value linear phase FIR filters with optimum normalized peak ripple magnitude," *IEEE Trans. Circuits Syst.*, vol. 37, pp. 1480–1486, Dec. 1990.

[54] B. Jaumard, M. Minoux, and P. Siohan, "Finite precision design of FIR digital filters using a convexity property," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 36, no. 3, pp. 407–411, 1988.

[55] N. I. Cho and S. U. Lee, "Optimal design of finite precision FIR filters using linear programming with reduced constraints," *IEEE Trans. Signal Process.*, vol. 46, no. 1, pp. 195–199, 1998.

[56] D. M. Kodek, "Performance limit of finite wordlength FIR digital filters," *IEEE Trans. Signal Process.*, vol. 53, no. 7, pp. 2462–2469, 2005.

[57] K. Nakayama, "A discrete optimization method for high-order FIR filters with finite wordlength coefficients," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 35, no. 8, pp. 1215–1217, 1987.

[58] Q. Zhao and Y. Tadokoro, "A simple design of FIR filters with powers-of-two coefficients," *IEEE Trans. Circuits Syst.*, vol. 35, no. 5, pp. 566–570, 1988.

[59] D. Ait-Boudaoud and R. Cemes, "Modified sensitivity criterion for the design of powers-of-two FIR filters," *Electronics Letters*, vol. 29, no. 16, pp. 1467–1469, 1993.

[60] T. Ciloglu and Z. Unver, "A novel method for discrete coefficient FIR digital filter design," in *Circuits and Systems, 1994. ISCAS '94., 1994 IEEE International Symposium on*, vol. 1, 1994, pp. 261–264.

[61] Y. C. Lim and B. Liu, "Design of cascade form FIR filters with discrete valued coefficients," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 36, no. 11, pp. 1735–1739, 1988.

[62] J. Kaiser and R. Hamming, "Sharpening the response of a symmetric nonrecursive filter by multiple use of the same filter," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 25, no. 5, pp. 415–422, 1977.

[63] H. H. Dam, S. Nordebo, K. L. Teo, and A. Cantoni, "Design of linear phase FIR filters with recursive structure and discrete coefficients," in *Proc. ICASSP '98*, vol. 3, 1998, pp. 1269–1272.

[64] O. Gustafsson, H. Johansson, and L. Wanhammar, "An MILP approach for the design of linear-phase FIR filters with minimum number of signed-power-of-two terms," in *Proc. Eur. Conf. Circuit Theory Design*, vol. 2, August 2001, pp. 217–220.

[65] J. Yli-Kaakinen and T. Saramaki, "A systematic algorithm for the design of multiplierless FIR filters," in *Proc. ISCAS '01*, vol. 2, 2001, pp. 185–188.

[66] M. Aktan, A. Yurdakul, and G. Dundar, "An algorithm for the design of low-power hardware-efficient FIR filters," *IEEE Trans. Circuits Syst. I*, vol. 55, no. 6, pp. 1536–1545, 2008.

[67] Y. C. Lim, R. Yang, D. Li, and J. Song, "Signed power-of-two term allocation scheme for the design of digital filters," *IEEE Trans. Circuits Syst. II*, vol. 46, no. 5, pp. 577–584, 1999.

[68] D. Li, Y. C. Lim, Y. Lian, and J. Song, "A polynomial-time algorithm for designing FIR filters with power-of-two coefficients," *IEEE Trans. Signal Process.*, vol. 50, pp. 1935–1941, Aug. 2002.

[69] C.-L. Chen, K.-Y. Khoo, and A. N. Willson, Jr., “An improved polynomial-time algorithm for designing digital filters with power-of-two coefficients,” in *Circuits and Systems, 1995. ISCAS ’95., 1995 IEEE International Symposium on*, vol. 1, 1995, pp. 223–226.

[70] A. J. Llorens, C. N. Hadjicostis, and H. C. Ni, “Quantization of FIR filters under a total integer cost constraint,” *IEEE Trans. Circuits Syst. II*, vol. 52, no. 9, pp. 576–580, 2005.

[71] Y. Lim and S. Parker, “Discrete coefficient FIR digital filter design based upon an LMS criteria,” *IEEE Trans. Circuits Syst.*, vol. 30, no. 10, pp. 723–739, 1983.

[72] J.-J. Shyu and Y.-C. Lin, “A new approach to the design of discrete coefficient FIR digital filters,” *IEEE Trans. Signal Process.*, vol. 43, no. 1, pp. 310–314, 1995.

[73] D. E. Quevedo and G. C. Goodwin, “Moving horizon design of discrete coefficient FIR filters,” *IEEE Trans. Signal Process.*, vol. 53, no. 6, pp. 2262–2267, 2005.

[74] S. R. Powell and P. M. Chau, “Efficient narrowband FIR and IFIR filters based on powers-of-two sigma-delta coefficient truncation,” *IEEE Trans. Circuits Syst. II*, vol. 41, no. 8, pp. 497–505, 1994.

[75] N. Benvenuto, M. Marchesi, and A. Uncini, “Applications of simulated annealing for the design of special digital filters,” *IEEE Trans. Signal Process.*, vol. 40, no. 2, pp. 323–332, 1992.

[76] P. Persson, S. Nordebo, and I. Claesson, “Design of discrete coefficient FIR filters by a fast entropy-directed deterministic annealing algorithm,” *IEEE Trans. Signal Process.*, vol. 53, no. 3, pp. 1006–1014, 2005.

[77] S. S. Rao and K. Chellapilla, “Design of discrete coefficient FIR filters using fast simulated evolutionary optimization,” in *IEEE Int’l Conf. Neural Networks ’96*, vol. 2, 1996, pp. 1185–1190.

[78] S. U. Ahmad and A. Antoniou, “Cascade-form multiplierless FIR filter design using orthogonal genetic algorithm,” in *Signal Processing and Information Technology, 2006 IEEE International Symposium on*, 2006, pp. 932–937.

[79] A. G. Dempster and M. D. Macleod, “Use of minimum adder multiplier blocks in FIR digital filters,” *IEEE Trans. Circuits Syst.*, vol. 42, pp. 569–577, September 1995.

[80] R. Hartley, “Subexpression sharing in filters using canonic signed digit multipliers,” *IEEE Trans. Circuits Syst. II*, vol. 43, no. 10, pp. 677–688, October 1996.

[81] M. Potkonjak, M. B. Srivastava, and A. P. Chandrakasan, “Multiple constant multiplications: Efficient and versatile framework and algorithms for exploring common subexpression elimination,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 15, pp. 151–165, February 1996.

[82] R. Pasko, P. Schaumont, V. Derudder, S. Vernalde, and D. Iuraekova, “A new algorithm for elimination of common subexpressions,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 18, no. 1, pp. 58–68, January 1999.

[83] A. P. Vinod and E. M.-K. Lai, "On the implementation of efficient channel filters for wideband receivers by optimizing common subexpression elimination methods," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 2, pp. 295–304, February 2005.

[84] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, pp. 489–509, Feb. 2006.

[85] ——, "Stable signal recovery from incomplete and inaccurate measurements," *Comm. Pure Appl. Math.*, vol. 59, pp. 1207–1223, Aug. 2006.

[86] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, pp. 33–61, Aug. 1998.

[87] M. A. T. Figueiredo, J. M. Bioucas-Dias, and R. D. Nowak, "Majorization-minimization algorithms for wavelet-based image restoration," *IEEE Trans. Image Process.*, vol. 16, no. 12, pp. 2980–2991, Dec. 2007.

[88] C. R. Berger, S. Zhou, J. C. Preisig, and P. Willett, "Sparse channel estimation for multicarrier underwater acoustic communication: From subspace methods to compressed sensing," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1708–1721, Mar. 2010.

[89] W. U. Bajwa, J. Haupt, A. M. Sayeed, and R. D. Nowak, "Compressed channel sensing: A new approach to estimating sparse multipath channels," *Proc. IEEE*, vol. 98, no. 6, pp. 1058–1076, Jun. 2010.

[90] A. Gomaa and N. Al-Dhahir, "A new design framework for sparse FIR MIMO equalizers," *IEEE Trans. Commun.*, 2011, to appear.

[91] D. H. Johnson and D. E. Dudgeon, *Array signal processing.* Englewood Cliffs, NJ: Prentice-Hall, Inc., 1993.

[92] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis.* Cambridge, UK: Cambridge University Press, 1994.

[93] A. J. Miller, *Subset selection in regression*, 2nd ed. Boca Raton, FL: Chapman & Hall/CRC, 2002.

[94] T. H. Cormen, C. E. Leiserson, R. L. Stein, and C. Stein, *Introduction to Algorithms.* Cambridge, MA: MIT Press, 2001.

[95] D. Bertsimas and R. Weismantel, *Optimization Over Integers.* Belmont, MA: Dynamic Ideas, 2005.

[96] D. Bertsimas and J. N. Tsitsiklis, *Introduction to Linear Optimization.* Nashua, NH: Athena Scientific, 1997.

[97] C. Lemaréchal and F. Oustry, "Semidefinite relaxations and Lagrangian duality with application to combinatorial optimization," INRIA, Tech. Rep. RR-3710, 1999.

[98] D. P. Bertsekas, *Nonlinear Programming.* Belmont, MA: Athena Scientific, 1999.

[99] K. C. Toh, M. J. Todd, and R. H. Tütüncü, "SDPT3 — a MATLAB software package for semidefinite programming," *Optim. Method. Softw.*, vol. 11, pp. 545–581, 1999, latest version available at http://www.math.nus.edu.sg/~mattohkc/sdpt3.html.

[100] R. H. Tütüncü, K. C. Toh, and M. J. Todd, "Solving semidefinite-quadratic-linear programs using SDPT3," *Math. Program., Ser. B*, vol. 95, no. 2, pp. 189–217, 2003.

[101] J. F. Sturm, "Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones," *Optim. Method. Softw.*, vol. 11, pp. 625–653, 1999.

[102] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 1.21," http://cvxr.com/cvx, Apr. 2011.

[103] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Rev.*, vol. 38, no. 1, pp. 49–95, Mar. 1996.

[104] M. J. Todd, "Semidefinite optimization," *Acta Numerica*, vol. 10, pp. 515–560, 2001.

[105] C. Helmberg, F. Rendl, R. J. Vanderbei, and H. Wolkowicz, "An interior-point method for semidefinite programming," *SIAM J. Optim.*, vol. 6, no. 2, pp. 342–361, May 1996.

[106] M. Kojima, S. Shindoh, and S. Hara, "Interior-point methods for the monotone semidefinite linear complementarity problem in symmetric matrices," *SIAM J. Optim.*, vol. 7, no. 1, pp. 86–125, 1997.

[107] R. D. C. Monteiro, "Primal-dual path-following algorithms for semidefinite programming," *SIAM J. Optim.*, vol. 7, no. 3, pp. 663–678, 1997.

[108] Y. E. Nesterov and A. S. Nemirovski, *Interior-point polynomial algorithms in convex programming*. Philadelphia: SIAM Publications, 1994.

[109] P. T. Boufounos and A. V. Oppenheim, "Quantization noise shaping on arbitrary frame expansions," *EURASIP Journal on Applied Signal Processing*, vol. 2006, pp. 1–12, 2006.

[110] L. Rabiner, "Linear program design of finite impulse response (FIR) digital filters," *IEEE Trans. Audio Electroacoust.*, vol. 20, no. 4, pp. 280–288, 1972.

[111] T. Baran, D. Wei, and A. V. Oppenheim, "Linear programming algorithms for sparse filter design," *IEEE Trans. Signal Process.*, vol. 58, pp. 1605–1617, Mar. 2010.

[112] D. Wei, "Non-convex optimization for the design of sparse FIR filters," in *IEEE 15th Workshop on Statistical Signal Processing*, Sep. 2009, pp. 117–120.

[113] L. G. Khachiyan and M. J. Todd, "On the complexity of approximating the maximal inscribed ellipsoid for a polytope," *Math. Program.*, vol. 61, pp. 137–159, 1993.

[114] K. M. Anstreicher, "Improved complexity for maximum volume inscribed ellipsoids," *SIAM J. Optim.*, vol. 13, no. 2, pp. 309–320, 2002.

[115] Y. Zhang and L. Gao, "On numerical solution of the maximum volume ellipsoid problem," *SIAM J. Optim.*, vol. 14, no. 1, pp. 53–76, 2003.

[116] H. L. van Trees, *Optimum Array Processing*, ser. Detection, Estimation, and Modulation Theory. New York: Wiley and Sons, 2002, vol. 4.

[117] O. Hoshuyama, A. Sugiyama, and A. Hirano, “A robust adaptive beamformer for microphone arrays with a blocking matrix using constrained adaptive filters,” *IEEE Trans. Signal Process.*, vol. 47, pp. 2677–2684, Oct. 1999.

[118] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques.* Cambridge, MA: MIT Press, 2009.

[119] P. A. Parrilo, “Semidefinite programming relaxations for semialgebraic problems,” *Math. Program., Ser. B*, vol. 96, no. 2, pp. 293–320, 2003.

[120] J. B. Lasserre, “Global optimization with polynomials and the problem of moments,” *SIAM J. Optim.*, vol. 11, no. 3, pp. 796–817, 2001.

[121] D. Bertsimas and R. Shioda, “Algorithm for cardinality-constrained quadratic optimization,” *Comput. Optim. Appl.*, vol. 43, no. 1, pp. 1–22, 2009.

[122] T. A. Baran, “Design and implementation of discrete-time filters for efficient sampling rate conversion,” Master’s thesis, Massachusetts Institute of Technology, Cambridge, MA, Feb. 2007.

[123] R. P. Stanley, *Enumerative Combinatorics.* Cambridge, UK: Cambridge University Press, 2000, vol. 1.