Aspects of quadratic optimization - nonconvexity, uncertainty, and applications

Marandi, Ahmadreza

Publication date:
2017

Document Version
Publisher's PDF, also known as Version of record

Aspects of Quadratic Optimization

Nonconvexity, Uncertainty, and Applications

Proefschrift

ter verkrijging van de graad van doctor aan Tilburg University op gezag van de rector magnificus, prof.dr. E.H.L. Aarts, in het openbaar te verdedigen ten overstaan van een door het college voor promoties aangewezen commissie in de aula van de Universiteit op

maandag 11 december 2017 om 14:00 uur
door

Ahmadreza Marandi

geboren op 16 september 1989 te Arak, Iran

Promotiecommissie:

Promotores:	prof.dr. Etienne de Klerk prof.dr.ir. Dick den Hertog
Overige leden:	prof.dr. Monique Laurent dr. Wolfram Wiesemann dr. Ruth Misener dr. Juan Vera Lizcano

Aspects of Quadratic Optimization: Nonconvexity, Uncertainty, and Applications

Acknowledgments

Writing acknowledgment is among the most challenging parts of writing a thesis. There have been many people who walked alongside of me to reach this point of my life, and words cannot express my appreciation to their unwavering support.

The first year of my Ph.D. was relatively tough. Since, there was so many things to learn, besides, working in a company as an intern and integrating to the new culture and environment made it more complicated. However, all of my friends, together with my supervisors, enormously helped me to overcome all the obstacles.

Now that I am in the last stage of my Ph.D., I would like to thank my supervisors. Without their consistent guidance, support, and encouragement, this thesis would never have existed. I benefited a lot from their mentorship, not only for my academic life but also for my personal life. Their support was immeasurable, and no word can describe how grateful I am.

I would also like to express my gratitude to the committee members. I appreciate their time and energy for reading my thesis, and I am grateful for their constructive comments that helped me improve the thesis.

I would also like to thank my friends: Ahmad, Alaa, Ali, Amparo, Atiyeh, Bas D., Bas H., Bartosz, Chen, Christos, Ehsan, Elisabeth, Fatemeh, Frans, Hamed, Jan, Jop, Krzysztof, Mahsa, Maria, Mario, Marieke, Marleen, Mohammad H., Mohammad Nasir, Mohammad T., Nick, Olga, Shobeir, Trevor, Victoria, and Zahra. Especially, I thank Frans for being a great office mate, Marleen, Marieke, and Krzysztof for all our tea-time discussions, Trevor for all our conversations/collaborations, Olga for all our scientific discussions and chatter, Shobeir and Christos for all our conversations, and Ehsan for all his support.

Furthermore, I would like to thank Dr. Majid Soleimani (my Bachelor's and Master's thesis supervisor) who introduced me to the world of Optimization, and Dr. Jawad Elomari (my mentor at ORTEC) who was not only my mentor but also my friend.

I wish I could thank my family properly for their love and encouragement. My parents and my siblings always supported me to keep on the hard work and follow my dreams. Furthermore, I would also like to thank my wife's family for their kind support over the last three years.

You may wonder why I have not yet mentioned my beloved wife, Zeinab. I have not thanked her yet because her support was beyond any description. Without her, I could not envision myself finishing my Ph.D. There were times that I felt disheartened, but she was there for me. It was Zeinab who calmed me, and helped me to revive my motivations. To show how much grateful I am to her, I dedicate the thesis to her.

Ahmadreza Marandi
September 2017, Tilburg

To my closest friend, my kindest supporter,
and the love of my life,
Zeinab

Contents

CHAPTER 1

Introduction

1.1 Quadratic optimization

A quadratically constrained quadratic optimization (QCQO) problem can be formulated as the mathematical optimization problem

$$\begin{aligned} \min_{y\in\mathbb{R}^n} \; & y^T A^0 y + {b^0}^T y + c_0 \\ \text{s.t. } & y^T A^j y + {b^j}^T y + c_j \leq 0, \;\; j = 1, ..., m, \end{aligned} \tag{1.1}$$

where, $n, m \in \mathbb{N}$, $A^j \in \mathbb{R}^{n\times n}$, $b^j \in \mathbb{R}^n$, and $c_j \in \mathbb{R}$, $j = 0, ..., m$. A convex QCQO problem, where all the matrices A^j, $j = 0, ..., m$, are positive semi-definite (PSD), is known to be tractable (polynomial-time solvable), see, e.g., [28, 100]. There are different approaches that can be applied to solve a convex QCQO problem. Many methods have been proposed to solve a convex QCQO problems directly (see, e.g., [2, 98]). Besides, since convex QCQO problems are a special case of the class of second-order cone optimization (SOCO) problems, all of the algorithms to solve an SOCO problem, like [10, 107], are applicable to solve a convex QCQO problem.

There are many problems that can be formulated as a QCQO problem; see, e.g., the applications in [91, 98, 104] for convex cases, and applications in [15, 46, 62] for nonconvex ones. In the next section, we briefly describe three applications, each of which belongs to a challenging class of QCQO problems that we consider in this thesis, namely nonconvex QCQO problems and convex ones containing uncertainty. We discuss these classes in Sections 1.3 and 1.4, respectively. More precisely, in Section 1.3 we first briefly talk about the underlying ideas of some existing methods that solve a general QCQO problem. Then, by use of examples, we explain the ideas behind the methods that we will propose in Chapters 2 and 3. In Section 1.4, after a short description of a QCQO problem with uncertainties we describe two standard ways of dealing with the uncertainties. Then, we describe the contributions of Chapters 4 and 5 using examples. A summary of the contributions of this thesis is provided in Section 1.6.

1.2 Applications

In this section, we describe three well-known problems that are considered in this thesis, namely pooling, portfolio choice, and norm approximation problems, as applications of our theoretical results.

1.2.1 Pooling problem

A class of nonconvex QCQO problems that we consider in this thesis is called *pooling problems*. Pooling problems arise in chemical processes, wastewater treatment, and petroleum industries, see, e.g., [5, 16, 42, 80].

Several different optimization formulations have been proposed for a pooling problem in the literature, all of which contain bilinear equality and inequality constraints. The extensive description of a pooling problem and three formulations are presented in Chapter 2, and here we only provide a simple example to illustrate the problem. This example is equivalent to *Haverly1* [70], a well-known pooling problem instance. We develop this simple example further throughout this chapter to illustrate the nonconvexity aspect of the problem, and the techniques we use to solve it.

Example 1.1 *Consider a water supplier who has to meet demands for two types of water, one at a temperature at most* $25°C$*, and another at a temperature at most* $15°C$*. The supplier plans to meet the demands by mixing three types of water with degrees* $30°C$*,* $10°C$*, and* $20°C$*, respectively, using one pool. Figure 1.1 shows the mixing diagram used by the supplier.*

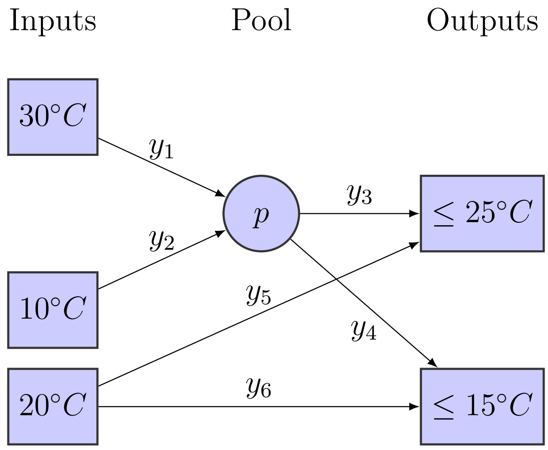


Figure 1.1: The diagram of the illustrative example for a pooling problem.

The temperatures of the water in the outputs depend on the temperature of the water in the pool. For example, assume that y_1 *and* y_2 *kiloliters (kl) of water of degrees*

$30°C$ and $10°C$, respectively, go to the pool. Then, the temperature of the water in the pool, denoted by p, is

$$p = \frac{30y_1 + 10y_2}{y_1 + y_2}.$$

Hence, the flows y_1 and y_2, and the temperature p satisfy the bilinear constraint

$$30y_1 + 10y_2 = p\,(y_1 + y_2)\,. \tag{1.2}$$

Similarly, by setting y_3 and y_5 to be the flows from the pool and the third input to the first output, respectively, the demand for the water of degree at most $25°C$ leads to the bilinear inequality $py_3 + 20y_5 \leq 25(y_3 + y_5)$. □

1.2.2 Portfolio choice problem

The first convex QCQO problem that we consider is called a portfolio choice problem. We explain this problem by the following example.

Example 1.2 *Assume that an investor wants to invest her money in two companies for the year 2018. The annual returns of the companies in the last seven years are as in Table 1.1.*

	2010	2011	2012	2013	2014	2015	2016
Company 1	-1.43	-2.52	1.54	1.31	1.33	3.01	0.77
Company 2	1.07	-2.31	5.39	1.58	0.66	2.26	-0.19

Table 1.1: Information of the return rates (%) of two companies in the seven consecutive years 2010-2016.

According to this information, the mean return vector and the covariance matrix of the return rates of the companies are respectively

$$\mu^T = [0.5729, 1.2086], \quad \Sigma = \begin{bmatrix} 3.6013 & 2.8915 \\ 2.8915 & 5.5640 \end{bmatrix}.$$

The investor wants to find a portfolio that minimizes the overall trade-off between the negative return and risk of the portfolio for the year 2018. Hence, she decides to solve

$$\min_{y \in \mathbb{R}^2} \left\{ \, -\mu^T y + \lambda y^T \Sigma y : \; y_1 + y_2 = 1, \; y \geq 0 \right\}, \tag{1.3}$$

where $\lambda > 0$ is a parameter that is chosen by the investor and controls the trade-off between the mean and the risk. Also, y is the decision variable, which shows how the investor should choose the portfolio for the year 2018, i.e., y_i is the percentage of the *money that will be invested on the ith company, $i = 1, 2$. This way of formulating a portfolio choice problem is first proposed in [94], and known as the Markowitz mean-variance formulation. Let $\lambda = 1$. By solving (1.3), the investor reaches the conclusion of investing 70% of the money on Company 1 and 30% of it on Company 2 (we will call this Portfolio 1). She has reached to this portfolio by estimating μ and Σ based on the historical data in Table 1.1. However, these estimations may have some errors, and the solution may be sensitive to them.*

One of the characteristics of this problem that we use later is concavity (more precisely linearity) of the objective function in μ and Σ. □

1.2.3 Norm approximation and linear regression problems

The last application that we present in this section is a norm approximation problem, which has a convex quadratic formulation. Consider a matrix $A \in \mathbb{R}^{m\times n}$ and a vector $b \in \mathbb{R}^m$. Assume that the goal is to find the closest point to the vector b in the range of the matrix A. Therefore, we are interested in the optimal solution to the norm approximation problem

$$\min_{y\in\mathbb{R}^n} \|Ay - b\|_2, \tag{1.4}$$

where $\|.\|_2$ is the Euclidean norm. Depending on the characteristics of A and b, the optimal solution may be extremely sensitive to a minor error in A and b. Notice that for this problem the objective function is convex in A and b.

Example 1.3 *A particular case of a norm approximation problem is finding a regression line. Consider two random variables X and Y, where Y depends on X. The regression line problem finds a line with the slope of c and the intercept b such that the line $Y = cX + b$ has the least distance to a set of data points with m observations, i.e., (c, b) is an optimal solution to*

$$\begin{aligned} \min_{\substack{\omega\in\mathbb{R}^m, c\in\mathbb{R}\\ b\in\mathbb{R}}} \quad & \|\omega\|_2 \\ s.t. \quad & \omega_i = cX^i + b - Y^i, \ \ i = 1, ..., m, \end{aligned}$$

where $(X^i, Y^i) \in \mathbb{R}^2$, $i = 1, ..., m$, are the data points. Figure 1.2 provides an example of linear regression for a data set. In this example, the red line is the regression line corresponding to the 7 data points.

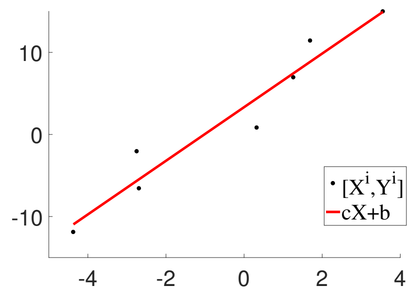


Figure 1.2: Illustrative example for a linear regression problem. The bullets are the data points and the red line is the regression line.

□

1.3 The nonconvexity aspect

In (1.1), if one of the matrices A^j, $j = 0, ..., m$, is not PSD, then the QCQO problem is not convex anymore. There are nonconvex QCQO problems that are proved to be tractable (see, e.g., [39, 73, 82]) or have convex reformulations (see, e.g., [20, 30, 113, 122]). Furthermore, there are nonconvex QCQO problems for which a global optimum can be approximated efficiently (see, e.g., [49, 121]).

In contrast to the tractable cases, there are nonconvex QCQO problems proved to be intractable. A simple example is when A^0 has one negative eigenvalue, and $A^j = 0$, $j = 1, ..., m$ [105]. A class of intractable nonconvex QCQO problems, which is considered in this thesis, is the class of pooling problems [7].

General QCQO problems have been studied in the literature extensively. The recently proposed methods to solve a QCQO problem can be classified into different classes, such as piecewise linear relaxations [4], semi-definite relaxations [77, 108], convex relaxations [11, 123], sum-of-squares (SOS) polynomial relaxations [89], methods for linear complementarity problems [47, 99], and heuristics [90, 106].

The state-of-the-art algorithm for solving a pooling problem is based on the piecewise linear relaxation proposed in [4], and implemented in the software APOGEE [97].

In this thesis, to approximate nonconvex QCQO problems, we modify two recently proposed hierarchies based on SOS polynomial relaxations, called bounded degree sum of squares (BSOS) [88] and sparse-BSOS hierarchies [120], for polynomial optimization (PO) problems. To have a better understanding of the two hierarchies we briefly explain the ideas behind them on an example. The detailed discussions on them are presented in Chapters 2 and 3, respectively.

Example ***1.1 (continued)*** *Assume that the capacity of the tanks storing the water with degrees at most* $25°C$ *and* $15°C$ *are* $100\ kl$ *and* $200\ kl$*, with selling price of* $9\frac{100\$}{kl}$

and $15\frac{100\$}{kl}$*, respectively. Also, assume that the costs of the different input water types with degrees* $30°C$*,* $10°C$*, and* $20°C$ *are* $6\frac{100\$}{kl}$*,* $16\frac{100\$}{kl}$*, and* $10\frac{100\$}{kl}$*, respectively. The supplier wants to find the cheapest flows, and the resulting temperature of the water in the pool that meet the demand. The following problem is called the P-formulation, which formulate the supplier's problem mathematically:*

$$\left.\begin{aligned} \min_{\substack{y\in\mathbb{R}^6\\ p\in\mathbb{R}}} \;& 6y_1 + 16y_2 + y_5 - 5y_6 - 9y_3 - 15y_4 \\ s.t. \;& y_1 + y_2 - y_3 - y_4 = 0, \\ & y_3 + y_5 \le 100, \\ & y_4 + y_6 \le 200, \\ & 30y_1 + 10y_2 - p(y_3 + y_4) = 0, \\ & 20y_5 + py_3 \le 25(y_5 + y_3), \\ & 20y_6 + py_4 \le 15(y_6 + y_4), \\ & y_i \ge 0, \;\; i = 1, ..., 6, \;\; 10 \le p \le 30. \end{aligned}\right\} \tag{1.5}$$

Note that the interpretation of the fourth and fifth constraint was explained earlier and the other constraints may easily be interpreted in the same way. Figure 1.3 shows the graph G *associated with this problem. This graph is constructed as follows: As the nodes, we set* $V = \{y_1, ..., y_7\}$*, where* $y_7 = p$*. The nodes* y_i *and* y_j *are adjacent if* y_i *and* y_j *are present in the definition of at least one constraint.*

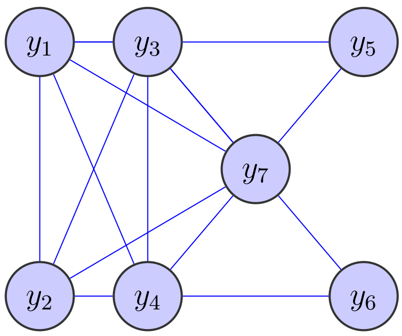


Figure 1.3: The associated graph G corresponding to the problem (1.5).

The maximal complete subgraphs of Graph G *(demonstrated in Figure 1.3) are*

$$\mathcal{D}_1 = \{y_1, y_2, y_3, y_4, y_7\}\,, \;\; \mathcal{D}_2 = \{y_3, y_5, y_7\}\,, \;\; \mathcal{D}_3 = \{y_4, y_6, y_7\}\,.$$

The sets $\mathcal{D}_1$*,* $\mathcal{D}_2$*, and* $\mathcal{D}_3$ *are called maximal cliques in graph theory.* □

The basic idea behind the BSOS and sparse-BSOS hierarchies is adding a type of redundant constraints to (1.5) and approximating its Lagrangian dual. The redundant constraints that are considered are constructed by multiplying d constraints, in the dth level of the hierarchy.

One of the differences in the hierarchies is in the constraint-multiplications used in them. The BSOS hierarchy makes use of all constraint-multiplications in contrast to the sparse-BSOS hierarchy that takes advantage of the maximal cliques of the associate graph. More precisely, the multiplication of two constraints is used in the sparse-BSOS hierarchy if both of them contain variables that are in the same maximal clique. This will be illustrated in the following example.

Example ***1.1 (continued)*** *For problem (1.5) the multiplication*

$$[y_1 + y_2 - y_3 - y_4]\, y_5 \tag{1.6}$$

plays no role in the sparse-BSOS hierarchy, since the set $\{y_1, y_2, y_3, y_4, y_5\}$ *is not a subset of any maximal clique. However, the multiplication (1.6) is involved in the BSOS hierarchy.* □

The major benefit of working with maximal cliques rather than all variables is for sparse problems, where the overlaps in the maximal cliques of the associated graph are small. For these problems, the sparse-BSOS hierarchy has levels that may be solved more efficiently than the ones in the BSOS hierarchy.

1.4 The uncertainty aspect

As it was mentioned, a challenge in optimization is dealing with uncertainties in the parameters, i.e., uncertainties in the matrix A^j, vector b^j, or scalar c_j, $j = 0, ..., m$, in (1.1). Uncertainties in parameters of a QCQO problem may arise from measurement, estimation, and implementation errors. In general, a QCQO problem is prone to any errors (uncertainties) in its parameters. Even a slight change in one parameter value may have an enormous impact on the feasibility or the quality of the solution. Let us give an example to show how the uncertainty may affect a solution.

Example ***1.2 (continued)*** *According to the information in Table 1.1, for Company 1 the return rate got positive from 2012 onward. Let us consider the mean return and covariance matrix of the returns from 2012 until 2016 instead of 2010 until 2016. By using this information, solving (1.3) results in the following portfolio:* 94% *investment on Company 1 and* 6% *on Company 2. We will call it Portfolio* 2.

Let us evaluate the portfolios on the objective function of (1.3) using the mean return vector and covariance matrix of the information from 2010 until 2016, and the information from 2012 until 2016 in the following table.

	2010-2016	2012-2016
Portfolio 1: 70% investment on Company 1 and 30% on Company 2	2.7163	-0.6783
Portfolio 2: 94% investment on Company 1 and 6% on Company 2	2.9173	-0.9004

This table shows that choosing different μ and Σ yields different solutions with noticeably different objective values. □

As it was mentioned in Example 1.2, the imprecise (uncertain) data results in a solution that may not be reliable. To handle this data uncertainty, we use *Robust Optimization* [24], where the only known information about data is a user-specified set that contains the values of the uncertain parameters against which the decision maker wants to be safeguarded. There are two approaches in Robust Optimization dealing with uncertainties: static robust optimization (SRO) [26] and adjustable robust optimization (ARO) [25]. SRO tries to find a solution by considering the worst-case scenario in the uncertainty set. In SRO all decisions are made at the moment of solving the problem ("here and now"), and before realization of the uncertain parameters.

Example *1.2 (continued)* *To compare the SRO with (1.3), let us assume that the possible values of the mean return and covariance matrix are obtained either from the information from 2010 until 2016 or from 2012 until 2016 ($\mathcal{Z} = \{(\mu_1, \Sigma_1), (\mu_2, \Sigma_2)\}$). SRO tries to optimize the problem with respect to the worst-case scenario in the uncertainty set:*

$$\begin{aligned} Opt(SRO) = \min_{\substack{y \in \mathbb{R}^2 \\ t \in \mathbb{R}}} \ & t \\ s.t. \ & -\mu^T y + y^T \Sigma y \leq t, \quad \forall (\mu, \Sigma) \in \mathcal{Z} \\ & y_1 + y_2 = 1, \\ & y_1, y_2 \geq 0. \end{aligned} \tag{1.7}$$

In (1.7) the uncertain parameters appear only in the first constraint. Therefore, the uncertain parameter can be replaced by a worst-case scenario in $\mathcal{Z}$. Since $\mathcal{Z}$ has only two elements, we conclude that the optimal solution of (1.7) is either Portfolio 1 or Portfolio 2. By solving (1.7), we find out that the optimal solution is Portfolio 1, which means it is the best solution considering the worst-case scenario in $\mathcal{Z}$. □

The uncertainty set $\mathcal{Z}$ in Example 1.2 contains only finite number of scenarios. In Chapter 4, we show how to construct a statistical sound uncertainty set based on historical data. This set contains infinitely many scenarios, but we will show that the problem (1.7) is still tractable.

Another approach that deals with uncertainty is ARO. In ARO, a part of the decisions is "here and now." The rest of the decisions will be made after the realization of the uncertain parameter ("wait and see"). We describe the ARO more precisely by means of another example.

Example 1.4 *([43]) A construction company plans to build two subway stations in two districts of a city that are separated by a river; see Figure 1.4.*

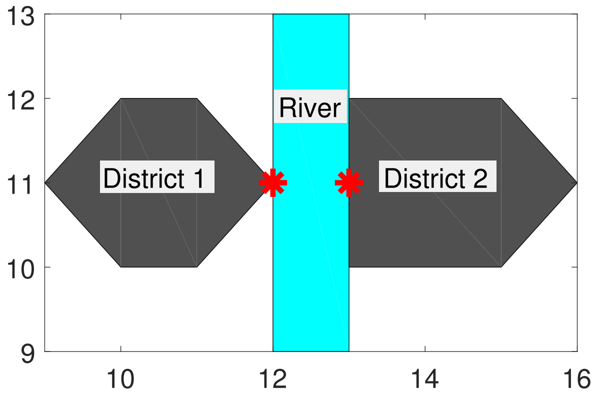


Figure 1.4: Locations of the districts in Example 1.4.

The manager uses the polyhedra $\{x \in \mathbb{R}^2 : A_x x \leq b_x\}$ *and* $\{y \in \mathbb{R}^2 : A_y y \leq b_y\}$ *to locate the districts. She then tries to find the location of the stations in such a way that their distance is as low as possible. Therefore, she decides to solve the following problem:*

$$\begin{aligned} \min_{\substack{x \in \mathbb{R}^2 \\ y \in \mathbb{R}^2}} \ & \|x - y\|_2 \\ s.t. \ & A_x x \leq b_x, \quad A_y y \leq b_y, \end{aligned} \tag{1.8}$$

where x *and* y *are the locations of the first and second station, respectively. Using Figure 1.4, she finds that the locations marked by "*" are the optimal solutions to (1.8). After looking at the previous constructions of the company on District 1, the manager finds out that because of the rocky ground on that area the solution to (1.8) cannot be implemented precisely in the first district and there is always an error (denoted by* ζ_x*) in it, which is between* ℓ_ζ *and* u_ζ*. Therefore, she decides to find the closest locations in Districts 1 and 2 that are immunized against any implementation error in District 1. Thus, she solves the corresponding SRO problem:*

$$\begin{aligned} \min_{\substack{x,y \in \mathbb{R}^2 \\ t \in \mathbb{R}}} \ & t \\ s.t. \ & \|x + \zeta_x - y\|_2 \leq t, \quad \forall \zeta_x \in [\ell_\zeta, u_\zeta] \\ & A_x (x + \zeta_x) \leq b_x, \quad \forall \zeta_x \in [\ell_\zeta, u_\zeta] \\ & A_y y \leq b_y, \end{aligned} \tag{1.9}$$

where $[\ell_\zeta, u_\zeta]$ denotes the box $[\ell_{\zeta_1}, u_{\zeta_1}]\times[\ell_{\zeta_2}, u_{\zeta_2}]$. After talking to a robust optimization expert, the manager notices that (1.9) is conservative. This is because the company first constructs the station in the first district and then according to its location they can find the location of the station in the second district. Thus, the location of the station in the second district is a function of the implementation error, and can be adjusted when the error has been realized. Therefore, she also considers the ARO problem:

$$\begin{aligned}
\min_{x\in\mathbb{R}^2}\ \max_{\ell_\zeta\le\zeta_x\le u_\zeta}\ \min_{\substack{y(\zeta_x)\in\mathbb{R}^2\\ t(\zeta_x)\in\mathbb{R}}}\ & t(\zeta_x)\\
s.t.\ & \|x+\zeta_x-y(\zeta_x)\|_2\le t(\zeta_x),\\
& A_x\,(x+\zeta_x)\le b_x,\\
& A_y y(\zeta_x)\le b_y.
\end{aligned} \tag{1.10}$$

By solving (1.10), the manager gets the optimal decision rule $y^*(\zeta_x)$, which not only yields a better objective value in (1.10) than the optimal value of (1.9), but also helps the manager to make a suitable decision about the location of the second station when the construction of the first station is finished. □

There are some challenges in Examples 1.2 and 1.4, that are general challenges in the field of Robust Optimization. The first challenge, as can be seen in Example 1.2, is the construction of the uncertainty set. There are different ways of constructing an uncertainty set for the mean vector and covariance matrix, e.g., using complicated statistical results [50] or using distance functions such as different norms without utilizing any statistical information [55, 64]. In Chapter 4, contrary to the results in the literature, we make use of some standard statistical results to construct an uncertainty set for (μ, Σ). We prove that, considering this uncertainty set, (1.7) has a tractable reformulation.

The other challenges are in obtaining the optimal values of SRO and ARO problems (we denote them by $Opt(SRO)$ and $Opt(ARO)$, respectively). The usual way in acquiring $Opt(SRO)$ is by making use of duality for each constraint. In this way, the robust counterpart of a constraint, which is mostly written as a maximization of a function over the uncertainty set, is reformulated in such a way that the "maximization" becomes "minimization", and then the "minimization" is reduced to a feasibility problem (see, e.g., [22, 66]).

Example *1.8(continued)* *Consider the second constraint in (1.9). Let us assume that A_x has m rows denoted by A_x^i, $i=1,...,m$. Using duality in linear optimization, for $i=1,...,m$, we have*

$$\max_{\ell_\zeta\le\zeta_x\le u_\zeta} A_x^i\,(x+\zeta_x)=\min_{\lambda_1^i,\lambda_2^i\in\mathbb{R}^2}\left\{A_x^i x+u_\zeta^T\lambda_1^i-\ell_\zeta^T\lambda_2^i : \lambda_1^i-\lambda_2^i=A_x^{i^T},\ \ \lambda_1^i,\lambda_2^i\ge 0\right\}.$$

Additionally, we know that the first constraint is equivalent to

$$\max_{\ell_\zeta \le \zeta_x \le u_\zeta} \|x + \zeta_x - y\|_2 \le t,$$

and the maximum is attained in a corner point of $\ell_\zeta \le \zeta_x \le u_\zeta$. *Therefore, the first constraint in (1.9) is equivalent to*

$$\begin{aligned}
&\left\| x + \begin{bmatrix} \ell_{\zeta_1} \\ \ell_{\zeta_2} \end{bmatrix} - y \right\|_2 \le t, \qquad && \left\| x + \begin{bmatrix} \ell_{\zeta_1} \\ u_{\zeta_2} \end{bmatrix} - y \right\|_2 \le t, \\
&\left\| x + \begin{bmatrix} u_{\zeta_1} \\ u_{\zeta_2} \end{bmatrix} - y \right\|_2 \le t, \qquad && \left\| x + \begin{bmatrix} u_{\zeta_1} \\ \ell_{\zeta_2} \end{bmatrix} - y \right\|_2 \le t.
\end{aligned}$$

Hence, (1.9) is equivalent to

$$\begin{aligned}
\min_{\substack{x,y \in \mathbb{R}^2 \\ t \in \mathbb{R}}} \quad & t \\
s.t. \quad & \left\| x + \begin{bmatrix} \ell_{\zeta_1} \\ \ell_{\zeta_2} \end{bmatrix} - y \right\|_2 \le t, \qquad \left\| x + \begin{bmatrix} \ell_{\zeta_1} \\ u_{\zeta_2} \end{bmatrix} - y \right\|_2 \le t, \\
& \left\| x + \begin{bmatrix} u_{\zeta_1} \\ u_{\zeta_2} \end{bmatrix} - y \right\|_2 \le t, \qquad \left\| x + \begin{bmatrix} u_{\zeta_1} \\ \ell_{\zeta_2} \end{bmatrix} - y \right\|_2 \le t, \\
& A_x^i x + u_\zeta^T \lambda_1^i - \ell_\zeta^T \lambda_2^i \le b_x^i, \qquad \lambda_1^i - \lambda_2^i = A_x^{i^T}, \ i = 1, ..., m, \\
& A_y y \le b_y, \qquad \lambda_1^i, \lambda_2^i \ge 0, \ i = 1, ..., m.
\end{aligned} \tag{1.11}$$

□

The known results in the literature to reformulate an SRO problem corresponding to an uncertain convex QCQO problem can be split into two classes. The first class contains convex uncertainty sets over (A^j, b^j, c_j), $j = 0, ..., m$, in problem (1.1). The second class contains convex uncertainty sets over (L^j, b^j, c_j), where $A^j = L^j L^{j^T}$, $j = 0, ..., m$. We call these classes convex QCQO problems with concave and convex uncertainties, respectively. Surprisingly, the focus of the literature is more on the second class. As one can see, problems (1.3) and (1.4) belong to the first and second class, respectively.

For each of the classes, there are only some specific-structured uncertainty sets that are dealt with in the literature. As an example, for convex QCQO problems with concave uncertainty, tractable reformulations exist only for specific polyhedral and ellipsoidal uncertainty sets. In this thesis, we treat both classes in Chapter 4 considering a broad range of uncertainty sets.

It is known that $Opt(ARO) \le Opt(SRO)$. Therefore, for an uncertain convex QCQO problem, ARO may yield a better objective value than SRO, but, computational intractability is an obstacle in the path of acquiring the optimal value of ARO, even for linear optimization problems [57]. One way of finding an upper bound on the optimal value of an ARO is by solving the corresponding SRO, because even though

ARO is intractable in many cases, the corresponding SRO may be tractable (see, e.g., [24, 66]). This can be seen in Example (1.4), where the ARO problem (1.10) on page 10 is intractable but the SRO problem (1.9) is tractable.

For an uncertain linear optimization (LO) problem, it is shown in [27] that if the uncertain parameters in each constraint are independent of ones in the other constraints, then any optimal solutions to the SRO problem is optimal for the ARO problem; however, there is no such a result for an uncertain QCQO problem. In Chapter 5, we obtain such results not only for a convex QCQO, but also for some general nonlinear optimization problems.

Another way of finding an upper bound on $Opt(ARO)$ is by restricting the "wait and see" variable y to be affine in the uncertain parameter ζ. This approximation, which is called *using affine decision rules*, is tractable for some specific QCQO problems where the functions in the optimization problem are linear in the "wait and see" variables. However, affine decision rules are not efficient, in general, even when at least one of the constraints or objective function is quadratic in "wait and see" variables, such as in (1.10). In the following example, we illustrate the impact of using affine decision rules for the ARO problem (1.10).

Example ***1.8(continued)*** *For the ARO problem (1.10), using affine decision rules means restricting $y(\zeta_x)$ to be affine in ζ_x, i.e., $y(\zeta_x) = u + V\zeta_x$, where $u \in \mathbb{R}^2$ and $V \in \mathbb{R}^{2\times 2}$. Therefore, (1.10) can be approximated by*

$$
\begin{aligned}
\min_{\substack{x,t\\u,V}} \quad & t \\
s.t. \quad & \|x + \zeta_x - u - V\zeta_x\|_2 \leq t, && \forall \zeta_x \in [\ell_\zeta, u_\zeta] \\
& A_x\,(x + \zeta_x) \leq b_x, && \forall \zeta_x \in [\ell_\zeta, u_\zeta] \\
& A_y\,(u + V\zeta_x) \leq b_y, && \forall \zeta_x \in [\ell_\zeta, u_\zeta] \\
& x, u \in \mathbb{R}^2, V \in \mathbb{R}^{2\times 2}, t \in \mathbb{R},
\end{aligned}
\tag{1.12}
$$

which is an SRO problem. As we explained it earlier, the first constraint can be reformulated as a system of four constraints by checking the corner points of the uncertainty set. However, if the dimension of the uncertainty set increases, then this method is not applicable anymore, and (1.12) becomes intractable. We emphasize here that (1.12) belongs to the second class of SRO problems, since the first constraint is convex in the uncertain parameter. □

1.5 Contributions of the thesis

The contributions of the thesis can be split into two parts: ones on approximating a nonconvex QCQO problem such as a pooling problem, and ones on dealing with uncertainties in the parameters of a convex QCQO problem.

1.5.1 The nonconvexity aspect

As it was mentioned, we use the BSOS and sparse-BSOS hierarchies in PO to approximate a general QCQO problem and specially a pooling problem. Here, we list the contributions of the thesis regarding this aspect:

(1) We provide, for the first time in the literature of pooling problems, a systematic method of eliminating all linear and nonlinear equality constraints in a mathematical formulation of a pooling problem.

(2) We make a contribution to the performance improvement of the BSOS hierarchy by reducing the number of variables and constraints in each level of the hierarchy for a general QCQO problem.

(3) We introduce a generalization of the BSOS and sparse-BSOS hierarchies for a general PO problem, where the functions in the problem are all polynomial. This generalization is made to handle problems with equality constraints without increasing the size and destroying the sparsity pattern of the problem, while keeping the convergence results.

(4) The performance assessment of the hierarchies with and without our contributions are carried out on pooling problems. Based on our numerical experiments on some well-known instances and the one constructed in this thesis, we conclude that the contributions have a significant impact on improving the performance of the hierarchies and make them comparable with the state-of-the-art algorithm [97] for small-sized instances.

1.5.2 The uncertainty aspect

The contribution of this thesis in the realm of Robust Optimization is five-fold:

(5) We extend the scope of the robust convex QCQO problems and provide a tractable reformulation of a convex quadratic constraint with concave uncertainty for a vast range of vector and matrix uncertainty sets. This extends the results in the literature, which are only for some specific vector-valued and matrix-valued uncertainty sets.

(6) We construct a new uncertainty set over a vector containing the mean vector and vectorized covariance matrix, using historical data and standard statistical results. The advantage of the uncertainty set compared to the one in [50] is that we do not have a restrictive assumption on the statistical information derived from the historical data. Moreover, we use some standard statistical results,

rather than some complicated ones that are used in [50], to derive the uncertainty set for the mean vector and covariance matrix. In addition to our theoretical results, we assess the effectiveness of this uncertainty set on deriving a tractable reformulation of a robust portfolio choice problem (problem (1.7)) using real-life data.

(7) We provide inner and outer tractable approximations of a convex quadratic constraint with convex uncertainty and compact uncertainty set. The approximations are shown to be tight for a class of problems. Our results can handle a broad class of uncertainty sets, whereas the results in [24, 29, 54, 63] are for specific sets. Moreover, we assess the performance of our approximations by conducting numerical experiments on a norm approximation and a linear regression problem.

(8) We provide conditions under which the optimal solutions to SRO problems are also optimal for the ARO problems not only for an uncertain QCQO problem but also for a general uncertain nonlinear optimization problem. The main assumption for this equivalence is that the uncertain parameters in each constraint are independent of the ones in the other constraints (constraint-wise uncertainty).

(9) We show under some mild assumptions that for problems in which some, but not all, of the uncertain parameters are constraint-wise, there exist optimal solutions to the ARO problems in which the "wait and see" variables are independent of the constraint-wise uncertain parameters. Moreover, we show that for a class of problems, to approximate the ARO problems by using affine decision rules we can restrict the decision rules to be affine in the non-constraint-wise uncertain parameters and constant in the others, and get the same approximation.

1.6 Structure of the thesis and disclosure

This thesis was partially supported by the EU Marie Curie Initial Training Network, grant number 316647 ("Mixed Integer Nonlinear Optimization (MINO)"), and it is based on the following four research papers:

Chapter 2 A. Marandi, J. Dahl, E. de Klerk, "**A numerical evaluation of the bounded degree sum-of-squares hierarchy of Lasserre, Toh, and Yang on the pooling problem**", *Annals of Operations Research* (online first), DOI: 10.1007/s10479-017-2407-5, 2017.

Chapter 3 A. Marandi, E. de Klerk, J. Dahl, "**Solving sparse polynomial optimization problems with chordal structure using the sparse, bounded degree sum-of-squares hierarchy**," Optimization Online, 2017, first revision submitted to *Discrete Applied Mathematics*.

Chapter 4 A. Marandi, A. Ben-Tal, D. den Hertog, B. Melenberg, "**Extending the scope of robust quadratic optimization**," Optimization Online, 2017, Submitted to *Operations Research*.

Chapter 5 A. Marandi, D. den Hertog, "**When are static and adjustable robust optimization problems with constraint-wise uncertainty equivalent?**" *Mathematical Programming*, (online first), DOI: 10.1007/s10107-017-1166-z, 2017.

This thesis is split into two parts. In the first part, which consists of the first two chapters, we present the contributions regarding solving a general QCQO problem. In Chapter 2, after providing some preliminaries, we show how one can eliminate the equality constraints in a formulation of a pooling problem (Contribution 1) using techniques from linear algebra. Then, we explain a procedure to reduce the number of variables and constraints in each level of the BSOS hierarchy (Contribution 2). Chapter 3 contains the generalization of the BSOS and sparse-BSOS hierarchies (Contribution 3). Moreover, we show how one can find a sparsity pattern in a PO problem using some techniques from graph theory. The performance assessments (Contribution 4) of the first two contributions are provided in Chapter 2, and ones corresponding to the third contribution are presented in Chapter 3.

The numerical experiments in this thesis were carried out on an Intel i7-4790 3.60GHz Windows computer with 16GB of RAM in two programming languages. The results in Chapters 2 and 4 are obtained using MATLAB, but the results in Chapter 3 are achieved from Julia [38]. Therefore, to have a fair comparison, a part of the numerical experiments in Chapter 2 are repeated in Chapter 3. The semi-definite optimization (SDO) solver that is used in this thesis is MOSEK 8 [12].

The second part of the thesis, which consists of the last two chapters, contains the contributions regarding a convex QCQO problem with uncertainty. Particularly, in Chapter 4, we show how one can reformulate a convex quadratic constraint with concave uncertainty to a system of convex constraints (Contribution 5). Then, we show how one can approximate a convex quadratic constraint with convex uncertainty in two ways (Contribution 7). At the end, by making use of some standard statistical tools, we construct an uncertainty set over the mean vector and vectorized covariance matrix, based on historical data (Contribution 6). In Chapter 5, we present the results regarding solving and approximating an ARO problem. We provide some conditions for problems with constraint-wise uncertainty under which the SRO and ARO problems have the same optimal value (Contribution 8). Then, we show for problems in which a part of the uncertainty is constraint-wise and not all, that there exists an optimal decision rule for the ARO problem that is independent of the constraint-wise uncertain parameters (Contribution 9).

Part I

Nonconvex Quadratic Optimization

CHAPTER 2

A numerical evaluation of the BSOS hierarchy on the pooling problem

2.1 Introduction

Polynomial optimization (PO) is the class of nonlinear optimization problems involving polynomials only:

$$\begin{aligned} f^* = \inf_{x\in\mathbb{R}^n} \ & f(x) \\ \text{s.t.} \ & g_j(x) \geq 0, \ \ j = 1, ..., m, \end{aligned} \tag{2.1}$$

where f and all g_j are n-variate polynomials. We will assume throughout this chapter that

Assumption I) the feasible set $F = \{x \in \mathbb{R}^n \mid g_j(x) \geq 0, \ \ j = 1, ..., m\}$ is compact;

Assumption II) for all $x \in F$ one has $g_j(x) < 1, \ \ j = 1, ..., m$.

The Assumption II is theoretically without loss of generality. To see this, set

$$M_j := \max\left\{\max_{x\in F} g_j(x), 1\right\}.$$

Therefore, $g_j(x) \leq M_j$ for all $x \in F$. Now, instead of considering (2.1), we consider the following equivalent PO problem:

$$\begin{aligned} f^* = \inf_{x\in\mathbb{R}^n} \ & f(x) \\ \text{s.t.} \ & \frac{\epsilon}{M_j} g_j(x) \geq 0, \ \ j = 1, ..., m, \end{aligned} \tag{2.2}$$

where ϵ is a parameter in $(0, 1)$. Clearly, (2.2) satisfies the assumptions I and II. Note that, in practice, it may be difficult to find the values M_j, or even useful upper bounds on these values.

In general, PO problems are intractable, since they contain problems like the maximum cut problem[1] as special cases; see e.g. [89]. In 2015, Lasserre, Toh and Yang [88] introduced the so-called *bounded degree sum-of-squares (BSOS) hierarchy* to obtain a nondecreasing sequence of lower bounds on the optimal value of problem (2.1) when the feasible set is compact. Each lower bound in the sequence is the optimal value of an SDO problem. Moreover, the authors of [88] showed the following asymptotically convergent results:

Theorem 2.1 *([88]) Consider* (2.1)*, and let Assumption I hold. If* $\{1, g_1, \ldots, g_m\}$ *in* (2.1) *generates the ring of polynomials and* $g_j(x) \leq 1$*,* $j = 1, ..., m$*, for any* $x \in F$*, then the sequence of lower bounds obtained by the BSOS hierarchy (as defined in* (2.5) *below) converges asymptotically to the optimal value of problem* (2.1)*.* □

The authors in [88] from their numerical experiments concluded that the BSOS hierarchy was efficient for quadratic problems.

In this chapter, we analyze the BSOS hierarchy in more detail. We also study variants of the BSOS hierarchy where the number of variables is reduced.

The numerical results in this chapter are on pooling problems, that belong to the class of problems with bilinear functions. The pooling problem is well-studied in the chemical process and petroleum industries. It has also been generalised for application to wastewater networks; see, e.g., [80]. It is a generalization of a minimum cost network flow problem where products possess different specifications. There are many equivalent mathematical models for a pooling problem and all of them include bilinear functions in their constraints. Haverly [70] described the so-called P-formulation, and afterwards many researchers used this model, e.g., [1, 23, 58]. In the recent paper [17], Baltean-Lugojan and Misener show that the P-formulation of the pooling problem instances Haverly1-3 proposed by Haverly [70], which have been considered in the literature as test problems, belong to a class of polynomial-time solvable instances. There are other formulations in the literature for the pooling problem, such as Q-, PQ-, and TP-formulations; in this chapter, we use the P- and PQ-formulations and point the reader to the survey by Gupte et al. [68] where all the formulations are described, as well as the PhD thesis by Alfaki [6].

One way of getting a lower bound for a pooling problem is using convex relaxation, as done, e.g., by Foulds et al. [58]. Similarly, Adhya et al. [1] introduced a Lagrangian approach to get tighter lower bounds for pooling problems. Also, there are many other papers studying duality [23], piecewise linear approximation [97], heuristics for finding a good feasible solution [8], etc. A relatively recent survey on solution techniques is [96].

[1] The maximum cut problem is to find a subset S of vertex set V in a graph $G = (V, E)$ such that the number of edges between S and $V \backslash S$ is as large as possible. It is proved in [79] that a maximum cut problem is $\mathcal{NP}$-complete.

In a seminal paper in 2000, Lassere [85] first introduced a hierarchy of lower bounds for PO problems using SDO relaxations. Frimannslund et al. [60] tried to solve pooling problems with the linear matrix inequality (LMI) relaxations obtained by this hierarchy. They found that, due to the growth of the SDO problem sizes in the hierarchy, this method is not effective for the pooling problems. In this chapter, we therefore consider the BSOS hierarchy as an alternative, since it is not so computationally intensive.

The structure of this chapter is as follows: We describe the BSOS hierarchy in Section 2.2. In Section 2.3 the pooling problem is defined, and we review three mathematical models for it, namely the P-, Q- and PQ-formulations. Also, we solve some pooling problems by the BSOS hierarchy in this section. Section 2.5 contains the numerical results after a reduction in the number of linear variables, using Assumption II, and reduction in the number of constraints in each iteration of the BSOS hierarchy.

2.2 The bounded degree sum of squares hierarchy for polynomial optimization

In this section, we briefly review the background of the BSOS hierarchy from [88]. For easy reference, we will use the same notation as in [88].

In what follows $\mathbb{N}^k$ will denote all k-tuples of nonnegative integers, and we define

$$\mathbb{N}^k_d = \left\{ \alpha \in \mathbb{N}^k : \sum_{i=1}^{k} \alpha_i \leq d \right\}.$$

The space of $n \times n$ symmetric matrices will be denoted by S_n, and its subset of PSD matrices by S_n^+.

Consider the general nonlinear optimization problem (2.1). For fixed $d \geq 1$, the following problem is clearly equivalent to (2.1):

$$\begin{aligned} \min_x \ & f(x) \\ \text{s.t.} \ & \prod_{j=1}^{m} g_j(x)^{\alpha_j}(1-g_j(x))^{\beta_j} \geq 0, \ \ \forall (\alpha,\beta) \in \mathbb{N}^{2m}_d. \end{aligned} \tag{2.3}$$

The underlying idea of the BSOS hierarchy is to rewrite problem (2.1) as

$$f^* = \sup_t \left\{ t : f(x) - t \geq 0 \ \ \forall x \in F \right\}.$$

The next step is to use the following *Positivstellensatz* by Krivine [84] to remove the quantifier '$\forall x \in F$'.

Theorem 2.2 ([84], see also §3.6.4 in [89]) *Assume that $g_j(x) \le 1$ for all $x \in F$ and $j = 1, ..., m$, and $\{1, g_1, \dots, g_m\}$ generates the ring of polynomials. If a polynomial g is strictly positive on F then*

$$g(x) = \sum_{(\alpha,\beta)\in\mathbb{N}^{2m}} \lambda_{\alpha\beta} \prod_{j=1}^{m} g_j(x)^{\alpha_j}(1 - g_j(x))^{\beta_j}$$

for finitely many $\lambda_{\alpha\beta} > 0$. □

Defining

$$h_{\alpha\beta}(x) := \prod_{j=1}^{m} g_j(x)^{\alpha_j}(1 - g_j(x))^{\beta_j}, \quad x \in \mathbb{R}^n,\ \alpha, \beta \in \mathbb{N}^m,$$

we arrive at the following sequence of lower bounds (indexed by d) for problem (2.1):

$$f^* \ge \sup_{\substack{t\in\mathbb{R}\\ \lambda\ge 0}} \left\{ t : f(x) - t = \sum_{(\alpha,\beta)\in\mathbb{N}^{2m}_d} \lambda_{\alpha\beta} h_{\alpha\beta}(x) \quad \forall x \in \mathbb{R}^n \right\}. \tag{2.4}$$

For a given integer $d > 0$ the right-hand-side is a linear optimization (LO) problem, and the lower bounds converge to f^* in the limit as $d \to \infty$, by Krivine's *Positivstellensatz*. This hierarchy of LO bounds was introduced by Lasserre [86].

A subsequent idea, from [87, 88] was to strengthen the LO bounds by enlarging its feasible set as follow: If we fix $\kappa \in \mathbb{N}$, and denote by $\sum[x]_\kappa$ the space of SOS polynomials of degree at most 2κ, then we may define the bounds

$$q^\kappa_d := \sup_{t,\lambda\ge 0} \left\{ t : f(x) - t - \sum_{(\alpha,\beta)\in\mathbb{N}^{2m}_d} \lambda_{\alpha\beta} h_{\alpha\beta}(x) \ \in \Sigma[x]_\kappa \right\}.$$

The resulting problem is an SDO problem, and the size of the PSD matrix variable is determined by the parameter κ, hence the name *bounded-degree sum-of-squares* (BSOS) hierarchy. By fixing κ to a small value, the resulting SDO problem is not much harder to solve than the preceding LO problem, but potentially yields a better bound for given d.

For fixed κ and for each d, one has

$$\begin{aligned} q^\kappa_d = \sup_{t,\lambda,Q}\ & t \\ \text{s.t. }\ & f(x) - \sum_{(\alpha,\beta)\in\mathbb{N}^{2m}_d} \lambda_{\alpha\beta} h_{\alpha\beta}(x) - t = \operatorname{trace}\left(Q v_\kappa(x) v_\kappa(x)^T\right), \quad \forall x \in \mathbb{R}^n \\ & Q \in S^+_{s(\kappa)}, \quad \lambda \ge 0, \end{aligned} \tag{2.5}$$

where $s(\kappa) = \binom{n+\kappa}{\kappa}$, and $v_\kappa(x)$ is a vector with a basis for the n-variate polynomials up to degree κ.

Letting $\tau = \max\{\deg(f), 2\kappa, d\max_j \deg(g_j)\}$, we may eliminate the variables x in two ways to get an SDO:

- Equate the coefficients of the polynomials on both sides of the equality in (2.5), i.e., use the fact that two polynomials are identical if they have the same coefficients in some basis.
- Use the fact that two n-variate polynomials of degree τ are identical if their function values coincide on a finite set of $s(\tau) = \binom{n+\tau}{\tau}$ points in general position.

The second way of obtaining an SDO problem is called the 'sampling formulation', and was first studied in [92]. It was also used for the numerical BSOS hierarchy calculations in [88], with a set of $s(\tau)$ randomly generated points in $\mathbb{R}^n$.

It was proved, e.g. in [102, Theorem 3.1], that two polynomials are identical if they have the same values on the points in

$$\Delta(n,\tau) = \left\{ x \in \mathbb{R}^n \,\middle|\, \tau x \in \mathbb{N}^n,\ \sum_{i=1}^{n} x_i \le 1 \right\},$$

where τ is the largest degree of the polynomials. So, instead of randomly generated points we use $\Delta(n,\tau)$. Thus we obtain the following SDO reformulation of (2.5):

$$\begin{aligned} q_d^\kappa = \sup_{t,\lambda,Q}\ & t \\ & f(x) - \sum_{(\alpha,\beta)\in\mathbb{N}_d^{2m}} \lambda_{\alpha\beta} h_{\alpha\beta}(x) - t = \operatorname{trace}\left(Q v_\kappa(x) v_\kappa(x)^T\right), \forall x \in \Delta(n,\tau) \\ & Q \in S^+_{s(\kappa)},\quad \lambda \ge 0. \end{aligned} \tag{2.6}$$

In the following, we will mention results, proved in [88], that give some information on feasibility and duality issues for the BSOS relaxation. But we first mention a well-known result, which is called *the conic duality theorem.*

Theorem 2.3 *[see, e.g., Theorem 2.4.1 in [28]] Consider the following SDO problem:*

$$P^* := \min_{x\in\mathbb{R}^n} \left\{ c^T x : \ \sum_{j=1}^{n} A^j x_j - B \succeq 0 \right\}, \tag{2.7}$$

for given matrices $B, A^j \in \mathbb{R}^{n\times n}$, $j = 1, ..., n$. Then, the dual of (2.7) is

$$D^* := \max_{\Lambda\in\mathbb{R}^{n\times n}} \left\{ trace\,(B\Lambda) : \ \ trace\left(A^j\Lambda\right) = c_j,\ j = 1, ..., n,\ \ \Lambda \succeq 0 \right\}, \tag{2.8}$$

and we have:

1. *trace* $(B\Lambda) \leq c^T x$ *for any feasible* x *in* (2.7) *and feasible* Λ *in* (2.8)*;*

2. *If* (2.7) *is bounded below and strictly feasible* ($\sum_{j=1}^{n} A^j x_j - B \succ 0$ *for some feasible* x*), then* (2.8) *is solvable and* $P^* = D^*$*;*

3. *If* (2.8) *is bounded above and strictly feasible (there exists* $\Lambda \succ 0$ *such that trace* $(A^j\Lambda) = c_j$ *for all* $j = 1, ..., n$*), then* (2.7) *is solvable and* $P^* = D^*$. □

Now, we are ready to mention the results from [88] about the dual of (2.6).

Theorem 2.4 ([88]) *If the feasible region of problem* (2.1) *contains a solution* $\bar{x}$ *such that* $g_j(\bar{x}) > 0$*, for all* $j = 1, ..., m$ *((2.1) is strictly feasible), then the dual SDO problem of* (2.6) *is strictly feasible.* □

Theorem 2.4 asserts the link between the strictly feasibility of the dual of (2.6) and the PO problem (2.1). Now we mention a straightforward corollary of this theorem for solvability of (2.6).

Corollary 2.1 *Let the problem* (2.1) *be strictly feasible. If the SDO problem* (2.6) *has a feasible solution, it has an optimal solution as well.*

Proof. Theorem 2.4 implies that the dual of (2.6) is strictly feasible. Also, since the SDO problem (2.6) is feasible by assumption, we can conclude that the dual of (2.6) is bounded below, using Theorem 2.3 part 1. Now Theorem 2.3 part 2 implies that (2.6) has an optimal solution. □

Note that problem (2.6) may be infeasible for given d and κ. One only knows that it will be feasible, and therefore q_d^κ will be defined, for sufficiently large d.

Remark 2.1 *Assume that at the* d*-th level of the hierarchy we have* $q_d^\kappa = f^*$*, i.e. finite convergence of the BSOS hierarchy, then*

$$f(x) - f^* = \sum_{(\alpha,\beta)\in\mathbb{N}_d^{2m}} \lambda_{\alpha\beta} h_{\alpha\beta}(x) + v_\kappa(x)^T Q v_\kappa(x) \quad \forall x \in \mathbb{R}^n. \tag{2.9}$$

Let $x^* \in F$ *be an optimal solution (*$f(x^*) = f^*$*), then it is clear from (2.9) that*

$$0 = \sum_{(\alpha,\beta)\in\mathbb{N}_d^{2m}} \lambda_{\alpha\beta} h_{\alpha\beta}(x^*) + v_\kappa(x^*)^T Q v_\kappa(x^*),$$

and due to the fact that Q *is PSD, then*

$$\lambda_{\alpha\beta} h_{\alpha\beta}(x^*) = 0 \quad \forall (\alpha, \beta) \in \mathbb{N}_d^{2m}. \tag{2.10}$$

Hence, for an $(\alpha, \beta) \in \mathbb{N}_d^{2m}$*, if* $h_{\alpha\beta}(x)$ *is not binding at an optimal solution, then* $\lambda_{\alpha\beta} = 0$*. We will use this observation to reduce the number of variables later on.* □

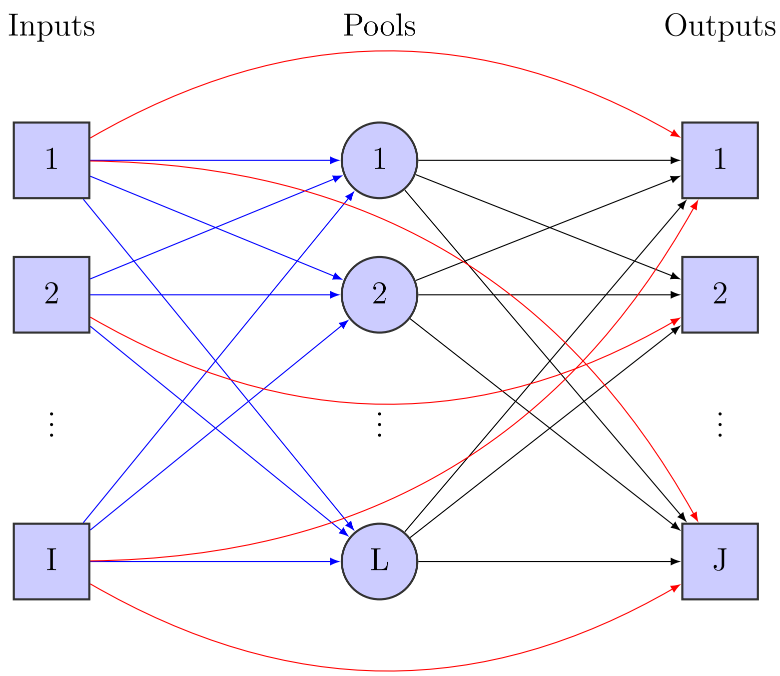


Figure 2.1: An example of a standard pooling problem with I inputs, L pools and J output.

2.3 The P-, Q- and PQ-formulations of the pooling problem

In this section, we describe the P-, Q- and PQ-formulations of the pooling problem. The notation we are using is the same as in [68]. To define the pooling problem, consider an acyclic directed graph $G = (\mathcal{N}, \mathcal{A})$ where $\mathcal{N}$ is the set of nodes and $\mathcal{A}$ is the set of arcs. This graph defines a pooling problem if:

i) the set $\mathcal{N}$ can be partitioned into three subsets $\mathcal{I}, \mathcal{L}$ and $\mathcal{J}$, where $\mathcal{I}$ is the set of inputs with I members, $\mathcal{L}$ is the set of pools with L members and $\mathcal{J}$ is the set of outputs with J members.

ii) $\mathcal{A} \subseteq (\mathcal{I} \times \mathcal{L}) \cup (\mathcal{I} \times \mathcal{J}) \cup (\mathcal{L} \times \mathcal{L}) \cup (\mathcal{L} \times \mathcal{J})$; see Figure 2.1.

In this chapter, we consider cases where $\mathcal{A} \cap \mathcal{L} \times \mathcal{L} = \emptyset$, which is called *standard pooling problem* because there is no arc between the pools.

For each arc $(i, j) \in \mathcal{A}$, let c_{ij} be the cost of sending a unit flow on this arc. For each node, there is possibly a capacity restriction, which is a limit for sum of the incoming (outgoing) flows to a node. The capacity restriction is denoted by C_i for each $i \in \mathcal{N}$. Also, there are some specifications for the inputs, e.g., the sulfur concentrations in them, which are indexed by k in a set of specifications $\mathcal{K}$ with K members. By letting y_{ij} be the flow from node i to node j, u_{ij} the restriction on y_{ij} that can be carried from i to j, and p_{lk} the concentration value of kth specification in the pool l, the pooling problem can be written as the following optimization model:

$$\min_{y,p} \quad \sum_{(i,j)\in\mathcal{A}} c_{ij}y_{ij} \tag{2.11a}$$

s.t.

$$\sum_{\substack{i\in\mathcal{I}:\\(i,l)\in\mathcal{A}}} y_{il} = \sum_{\substack{j\in\mathcal{J}:\\(l,j)\in\mathcal{A}}} y_{lj}, \quad l\in\mathcal{L} \tag{2.11b}$$

$$\sum_{\substack{j\in\mathcal{L}\cup\mathcal{J}:\\(i,j)\in\mathcal{A}}} y_{ij} \le C_i, \quad i\in\mathcal{I} \tag{2.11c}$$

$$\sum_{\substack{j\in\mathcal{J}:\\(l,j)\in\mathcal{A}}} y_{lj} \le C_l, \quad l\in\mathcal{L} \tag{2.11d}$$

$$\sum_{\substack{i\in\mathcal{I}\cup\mathcal{L}:\\(i,j)\in\mathcal{A}}} y_{ij} \le C_j, \quad j\in\mathcal{J} \tag{2.11e}$$

$$0 \le y_{ij} \le u_{ij}, \quad (i,j)\in\mathcal{A} \tag{2.11f}$$

$$\sum_{\substack{i\in\mathcal{I}:\\(i,l)\in\mathcal{A}}} \lambda_{ik}y_{il} = p_{lk}\sum_{\substack{j\in\mathcal{J}:\\(l,j)\in\mathcal{A}}} y_{lj}, \quad l\in\mathcal{L},\ k\in\mathcal{K} \tag{2.11g}$$

$$\sum_{\substack{i\in\mathcal{I}:\\(i,j)\in\mathcal{A}}} \lambda_{ik}y_{ij} + \sum_{\substack{l\in\mathcal{L}:\\(l,j)\in\mathcal{A}}} p_{lk}y_{lj} \le \mu_{jk}^{max}\sum_{\substack{i\in\mathcal{I}\cup\mathcal{L}:\\(i,j)\in\mathcal{A}}} y_{ij}, \quad j\in\mathcal{J},\ k\in\mathcal{K} \tag{2.11h}$$

$$\sum_{\substack{i\in\mathcal{I}:\\(i,j)\in\mathcal{A}}} \lambda_{ik}y_{ij} + \sum_{\substack{l\in\mathcal{L}:\\(l,j)\in\mathcal{A}}} p_{lk}y_{lj} \ge \mu_{jk}^{min}\sum_{\substack{i\in\mathcal{I}\cup\mathcal{L}:\\(i,j)\in\mathcal{A}}} y_{ij}, \quad j\in\mathcal{J},\ k\in\mathcal{K} \tag{2.11i}$$

where μ_{jk}^{max} and μ_{jk}^{min} are the upper and lower bound of the kth specification in output $j\in\mathcal{J}$, and λ_{ik} is the concentration of kth specification in the input i. Here is a short interpretation of the constraints:
(2.11b): volume balance between the incoming and outgoing flows in each pool.
(2.11c): capacity restriction for each input.
(2.11d): capacity restriction for each pool.
(2.11e): capacity restriction for each output.
(2.11f): limitation on each flow.
(2.11g): specification balance between the incoming and outgoing flows in each pool.
(2.11h): upper bound of the output specification.
(2.11i): lower bound of the output specification.

For a general pooling problem, the aforementioned model is called the P-formulation. Consider a pool $l\in\mathcal{L}$ and the arc incident to it from input $i\in\mathcal{I}$. Let us denote by q_{il} the ratio between the flow in this arc and the total incoming flow to this pool.

So, $y_{il} = q_{il}\sum_{j\in\mathcal{J}} y_{lj}$, and $p_{lk} = \sum_{i\in\mathcal{I}} \lambda_{ik}q_{il}$ for any $k \in \mathcal{K}$. Applying these to the P-formulation yields the following problem called the Q-formulation:

$$\begin{aligned}
\min_{y,p} \quad & \sum_{(i,j)\in\mathcal{A}} c_{ij}y_{ij} \qquad (2.12)\\
\text{s.t.} \quad & \\
& (2.11c) - (2.11f)\\
& \sum_{\substack{i\in\mathcal{I}:\\(i,l)\in\mathcal{A}}} q_{il} = 1,\ q_{il} \geq 0, \quad l\in\mathcal{L},\ i\in\mathcal{I},\ (i,l)\in\mathcal{A}\\
& y_{il} = q_{il}\sum_{\substack{j\in\mathcal{J}:\\(l,j)\in\mathcal{A}}} y_{lj}, \quad l\in\mathcal{L},\ i\in\mathcal{I},\ (i,l)\in\mathcal{A}\\
& \sum_{\substack{i\in\mathcal{I}:\\(i,j)\in\mathcal{A}}} \lambda_{ik}y_{ij} + \sum_{\substack{l\in\mathcal{L}:\\(l,j)\in\mathcal{A}}}\sum_{\substack{i\in\mathcal{I}:\\(i,l)\in\mathcal{A}}} \lambda_{ik}q_{il}y_{lj} \leq \mu_{jk}^{max}\sum_{\substack{i\in\mathcal{I}\cup\mathcal{L}:\\(i,j)\in\mathcal{A}}} y_{ij}, \quad j\in\mathcal{J},\ k\in\mathcal{K}\\
& \sum_{\substack{i\in\mathcal{I}:\\(i,j)\in\mathcal{A}}} \lambda_{ik}y_{ij} + \sum_{\substack{l\in\mathcal{L}:\\(l,j)\in\mathcal{A}}}\sum_{\substack{i\in\mathcal{I}:\\(i,l)\in\mathcal{A}}} \lambda_{ik}q_{il}y_{lj} \geq \mu_{jk}^{min}\sum_{\substack{i\in\mathcal{I}\cup\mathcal{L}:\\(i,j)\in\mathcal{A}}} y_{ij}, \quad j\in\mathcal{J},\ k\in\mathcal{K}.
\end{aligned}$$

Adding two sets of redundant constraints

$$y_{lj}\sum_{\substack{i\in\mathcal{I}:\\(i,l)\in\mathcal{A}}} q_{il} = y_{lj}, \quad l\in\mathcal{L},\ j\in\mathcal{J},\ (l,j)\in\mathcal{A}, \tag{2.13a}$$

$$q_{il}\sum_{\substack{j\in\mathcal{J}:\\(l,j)\in\mathcal{A}}} y_{lj} \leq C_l q_{il}, \quad i\in\mathcal{I},\ l\in\mathcal{L},\ (i,l)\in\mathcal{A}, \tag{2.13b}$$

gives an equivalent problem, called the PQ-formulation. It is clear that all formulations are nonconvex quadratic optimization problems which are not easy to solve [69].

2.3.1 McCormick relaxation and the pooling problem

Assume that x and y are variables with given lower and upper bounds

$$\ell_x \leq x \leq u_x, \quad \ell_y \leq y \leq u_y.$$

Then, the following inequalities are implied when $\chi = xy$:

$$\chi \geq \ell_x y + \ell_y x - \ell_x\ell_y, \tag{2.14a}$$

$$\chi \geq u_x y + u_y x - u_x u_y, \tag{2.14b}$$

$$\chi \leq \ell_x y + u_y x - \ell_x u_y, \tag{2.14c}$$

$$\chi \leq u_x y + \ell_y x - u_x\ell_y. \tag{2.14d}$$

It is known that the convex hull of

$$\mathcal{B} := \{(x, y, \chi) \,|\, \chi = xy,\ \ell_x \leq x \leq u_x, \quad \ell_y \leq y \leq u_y\},$$

which is called the McCormick relaxation, is exactly the set of (x, y, χ) that satisfies the inequalities (2.14); see, e.g. [68].

In the pooling problem, the following lower and upper bounds on the variables are implied:

$$\begin{array}{rl} m_\lambda := \min_{i \in \mathcal{I}} \lambda_{ik} \leq p_{lk} \leq M_\lambda := \max_{i \in \mathcal{I}} \lambda_{ik}, & \forall l \in \mathcal{L},\ k \in \mathcal{K}, \\ 0 \leq y_{lj} \leq \min\{C_j, u_{lj}\}, & \forall j \in \mathcal{J},\ l \in \mathcal{L}. \end{array}$$

So, one can get a lower bound by using the McCormick relaxation of each bilinear term in the P- or PQ-formulations.

The redundant constraints (2.13) guarantee that the relaxation obtained by using the McCormick relaxation for the PQ-formulation is stronger than that for the P-formulation (see, e.g., [68] for the proof).

In this chapter, we are going to use the BSOS hierarchy to find a sequence of a lower bounds that approximate the optimal value of the pooling problem. First we analyze the P-formulation and in Section 2.5.3 we compare the results by using the PQ-formulation.

The BSOS hierarchy is only defined for problems without equality constraints and the P-formulation has $(K+1)L$ equality constraints. The simplest way of dealing with equality constraints is to replace each equality constraint by two inequalities; however, this process increases the number of constraints which is not favorable for the BSOS hierarchy. In Chapter 3, we provide a modification of the BSOS hierarchy to deal with the equality constraints directly without increasing the size of the problem. Another way of dealing with the equality constraints is eliminating the equality constraints (2.11b) and (2.11g), if possible.

2.3.2 Eliminating equality constraints

Let $l \in \mathcal{L}$. We assume without loss of generality that the first t inputs feed the pool l. Therefore, equality constraints (2.11b) and (2.11g) can be written as follows:

$$\underbrace{\begin{pmatrix} 1 & 1 & \dots & 1 \\ \lambda_{11} & \lambda_{21} & \dots & \lambda_{t1} \\ \lambda_{12} & \lambda_{22} & \dots & \lambda_{t2} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{1K} & \lambda_{2K} & \dots & \lambda_{tK} \end{pmatrix}}_{A:=} \begin{bmatrix} y_{1l} \\ y_{2l} \\ \vdots \\ y_{tl} \end{bmatrix} = \sum_{\substack{j \in \mathcal{J}: \\ (l,j) \in \mathcal{A}}} y_{lj} \begin{bmatrix} 1 \\ p_{l1} \\ p_{l2} \\ \vdots \\ p_{lK} \end{bmatrix} \qquad \forall l \in \mathcal{L}. \tag{2.15}$$

Let rank$(A) = r$. Applying Singular Value Decomposition [44, Apendix A.5.4], there are matrices $U = [U_1, U_2] \in \mathbb{R}^{(K+1)\times(K+1)}, V = [V_1, V_2] \in \mathbb{R}^{t\times t}, \Sigma \in \mathbb{R}^{r\times r}$ such that

$$A = U \begin{bmatrix} \Sigma & 0_1 \\ 0_2 & 0_3 \end{bmatrix} V^T, U^T U = I, \quad V^T V = I,$$
$$\Sigma = diag(\sigma_1, ..., \sigma_r), \quad \sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r > 0,$$

where $V_1 \in \mathbb{R}^{t\times r}$, $V_2 \in \mathbb{R}^{t\times(t-r)}$, $U_1 \in \mathbb{R}^{(K+1)\times r}$, $U_2 \in \mathbb{R}^{(K+1)\times(K+1-r)}$, $0_1 \in \mathbb{R}^{r\times(t-r)}$, $0_2 \in \mathbb{R}^{(K+1-r)\times r}$, $0_3 \in \mathbb{R}^{(K+1-r)\times(t-r)}$. Therefore, (2.15) can be written as

$$V_1^T \begin{bmatrix} y_{1l} \\ y_{2l} \\ \vdots \\ y_{tl} \end{bmatrix} = \left(\sum_{\substack{j\in\mathcal{J}: \\ (l,j)\in\mathcal{A}}} y_{lj} \right) \Sigma^{-1} U_1^T \begin{bmatrix} 1 \\ p_{l1} \\ p_{l2} \\ \vdots \\ p_{lK} \end{bmatrix} \qquad \forall l \in \mathcal{L}, \tag{2.16}$$

$$0 = \left(\sum_{\substack{j\in\mathcal{J}: \\ (l,j)\in\mathcal{A}}} y_{lj} \right) U_2^T \begin{bmatrix} 1 \\ p_{l1} \\ p_{l2} \\ \vdots \\ p_{lK} \end{bmatrix} \qquad \forall l \in \mathcal{L}. \tag{2.17}$$

The fact that $V^T V = I$, implies that all columns in V, and hence in V_1 are linearly independent. Therefore, taking the QR decomposition of V_1^T, i.e., $V_1^T = Q[R_1, R_2]$, where $R_1 \in \mathbb{R}^{r\times r}$ is upper triangular and invertible, $R_2 \in \mathbb{R}^{r\times(t-r)}$, and $Q \in \mathbb{R}^{r\times r}$ is orthonormal ($Q^T Q = QQ^T = I$), (2.16) is equivalent to

$$\begin{bmatrix} y_{1l} \\ y_{2l} \\ \vdots \\ y_{rl} \end{bmatrix} = R_1^{-1} \left(\left(\sum_{\substack{j\in\mathcal{J}: \\ (l,j)\in\mathcal{A}}} y_{lj} \right) Q^T \Sigma^{-1} U_1^T \begin{bmatrix} 1 \\ p_{l1} \\ p_{l2} \\ \vdots \\ p_{lK} \end{bmatrix} - R_2 \begin{bmatrix} y_{(r+1)l} \\ y_{2l} \\ \vdots \\ y_{tl} \end{bmatrix} \right), \quad \forall l \in \mathcal{L}. \tag{2.18}$$

Concerning (2.17), if for a feasible solution of (2.11), $\sum_{j\in J} y_{lj} = 0$ then it means that there is no outflow from pool l, which implies that there is no input to it and p_{lk}, $k = 1, ..., K$, can attain any real values. So, among all of the possible values for p_{lk} we choose one satisfying

$$0 = U_2^T \begin{bmatrix} 1 \\ p_{l1} \\ p_{l2} \\ \vdots \\ p_{lK} \end{bmatrix}, \tag{2.19}$$

which is a system of K variables and $K-r+1$ linearly independent equalities with $r \geq 1$. Moreover, a feasible solution with the property $\sum_{j \in J} y_{lj} \neq 0$ definitely satisfies (2.19). So, instead of (2.17), we may solve (2.19), which may be done using the QR decomposition.

By solving (2.19), we may write $[p_{lr}, ..., p_{lK}]$ as a linear function of $[p_{l1}, ..., p_{l(r-1)}]$, and by substitution in (2.18), we find the quadratic function corresponding to $[y_{1l}, y_{2l}, \ldots, y_{rl}]$.

Remark 2.2 *We emphasize that after these substitutions, the equivalent mathematical model to the pooling problem is still a nonconvex QCQO problem.*

Remark 2.3 *The interpretation of eliminating equality constraints is as follows when the matrix A is full rank ($rank(A) = \min\{K+1, t\}$): For pools with exactly $K+1$ entering arcs, the entering flow values are given by the total leaving flow and the concentrations in the pool. With more than $K+1$ arcs, say t, $t-K-1$ flow values can be chosen freely and the remaining $K+1$ determined by total leaving flow and concentrations. When $t < K+1$, a basis of t concentration values define the $K+1-t$ remaining ones.*

2.4 First numerical Results

In this section, we study convergence of the BSOS hierarchy of lower bounds q_d^1 ($d = 1, 2, ...$) for pooling problems ($\kappa = 1$). First, it is worth pointing out the number of variables and constraints needed to compute q_d^1. The number of constraints, as it is mentioned in the previous section, is $\binom{n+2d}{2d}$. Also, the number of linear variables is one more than the size of $\mathbb{N}_d^{2m}$, namely $\binom{2m+d}{d} + 1$.

Table 2.1 gives some information of the standard pooling problem instances we use in this chapter. The GAMS files of the pooling problem instances that we use in this chapter, except DeyGupte4, can be found on the website `http://www.ii.uib.no/~mohammeda/spooling/`.

The DeyGupte4 instance is constructed in this section by using the results of [51] as follows. Consider a standard pooling problem with $I = 2$ inputs, $L = 2$ pools and $J = 4$ outputs. Assume that both inputs are connected to the pools and both pools are connected to the outputs (see Figure 2.2). Let $K = 2$ and the concentration of specifications be $(1, 0)$ and $(0, 1)$ for the first and second input, respectively. We number the inputs by $1, 2$, pools by $3, 4$, and outputs by $5, 6, 7, 8$. Let $\mu_{jk}^{max} = \mu_{jk}^{min}$ (given in Figure 2.2), $u_{il} = 4$ and $u_{lj} = 1$, for $i = 1, 2$, $l = 3, 4$, $j = 5, 6, 7, 8$, and $k = 1, 2$. Set the capacity of inputs, pools, and outputs to $C_1 = C_2 = 8$, $C_3 = C_4 = 4$, and $C_5 = C_6 = C_7 = C_8 = 1$. Let

$$\delta := \min\{\|\mu_{\hat{j}k}^{max} - \mu_{\bar{j}k}^{max}\|_2 \ : \ \hat{j} \neq \bar{j} \quad \hat{j}, \bar{j} = 5, 6, 7, 8, \ k = 1\} \approx 0.014,$$

	optimal value [7], [51]	PQ-linear relaxation value [7], [51]	I	J	L	K	# var.	# const.
Haverly1	-400.00	-500.00	3	2	1	1	5	11
Haverly2	-600.00	-1,000.00	3	2	1	1	5	11
Haverly3	-750.00	-800.00	3	2	1	1	5	11
Ben-Tal4	-450.00	-550.00	4	2	1	1	6	13
Ben-Tal5	-3,500.00	-3,500.00	5	5	3	2	29	54
DeyGupte4	-1.00	[-4,-3]	2	4	2	2	10	52
Foulds2	-1,100.00	-1,100.00	6	4	2	1	18	38
Foulds3	-8.00	-8.00	11	16	8	1	152	219
Foulds4	-8.00	-8.00	11	16	8	1	152	219
Adhya1	-549.80	-840.27	5	4	2	4	11	41
Adhya2	-549.80	-574.78	5	4	2	6	11	53
Adhya3	-561.05	-574.78	8	4	3	6	17	66
Adhya4	-877.6.	-961.93	8	5	2	4	16	51
RT2	-4,391.83	-6,034.87	3	3	2	8	14	67
sppA0	Unknown *	-37,772.75	20	15	10	24	161	816

Table 2.1: Details for some well-known pooling problem instances.
* The optimal value for this instance is not known, and it lies in $[-36233.40, -35812.33]$, using [7] and NEOS server [48].

$c_{il} = 0$, for $i = 1, 2$ and $l = 3, 4$. Set $c_{3j} = -1$, $c_{4j} = \frac{2}{\delta}$, for all $j = 5, 6, 7, 8$, and the rest of the costs as 0.

The optimal value of this problem is -1 with the optimal solution constructed by sending flows from inputs to the first pool, and from it to one of the outputs such that the restriction in the specification in it is satisfied [51]. As an example, one of the optimal solutions is constructed by sending 0.13 and 0.87 unit flow from the first and second inputs, respectively, to the first pool, and then 1 unit flow from the first pool to the first output.

DeyGupte4 is constructed to show that a specific class of approximations of the bilinear terms in the PQ-formulation, including the McCromick relaxation, provides bounds far from the optimal value. In particular, one can show the following:

Theorem 2.5 *([51]) Consider a scaled PQ-formulation of the DeyGupte4 instance where each variable lies in* $[0, 1]$. *Let* $g, h : [0, 1] \times [0, 1] \to \mathbb{R}$ *be piecewise linear functions such that*

$$g(\alpha, \beta) \geq \alpha\beta \geq h(\alpha, \beta), \quad \forall \alpha, \beta \in [0, 1].$$

Also, let us replace any bilinear term $\alpha\beta$ *in the scaled PQ-formulation by a new*

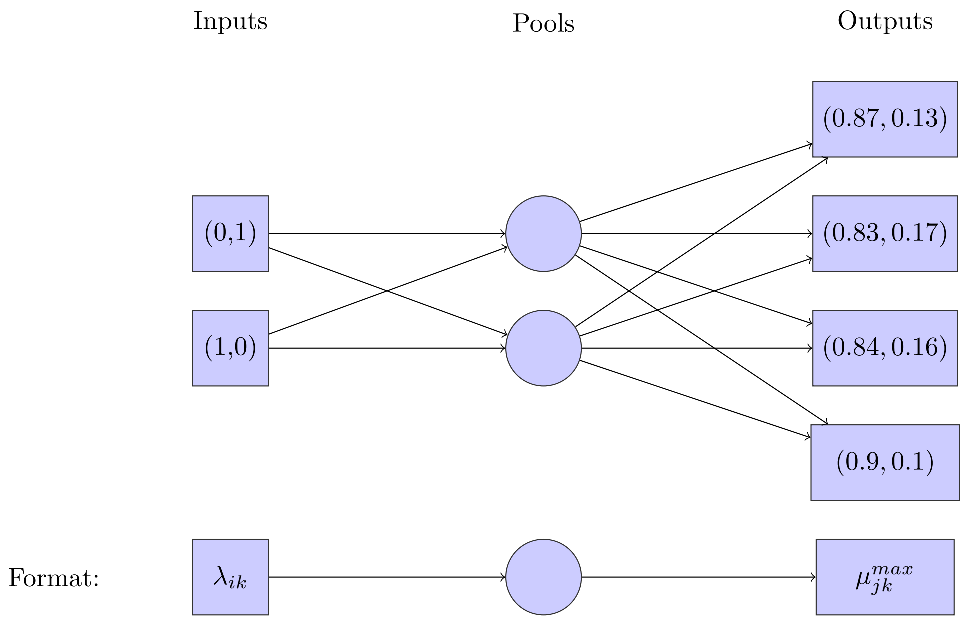


Figure 2.2: The diagram of the DeyGupte4 instance.

variable χ *and add the following constraints to it:*

$$g(\alpha, \beta) \geq \chi \geq h(\alpha, \beta).$$

If $(\alpha, \beta, \alpha\beta + e), (1 - \alpha, \beta, (1 - \alpha)\beta + e) \in \mathcal{S}$ *for* $\alpha = 0.87$ *and* $\beta = 0.67$*, and any* $|e| \leq 0.054$*, where*

$$\mathcal{S} = \{(\alpha, \beta, \chi) \,|\, g(\alpha, \beta) \geq \chi \geq h(\alpha, \beta),\ \alpha, \beta \in [0, 1]\},$$

then the optimal value of the reformulation is in $[-4, -3]$*.* □

Remark 2.4 *The restriction that we have put here on the approximation* $\mathcal{S}$ *is weaker than what we imposed in our paper [93], which restricted the approximation* $\mathcal{S}$ *to be such that, for any* $\alpha, \beta \in [0, 1]$ *and* $|e| \leq 0.05$*, the point* $(\alpha, \beta, \alpha\beta + e)$ *lies in* $\mathcal{S}$*. Clearly, McCormick relaxation does not satisfies this assumption, since the piecewise under and over estimators that we get from it touch the manifold* $\mathcal{B}$ *on page 28 at some points. The weaker assumption in this section, however, holds for the McCormick relaxation, even when the box* $[0, 1] \times [0, 1]$ *is split into the four boxes*

$$\left[0, \frac{1}{2}\right] \times \left[0, \frac{1}{2}\right] \bigcup \left[0, \frac{1}{2}\right] \times \left[\frac{1}{2}, 1\right] \bigcup \left[\frac{1}{2}, 1\right] \times \left[0, \frac{1}{2}\right] \bigcup \left[\frac{1}{2}, 1\right] \times \left[\frac{1}{2}, 1\right].$$

□

In Table 2.1, we recall in column "PQ-linear relaxation value" the lower bound proposed in [7] of each instance. This lower bound is the optimal value of the

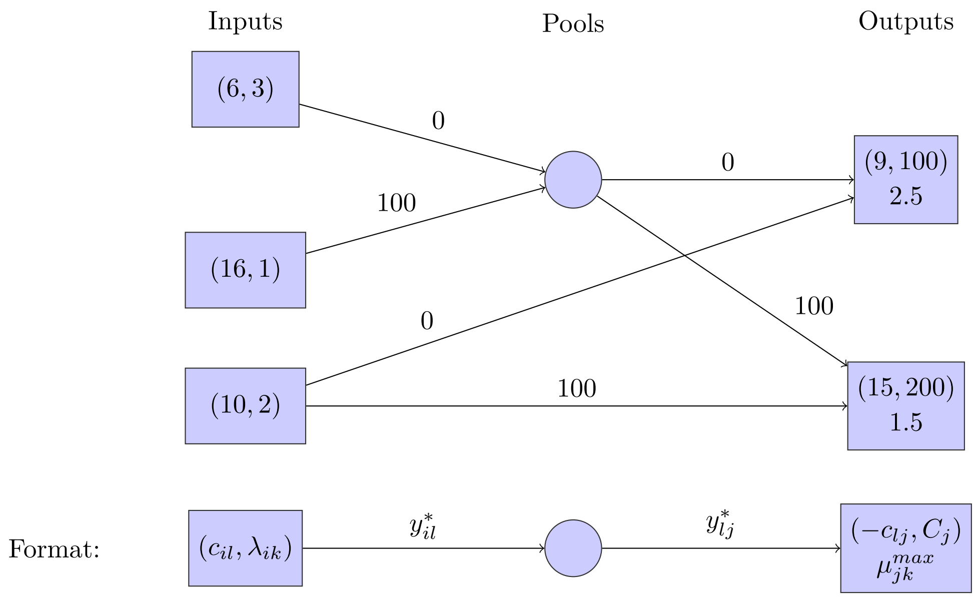


Figure 2.3: Optimal solution for Haverly1

PQ-formulation after applying McCormick relaxation for each bilinear term. Furthermore, columns "# var." and "# const." in this table represents the number of variables and constraints in the P-formulation after eliminating equality constraints, respectively.

Example 2.1 *By way of example, we give the details for the first instance in Table 2.1, called Haverly1. Its optimal solution is shown in Figure 2.3, and the optimal value is* -400 *[1]. The optimal flow from node* i *to node* j *is denoted by* y^*_{ij} *in Figure 2.3.*

This instance has three inputs (denoted by $1,\ 2,\ 3$*), one pool (denoted by* 4*), two outputs (denoted by* $5,\ 6$*), and one specification. The mathematical model for this instance is as follows:*

$$\begin{aligned}
\min \quad & 6y_{14} + 16y_{24} + 10\left[y_{35} + y_{36}\right] - 9\left[y_{45} + y_{35}\right] - 15\left[y_{46} + y_{36}\right] \\
s.t. \quad & y_{14} + y_{24} = y_{45} + y_{46}, \\
& 0 \le y_{45} + y_{35} \le 100, && \text{(2.20a)} \\
& 0 \le y_{46} + y_{36} \le 200, && \text{(2.20b)} \\
& 3y_{14} + y_{24} = p_1\left[y_{45} + y_{46}\right], \\
& 2y_{35} + p_1 y_{45} \le 2.5\left[y_{35} + y_{45}\right], && \text{(2.20c)} \\
& 2y_{36} + p_1 y_{46} \le 1.5\left[y_{36} + y_{46}\right], && \text{(2.20d)} \\
& y_{ij} \ge 0,\ \ p_1 \ge 0.
\end{aligned}$$

So, we can use the elimination method described in the previous section, which implies *that* $y_{14} = \frac{1}{2}(y_{45}+y_{46})(p_1-1)$, $y_{24} = \frac{1}{2}(y_{45}+y_{46})(3-p_1)$. *Therefore, the reformulated model of this instance using scaling* $x_1 := \frac{p_1}{3}$, $x_2 := \frac{y_{45}}{200}$, $x_3 := \frac{y_{46}}{200}$, $x_4 := \frac{y_{35}}{200}$, $x_5 := \frac{y_{36}}{200}$, *is*

$$
\begin{aligned}
\min \quad & -200x_2(15x_1-12) - 200x_3(15x_1-6) + 200x_4 - 1000x_5 \\
s.t. \quad & 1 \geq -\frac{3}{4}(x_1-1)(x_2+x_3) \geq 0 && \text{(2.21a)} \\
& 1 \geq \frac{1}{4}(3x_1-1)(x_2+x_3) \geq 0 && \text{(2.21b)} \\
& 1 \geq 1-2(x_2+x_4) \geq 0 && \text{(2.21c)} \\
& 1 \geq 1-(x_3+x_5) \geq 0 && \text{(2.21d)} \\
& 1 \geq \frac{1}{2}(x_4+x_2) - \frac{2}{5}x_4 - \frac{3}{5}x_1x_2 \geq 0 && \text{(2.21e)} \\
& 1 \geq \frac{1}{2}(x_5+x_3) - \frac{2}{3}x_5 - x_1x_3 \geq 0 && \text{(2.21f)} \\
& 1 \geq x_i \geq 0, \quad i = 1, ..., 5,
\end{aligned}
$$

where the leftmost inequalities are redundant, (2.21a) and (2.21b) are from the sign constraints after the elimination, (2.21c), (2.21d), (2.21e), and (2.21f) are from (2.20a), (2.20b), (2.20c) and (2.20d), respectively.

The last step is to multiply the constraint functions by a factor 0.9 *(any value in* $(0, 1)$ *will do, but we used* 0.9 *for our computations), to ensure that the '*≤ 1*' conditions hold with strict inequality on the feasible set. Thus, we define* $g_1(x) = -0.9 \cdot \frac{3}{4}(x_1-1)(x_2+x_3)$, *etc.*

We will use the BSOS hierarchy to find the optimal value of this example (Table 2.2 below). □

The results of applying the BSOS hierarchy to *Haverly1* and the other pooling problem instances are listed in Table 2.2. "Numerical Prob." and "≈" in the tables mean the solver reported a numerical problem, and only obtained an approximate optimal value, respectively. For the model construction, we have put a time limit of 5 hours. In all the tables from now on, columns "# lin. var.", "size of SD var." and "# const." present the number of linear variables, the size of the PSD matrix variable and the number of constraints in the hierarchy (2.6). Also, "-" in the tables means that the time limit for the model construction has been reached.

As it is clear from Table 2.2, in order to compute q_d^κ we can have a large number of linear variables and constraints (depending of d), which affects the speed and the time we need to solve (2.6). In the coming section, we describe how one can reduce the number of linear variables and constraints at each level of the BSOS hierarchy significantly.

	iteration	time	solution	# lin. var.	size of SD var.	# const.
Haverly1	d=1	0.01s	-600.00	23	6	21
	d=2	0.03s	-417.20	276	6	126
	d=3	0.47s	**-400.00**	2,300	6	462
Haverly2	d=1	0.01s	-1,200	23	6	21
	d=2	0.03s	-601.67	276	6	126
	d=3	0.39s	**-600.00**	2,300	6	462
Haverly3	d=1	0.02s	-875.00	23	6	21
	d=2	0.03s	**-750.00**	276	6	126
Ben-Tal4	d=1	0.02s	-650.00	27	7	28
	d=2	0.03s	-467.20	378	7	210
	d=3	1.44s	**-450.00**	3,654	7	924
Ben-Tal5	d=1	0.06s	**-3,500.00**	109	30	465
DeyGupte4	d=1	0.02s	-4.00	105	11	66
	d=2	4.60s	-3.86	5,565	11	1,001
	d=3	-	-	198,485	11	8,008
Foulds2	d=1	0.01s	-1,200.00	77	19	190
	d=2	109.20s	-1,191.30	3,003	19	7,315
	d=3	-	-	79,079	19	134,596
Foulds3	d=1	90.84s	**-8.00**	439	153	11,781
Foulds4	d=1	92.85s	**-8.00**	439	153	11,781
Adhya1	d=1	0.02s	-999.32	83	12	78
	d=2	4.26s	≈-723.94	3,486	12	1,365
	d=3	-	-	98,770	12	12,376
Adhya2	d=1	0.02s	-798.29	107	12	78
	d=2	11.51s	≈-576.82	5,778	12	1,365
	d=3	-	-	209,934	12	12,376
Adhya3	d=1	0.03s	-882.84	133	18	171
	d=2	135.39s	≈-802.89	8,911	18	5,985
	d=3	-	-	400,995	18	100,947
Adhya4	d=1	0.02s	-1,055.00	103	17	153
	d=2	52.59s	≈-1,035.00	5,356	17	4,845
	d=3	-	-	187,460	17	74,613
RT2	d=1	0.02s	-45,420.50	135	15	120
	d=2	30.84s	-36,542.19	9,180	15	3,060
	d=3	-	-	419,220	15	38,760
sppA0	d=1	273.00s	-47,675.00	1,633	162	13,203
	d=2	-	-	1,334,161	162	29,772,765

Table 2.2: Results for computing the lower bounds q_d^1 for various pooling problem instances.

2.5 Reduction in the number of linear variables and constraints

In this section, we propose a method to reduce the number of linear variables and an upper bound for the number of linearly independent constraints in each iteration of the BSOS hierarchy.

2.5.1 Reduction in the number of variables

As it is mentioned in Remark 2.1, if we can identify constraints that are not binding at optimality, then we can reduce the number of variables.

In particular, because of Assumption II on page 19 the constraints $g_j(x) \leq 1$ will never be binding at optimality. Recalling that the variable $\lambda_{\alpha\beta}$ corresponds to

$$h_{\alpha\beta}(x) := \prod_{j=1}^{m} g_j(x)^{\alpha_j}(1 - g_j(x))^{\beta_j}, \quad x \in \mathbb{R}^n,$$

we know from Remark 2.1 that, in case of finite convergence, we will have $\lambda_{\alpha\beta} = 0$ whenever $\alpha = 0$.

Hence, instead of solving (2.6) to compute q_d^κ, we may compute the following (weaker) bound more efficiently:

$$\begin{aligned} \hat{q}_d^\kappa := &\sup_{t,\lambda,Q} t \\ &f(x) - \sum_{\substack{(\alpha,\beta)\in\mathbb{N}_d^{2m} \\ \alpha\neq 0}} \lambda_{\alpha\beta} h_{\alpha\beta}(x) - t = \operatorname{trace}\left(Q v_\kappa(x) v_\kappa(x)^T\right), \ \forall x \in \Delta(n,\tau), \\ &Q \in S_{s(\kappa)}^+, \quad \lambda \geq 0. \end{aligned} \tag{2.22}$$

The advantage of (2.22) is that it has $\binom{m+d}{d}$ fewer nonnegative variables than (2.6). We emphasize that problem (2.22) is not equivalent to (2.6), i.e., the lower bounds q_d^κ and $\hat{q}_d^\kappa$ are not equal in general — the bound $\hat{q}_d^\kappa$ is weaker, and may be strictly weaker.

The precise relation of the bounds q_d^κ and $\hat{q}_d^\kappa$ is spelled out in the next theorem, which follows from the argument in Remark 2.1.

Theorem 2.6 *If, for given d and κ, q_d^κ and $\hat{q}_d^\kappa$ are both finite, then $\hat{q}_d^\kappa \leq q_d^\kappa$. Moreover, if the sequence q_d^κ $(d = 1, 2, \ldots)$ from (2.6) converges finitely to f^*, then so does $\hat{q}_d^\kappa$ $(d = 1, 2, \ldots)$ from (2.22). More precisely, if $q_{d^*}^\kappa = f^*$ for some $d^* \in \mathbb{N}$, then $\hat{q}_{d^*}^\kappa = f^*$.*

It is important to note that finite convergence of the sequence q_d^κ $(d = 1, 2, \ldots)$ is not guaranteed in general. Sufficient conditions for finite convergence are described in [88].

The numerical results for using (2.22) for the pooling problem instances is demonstrated in Table 2.3. The "rel. time" column from this table onward gives the solution time for each level of the hierarchy as a ratio of that in Table 2.2, which shows that there is a significant reduction in computational times when compared to Table 2.2. This time reduction is because of the smaller number of variables in (2.22). Regarding the quality of the lower bounds that we get from (2.22), even though $\hat{q}_d^\kappa$ could theoretically be weaker than q_d^κ, but for the pooling problem instances we get the same values.

2.5.2 Reduction in the number of constraints

From now on we fix $\kappa = 1$ and $v_1(x) = (1, x_1, ..., x_n)$. As it was mentioned, the number of constraints in each level of the BSOS hierarchy is $\binom{n+2d}{2d}$, where n is the number of variables in the original problem (2.1) and d is the level of the BSOS hierarchy. So, the number of constraints increases quickly with d. In this subsection, we discuss the redundancy of constraints and how we can remove linearly dependent constraints.

Let svec denote the map from the $(n+1) \times (n+1)$ symmetric matrix space S_{n+1} to $\mathbb{R}^{1\times\binom{n+2}{2}}$ given by

$$\mathrm{svec}(X) = \left[X_{11}, \sqrt{2}X_{12}, X_{22}, ..., \sqrt{2}X_{n(n+1)}, X_{(n+1)(n+1)}\right], \quad \forall X \in S_{n+1}.$$

It will also be convenient to number the elements of $\Delta(n, \tau)$ as $x^1, \ldots, x^L$ where $L = s(\tau)$. Finally, we will use the notation $|\beta| = \sum_i \beta_i$.

So, for $d \geq 1$ and $\kappa = 1$ we may write the linear equality constraints in (2.6) as $H_d y_d = b_d$, where

$$H_d = \begin{bmatrix} 1 & (h_{\alpha\beta}(x^1))_{(\alpha,\beta)\in\mathbb{N}_d^{2m}} & \mathrm{svec}\left(v_1(x^1)v_1(x^1)^T\right) \\ \vdots & \vdots & \vdots \\ 1 & \left(h_{\alpha\beta}(x^L)\right)_{(\alpha,\beta)\in\mathbb{N}_d^{2m}} & \mathrm{svec}\left(v_1(x^L)v_1(x^L)^T\right) \end{bmatrix},$$

$$b_d = \begin{bmatrix} f(x^1) \\ \vdots \\ f(x^L) \end{bmatrix}, \quad y_d = \begin{bmatrix} t \\ (\lambda_{\alpha\beta})_{(\alpha,\beta)\in\mathbb{N}_d^{2m}} \\ \mathrm{svec}(Q)^T \end{bmatrix},$$

and $L = \binom{n+2d}{2d}$. It is clear that $H_d \in \mathbb{R}^{L\times\left[\binom{2m+d}{d}+L+1\right]}$.

In the following theorem we prove that all the constraints are linearly independent when $d = 1$.

	iteration	rel. time	solution	# lin. var.	size of SD var.	# const.
Haverly1	d=1	1	-600.00	11	6	21
	d=2	1	-417.20	198	6	126
	d=3	0.60	**-400.00**	1,936	6	462
Haverly2	d=1	1	-1,200.00	11	6	21
	d=2	1	-601.67	198	6	126
	d=3	0.79	**-600.00**	1,936	6	462
Haverly3	d=1	1	-875.00	11	6	21
	d=2	1	**-750.00**	198	6	126
Ben-Tal4	d=1	1	-650.00	14	7	28
	d=2	1	-467.20	274	7	210
	d=3	0.73	**-450.00**	3,095	7	924
Ben-Tal5	d=1	0.83	**-3,500.00**	55	30	465
DeyGupte4	d=1	1	-4.00	53	11	66
	d=2	0.62	-3.86	4,135	11	1,001
	d=3	-	-	172,250	11	8,008
Foulds2	d=1	1	-1,200.00	39	19	190
	d=2	0.85	-1,191.29	2,224	19	7,315
	d=3	-	-	66,419	19	134,596
Foulds3	d=1	0.94	**-8.00**	220	153	11,781
Foulds4	d=1	0.92	**-8.00**	220	153	11,781
Adhya1	d=1	1	-999.32	42	12	78
	d=2	0.95	$\approx$-723.94	2,583	12	1,365
	d=3	-	-	85,526	12	12,376
Adhya2	d=1	1	-798.29	54	12	78
	d=2	0.55	$\approx$-576.82	4,293	12	1,365
	d=3	-	-	182,214	12	12,376
Adhya3	d=1	1	-882.84	67	18	171
	d=2	0.69	$\approx$-802.82	6,634	18	5,985
	d=2	-	-	348,601	18	100,947
Adhya4	d=1	1	-1,055.00	52	17	153
	d=2	0.71	$\approx$-1,035.10	3,979	17	4,845
	d=3	-	-	162,657	17	74,613
RT2	d=1	1	-45,420.48	68	15	120
	d=2	0.65	-36,542.06	6,836	15	3,060
	d=3	-	-	419,220	15	38,760
sppA0	d=1	0.99	-47,6750.00	817	162	13,203
	d=2	-	-	1,000,008	162	29,772,765

Table 2.3: Results for computing the lower bounds $\hat{q}^1_d$ for pooling problem instances using (2.22).

Theorem 2.7 *For the general problem (2.1) with quadratic functions $f(x)$ and $g_j(x)$, $j = 1, ..., m$, all of the constraints in the first iteration of the BSOS hierarchy are linearly independent, i.e. if $d = 1$, all of the constraints of (2.6) are linearly independent.*

Proof. Fix $d = 1$, which implies $\tau = 2$ and $L = \binom{n+2}{2}$ in (2.6). Then,

$$H_1 = \begin{bmatrix} 1 & g_1(x^1) & \dots & g_m(x^1) & 1-g_1(x^1) & \dots & 1-g_m(x^1) & \mathrm{svec}\left(v_1(x^1)v_1(x^1)^T\right) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & g_1(x^L) & \dots & g_m(x^L) & 1-g_1(x^L) & \dots & 1-g_m(x^L) & \mathrm{svec}\left(v_1(x^L)v_1(x^L)^T\right) \end{bmatrix},$$

and,

$$b_1 = \begin{bmatrix} f(x^1) \\ \vdots \\ f(x^L) \end{bmatrix}, \quad y_1 = \begin{bmatrix} t \\ (\lambda_{\alpha\beta})_{(\alpha,\beta)\in\mathbb{N}_1^{2m}} \\ \mathrm{svec}(Q)^T \end{bmatrix},$$

for $x^1, ..., x^L \in \Delta(n, 2)$, defined in (2.6). To show that all of the rows in H_1 are linearly independent, we prove that the submatrix

$$V_n^1 = \begin{bmatrix} \mathrm{svec}\left(v_1(x^1)v_1(x^1)^T\right) \\ \vdots \\ \mathrm{svec}\left(v_1(x^L)v_1(x^L)^T\right) \end{bmatrix} \in \mathbb{R}^{\binom{n+2}{2}\times\binom{n+2}{2}} = \mathbb{R}^{L\times L},$$

is a full rank matrix by induction over n, the dimension of x. Assume that $n = 1$. Because $\Delta(1, 2) = \{0, \frac{1}{2}, 1\}$, it is clear that the rank of the matrix $V_1^1 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & \frac{\sqrt{2}}{2} & \frac{1}{4} \\ 1 & \sqrt{2} & 1 \end{bmatrix}$, is 3, which means that V_1^1 is a full rank matrix.

Now, suppose that V_n^1 is a full rank matrix, and let us show it is full rank for $n + 1$. When $x \in \mathbb{R}^{n+1}$, we can partition the points in $\Delta(n + 1, 2)$ into three cases:

I) points with $x_{n+1} = 0$. These points can be generated by considering all of the points in $\Delta(n, 2)$, and adding a 0 as their last component.

II) points with $x_{n+1} = \frac{1}{2}$. The points in this class can be sub-partitioned into two groups:

   i) points with one nonzero component.

   ii) points with two nonzero components.

III) points with $x_{n+1} = 1$. Clearly, there is only one point in this class.

According to the definition of $\mathrm{svec}\left(v_1(x)v_1(x)^T\right)$, each of V_{n+1}^1's column is related to x^γ, where $\gamma \in \mathbb{N}_2^{n+1}$. Let us order the columns of V_{n+1}^1 as follows: first we put all of the columns related to x^α, where $(\alpha, 0) \in \mathbb{N}_2^{n+1}$, after that the columns related to x_{n+1}, x_{n+1}^2, $x_{n+1}x_i$, $i = 1, ..., n$. So, because each row of V_{n+1}^1 is related to a point in $\Delta(n+1, 2)$, after ordering its rows, the matrix looks like this:

$$V_{n+1} = \begin{array}{c} \\ \text{Case I} \\ \text{Case II}_{ii} \\ \text{Case II}_i \\ \text{Case III} \end{array} \begin{array}{c} \begin{array}{cccc} \underset{(\alpha,0)\in\mathbb{N}_2^{n+1}}{x^\alpha} & x_{n+1} & x_{n+1}^2 & \underset{i=1,\dots,n}{x_{n+1}x_i} \end{array} \\ \begin{pmatrix} V_n^1 & 0_{L\times 1} & 0_{L\times 1} & 0_{L\times n} \\ a_1 & \frac{\sqrt{2}}{2}\mathbb{1}_{n\times 1} & \frac{1}{4}\mathbb{1}_{n\times 1} & \frac{\sqrt{2}}{4}I_n \\ a_2 & \frac{\sqrt{2}}{2} & \frac{1}{4} & 0_{1\times n} \\ a_3 & \sqrt{2} & 1 & 0_{1\times n} \end{pmatrix} \end{array},$$

for some $a_1 \in \mathbb{R}^{n\times L}$, and $a_2, a_3 \in \mathbb{R}^{1\times L}$. Due to the induction assumption, V_n^1 is a full rank matrix, which implies that V_{n+1}^1 is a full rank matrix. Therefore, the constraints in the first iteration of the BSOS hierarchy are linearly independent. □

In Theorem 2.7, we prove that if $d = 1$, then all of the constraints in (2.6) are linearly independent. In the next theorem, we prove that for $d \geq 2$, if we rewrite H_d as $[\bar{H}_d, V_n^d]$, where

$$V_n^d = \begin{bmatrix} \mathrm{svec}\left(v_1(x^1)v_1(x^1)^T\right) \\ \vdots \\ \mathrm{svec}\left(v_1(x^L)v_1(x^L)^T\right) \end{bmatrix} \in \mathbb{R}^{L\times L},$$

then $\mathrm{rank}(H_d) = \mathrm{rank}(\bar{H}_d)$.

Theorem 2.8 *Suppose that f is quadratic, $d \geq 2$, and $\Theta \subseteq \Delta(n, 2d)$. The equality constraints in (2.6) corresponding to the points in Θ applied to the general problem (2.1) with sign constraints over all of the variables, are linearly independent if and only if rows in $\bar{H}_d$ corresponding to the points in Θ are linearly independent.*

Proof. The 'if' part is trivial.

To prove the 'only if' part, without loss of generality we assume that x^p, $p = 1, ..., t$ generate linearly independent constraints, which means that the first t rows of H_d are linearly independent. Since the objective function f is quadratic, b_d is a linear combination of the columns of V_n^d. Because of the sign constraints over all variables, each column of V_n^d is also a column in $\bar{H}_d$, for $d \geq 2$. This means that V_n^d is a submatrix of $\bar{H}_d$, which implies that the first t rows in $\bar{H}_d$ are linearly independent.

□

After elimination of the equality constraints in pooling problem (2.11), we rewrite the model with sign constraints over all of the remaining variables. So, when using Theorem 2.8 to find the linearly independent constraints, we only need to check $\bar{H}_d$.

Theorem 2.9 *Fix $d \geq 2$. Consider $\hat{H}_d$, which is a matrix with all columns of $\bar{H}_d$ related to (α, β) with $\beta = 0$. Then Range$(\bar{H}_d)$ = Range$(\hat{H}_d)$.*

Proof. Since we consider $\beta = 0$, we can write $\hat{H}_d$ as follows:

$$\hat{H}_d = \begin{bmatrix} (g(x^1)^\alpha)_{\alpha \in \mathbb{N}_d^m} \\ \vdots \\ \left(g(x^L)^\alpha\right)_{\alpha \in \mathbb{N}_d^m} \end{bmatrix},$$

where $L = \binom{n+2d}{2d}$, $g(x) = (g_1(x), ..., g_m(x))$, for each $\alpha \in \mathbb{N}_d^m$, $g(x^p)^\alpha = \prod_{j=1}^m g_j(x)^{\alpha_j}$, and $(g(x^p)^\alpha)_{\alpha \in \mathbb{N}_d^m} \in \mathbb{R}^{1 \times \binom{m+d}{d}}$, $p = 1, ..., L$.

Because the columns of $\hat{H}_d$ are a subset of the columns of $\bar{H}_d$, so Range$(\hat{H}_d) \subseteq$ Range$(\bar{H}_d)$. To prove the other containment, we show that all columns of $\bar{H}_d$ are linear combinations of $\hat{H}_d$'s columns. Each column of $\bar{H}_d$ is related to a function $h_{\alpha\beta}(x)$ for some $(\alpha, \beta) \in \mathbb{N}_d^{2m}$. If $\beta = 0$ for a column of $\bar{H}_d$, then it is a column of $\hat{H}_d$. Now consider a column with $\beta \neq 0$. Therefore, $h_{\alpha\beta}(x)$ related to this column is equal to

$$\prod_{j=1}^m g_j(x)^{\alpha_j} \prod_{j=1}^t (1 - g_j(x))^{\beta_j} = g(x)^\alpha \left(\sum_{i=1}^w a_i g(x)^{\gamma_i} \right),$$

for some $\gamma_i \in \mathbb{N}_{|\beta|}^m$, $a_i \in \mathbb{R}$, $i = 1, ..., w$, and $w \geq 0$. Hence, $h_{\alpha\beta}(x) = \sum_{i=1}^w a_i g(x)^{\gamma_i + \alpha}$. Because $\gamma_i + \alpha \in \mathbb{N}_d^m$, $g(x)^{\gamma_i + \alpha}$ is related to a column of $\hat{H}_d$, for each $i = 1, ..., w$. This means that any column of $\bar{H}_d$ is a linear combination of the columns in $\hat{H}_d$. $\square$

By Theorem 2.9, to find the number of linearly independent constraints in (2.6), we only need to check the columns related to $h_{\alpha\beta}(x)$ with $\beta = 0$.

It is clear that the results in this chapter, except Theorem 2.7, can be modified for the LO bounds (2.4). In fact, Theorems 2.8 and 2.9 are true in each level, even $d = 1$. In Table 2.4, the results of solving the pooling problem instances in Table 2.1 are shown after removing the linearly dependent constraints using (2.4) and $\hat{q}_d^1$ in (2.22). Note that the computational times at the $d = 2$ and $d = 3$ levels are greatly reduced when compared to the times in Table 2.3. For some instances because of the large number of constraints in the last level of the hierarchy, we could not find the number of linearly independent constraints and we put "-" as in Table 2.4. Also in this table we show how much stronger the BSOS hierarchy is compared to the LO bounds (2.4) after reducing the number of variables and deleting the linear dependent constraints. As one can see, the main difference between the BSOS hierarchy and (2.4) is in the first level, in which the number of independent constraints in (2.4) is much smaller than the BSOS hierarchy. If there is a difference between two hierarchies, it is presented in Table 2.4 with "()", in which the value corresponds to the LO bounds (2.4). It can be seen that there is a pay-off between using (2.4) and the BSOS hierarchy. By using the LO bounds you may solve each level faster (4 cases) but the lower bound can be strictly weaker than the one from the BSOS hierarchy (2 cases).

2.5.3 Lower bounds using PQ-formulation

Up to now, we evaluated the BSOS hierarchy on the P-formulation. Since the McCormick relaxation (Section 2.3.1) of the PQ-formulation is stronger than that of the P-formulation [68], it is worthwhile to evaluate the BSOS hierarchy using the PQ-formulation. In Table 2.5 we present these results for the PQ-formulation. To deal with the equality constraints, we replace them by two inequalities. As one can see, the quality of the lower bounds obtained by using PQ-formulation are much better in a few of the instances, such as DeyGupte4 and RT2, than when using the P-formulation; however, the size of the problems do not allow us to go further than the first iteration for the moderate-sized instances.

2.5.4 Upper bound for the number of linearly independent constraints

According to Theorem 2.9, to find the number of linearly independent columns of H_d, for $d \geq 2$ we only need to find the rank of the linear space, say N_d, spanned by $\{g(x)^\alpha\}_{\alpha\in\mathbb{N}_d^m}$. Hence, the dimension of N_d is an upper bound on the number of linearly independent constraints. In this part we give an upper bound on the dimension of N_d, which is an upper bound on the number of linearly independent constraints in (2.6).

It is clear that N_d is a subspace of the linear space M_d spanned by $\{w(x)^\alpha\}_{\alpha\in\mathbb{N}_d^\omega}$, where $w(x)$ is a vector containing all of the monomial existing in (2.1), and ω in the size of $w(x)$. Therefore, rank(M_d) is an upper bound on rank(N_d), and hence an upper bound of the number of linearly independent constraints in each iteration of the BSOS hierarchy.

In the rest of this part, we try to find rank(M_d) for the pooling problems, and assume that the number of outgoing flows from each pool is equal to J. After elimination of equality constraints in the pooling problem (2.11), the functions defining the inequality constraints can be partitioned into three classes:

I) bilinear functions,

II) x_i, $i = 1, ..., n$,

III) some other affine functions.

	iteration	rel. time	solution	# lin. var.	# const.
Haverly1	d=1	1	-600.00	12	21(8)
	d=2	1	-417.20	199	33
	d=3	0.11	**-400.00**	1,937	98
Haverly2	d=1	1	-1,200.00	12	21(8)
	d=2	1	-601.67(-640.00)	199	33
	d=3	0.13	**-600.00**	1,937	98
Haverly3	d=1	1	-875.00	12	21(8)
	d=2	1	**-750.00**	199	33
Ben-Tal4	d=1	1	-650.00	14	28(9)
	d=2	1	-467.20	274	42
	d=3	0.08	**-450.00**	3,095	140
Ben-Tal5	d=1	1(0.11)	**-3,500.00**	55	465(44)
DeyGupte4	d=1	1	-4.00	53	66(16)
	d=2	0.03	-3.86	4,135	131
	d=3	-	-	172,250	-
Foulds2	d=1	1	-1,200.00	39	190(24)
	d=2	0.002	-1,191.30	2,224	295
	d=3	-	-	49,385	-
Foulds3	d=1	$0.94(10^{-4})$	**-8.00**	220	11,781(176)
Foulds4	d=1	$0.92(10^{-4})$	**-8.00**	220	11,781(176)
Adhya1	d=1	1	-999.32	42	78(24)
	d=2	0.06(0.5)	≈-723.95	2,583	260
	d=3	-	-	85,526	-
Adhya2	d=1	1	-798.29	54	78(24)
	d=2	0.12(0.3)	-576.83	4,293	260
	d=3	-	-	182,214	-
Adhya3	d=1	1	-882.84	67	171(38)
	d=2	0.02	≈-802.88(-806.64)	6,634	671
	d=3	-	-	348,602	-
Adhya4	d=1	1	-1,055.00	52	153(39)
	d=2	0.03	≈-1,035.54	3,979	732
	d=3	-	-	162,657	-
RT2	d=1	1	-45,420.48	68	120(23)
	d=2	0.02	-36,541.89	6,836	266
	d=3	-	-	364,480	-
sppA0	d=1	$0.99(10^{-4})$	-47,675.00	817	13,203(372)
	d=2	-	-	1,000,008	-

Table 2.4: Results for computing the bounds from (2.4) and $\hat{q}_d^1$ in (2.22) after removing of linearly dependent constraints. The values in "()" are corresponding to the LO bounds (2.4) if they are different than those from $\hat{q}_d^1$.

	iteration	rel. time	solution	# lin. var.	size of SD var.	# const.
Haverly1	d=1	1	-600.00	25	9	45
	d=2	1	-411.11	901	9	82
	d=3	-	Numerical Prob.	17,901	9	354
Haverly2	d=1	1	-1,200.00	25	9	45
	d=2	1	**-600.00**	901	9	82
Haverly3	d=1	1	-875.00	25	9	45
	d=2	1	**-750.00**	901	9	82
Ben-Tal4	d=1	1	-650.00	30	11	66
	d=2	1	-459.86	1,306	11	124
	d=3	-	-	31,031	11	-
Ben-Tal5	d=1	4.17	**-3,500.00**	127	45	1,035
DeyGupte4	d=1	1	-4.00	89	17	153
	d=2	0.35	≈-2.49	11,749	17	438
	d=3	-	-	818,445	17	-
Foulds2	d=1	1	-1,200.00	77	25	325
	d=2	-	-	8,779	25	-
Foulds3	d=1	2.26	**-8.00**	628	193	18,721
Foulds4	d=1	2.32	**-8.00**	628	193	18,721
Adhya1	d=1	1	-999.32	73	19	190
	d=2	-	-	7,885	19	-
Adhya2	d=1	1	-798.29	81	19	190
	d=2	-	-	9,721	19	-
Adhya3	d=1	2	-882.84	109	29	435
	d=2	-	-	17,659	29	-
Adhya4	d=1	2	-1,055.00	96	27	378
	d=2	-	-	13,680	27	-
RT2	d=1	1	-18,155.84	96	23	276
	d=2	-	-	13,680	23	-
sppA0	d=1	5.88	-47,675.00	1,165	234	27,495
	d=2	-	-	1,326,340	234	-

Table 2.5: Results for computing the bounds $\hat{q}_d^1$ in (2.22) after removing linearly dependent constraints on the PQ-formulation.

The bilinear functions are those related to constraints (2.11h) and (2.11i), or those related to the constraints (2.11f) after elimination of equality constraints. Hence, the only bilinear terms in the reformulated problem are $p_{lk}y_{lj}$, for each pool l and specification k, where there is an outgoing flow from pool l to output j. Therefore,

$$\left\langle \left\{ \left(1, \{y_{ij}\}_{\substack{i\in\mathcal{I}\\ j\in\mathcal{J}}}, \{y_{lj}\}_{\substack{l\in\mathcal{L}\\ j\in\mathcal{J}}}, \{y_{il}\}_{(i,l)\in\bar{\mathcal{I}}}, \{p_{lk}\}_{(l,k)\in\bar{\mathcal{L}}}, \{p_{lk}y_{lj}\}_{(l,k,j)\in\bar{\mathcal{J}}}\right)^{\alpha} \right\}_{\alpha\in\mathbb{N}_d^{\omega}} \right\rangle = M_d,$$

where $\bar{\mathcal{I}}$, $\bar{\mathcal{L}}$ and $\bar{\mathcal{J}}$ are respectively including (i,l), (l,k) and (l,k,j) that y_{il}, p_{lk} and $p_{lk}y_{lj}$ appear in (2.11) after elimination of the equality constraints, and

$$\omega = 1 + I\times L + L\times J + |\bar{\mathcal{I}}| + |\bar{\mathcal{L}}| + |\bar{\mathcal{J}}|.$$

Clearly the number of variables in the pooling problem (2.11) after elimination of equality constraints is $I\times L + L\times J + |\bar{\mathcal{I}}| + |\bar{\mathcal{L}}|$. For $d = 1$, we prove in Theorem 2.7 that all of the constraints in (2.6) are linearly independent, with the number of $\binom{n+2}{2}$. For $d \geq 2$, we are seeking for the monomials up to degree $2d$ that appear in M_d. If $d = 2$, the number of monomials with degree at most 2 is $\binom{n+2}{2}$. The number of monomials with degree 3 that appear in M_d is at most

$$K\times L\times\left[\binom{n+1}{2} - \binom{n-J+1}{2}\right],$$

because for each $k\in\mathcal{K}$ and $l\in\mathcal{L}$, in this case the only way of having a monomial with degree 3 is by multiplying a monomial of degree 2 with a variable, which makes $\binom{n+1}{2} - \binom{n-J+1}{2}$ monomials of degree 3. And finally, the number of monomials of degree 4 that appear in M_d is $\left[\binom{K\times L\times J}{2} + K\times L\times J\right]$, because the only ways to make such monomials are by taking the square of a monomial with degree 2, or multiplying two degree 2 monomials. Therefore, the number of linearly independent constraints for $d = 2$ is at most

$$\binom{n+2}{2} + K\times L\times\left[\binom{n+1}{2} - \binom{n-J+1}{2}\right] + \binom{K\times L\times J}{2} + K\times L\times J. \tag{2.23}$$

With the same line of reasoning as above, the number of monomials with degree at most 6 for $d = 3$ is less than or equal to

$$\begin{aligned}
&\underbrace{\binom{n+3}{3}}_{\substack{\text{monomials up}\\ \text{to degree 3}}} + \underbrace{K\times L\times\left[\binom{n+2}{3} - \binom{n-J+2}{3}\right]}_{\substack{\text{monomials of}\\ \text{degree 4}}}\\
&\underbrace{+K\times L\times\left[\binom{n+2}{3} - \binom{n-J+2}{3} - J\times\binom{n-J+1}{2}\right]}_{\substack{\text{monomials of}\\ \text{degree 5}}}\\
&+\underbrace{\left[\binom{K\times L\times J}{3} + K\times L\times J + 2\binom{K\times L\times J}{2}\right]}_{\substack{\text{monomials of}\\ \text{degree 6}}}.
\end{aligned} \tag{2.24}$$

Example 2.2 *Consider the example (2.21). The only bilinear terms in (2.21) are y_1y_2 and y_1y_3. So,*

$$M_d = \left\langle \{(1, y_1, y_2, y_3, y_4, y_5, y_1y_2, y_1y_3)^{\alpha}\}_{\alpha\in\mathbb{N}_d^8} \right\rangle$$

Therefore, the number of linearly independent constraints is at most

$$\binom{7}{2} + \binom{6}{2} - \binom{4}{2} + \binom{2}{2} + 2 = 33,$$

if $d = 2$, and

$$\binom{8}{3} + 2\times\binom{7}{3} - 2\times\binom{5}{3} - 2\times\binom{4}{2} + 2\times\binom{2}{2} + 2 = 98,$$

if $d = 3$. □

2.6 Improving lower bounds by adding valid inequalities

Adding redundant constraints to the original problem (2.1) increases the number of linear variables in (2.5); this introduces some flexibility in each level of the hierarchy because of the new linear variables and may provide a stronger lower bound. As it was mentioned in Section 2.3.1, for each bilinear term in the P- or PQ-formulations there are four valid inequalities given by (2.14). So, in Table 2.6 we present the result of adding these valid inequalities to the P-formulation and using $\hat{q}_d^1$ in (2.22) to solve the problem. In each level of the hierarchy in this table, we use the upper bounds for the number of constraints proposed in Section 2.5.4. As Table 2.6 shows, this improvement helps to obtain the optimal values of Haverly1, Harverly2, Ben-Tal4, and DeyGupte4, and to get a good approximation of the optimal value of Foulds2 in the second level of the hierarchy. Also, for Adhya1,2,4 we obtained better lower bounds than the PQ-linear relaxation values in Table 2.1.

2.7 Conclusion

In this chapter we analyzed and evaluated the bounded degree sum-of-squares (BSOS) hierarchy of Lasserre, Toh and Yang [88] for a class of bilinear optimization problems, namely pooling problems. We showed that this approach is successful in obtaining the global optimal values for smaller instances, but scalability remains a problem for larger instances. In particular, the number of nonnegative variables and linear constraints grows quickly with the level of the hierarchy. We have showed that it is

	iteration	rel. time	solution	# lin. var.	size of SD var.	# const.
Haverly1	d=1	1	-600.00	18	6	21
	d=2	1	**-400.00**	460	6	33
Haverly2	d=1	1	-1,100.00	18	6	21
	d=2	1	**-600.00**	460	6	33
Haverly3	d=1	1	-850.00	18	6	21
	d=2	1	**-750.00**	460	6	33
Ben-Tal4	d=1	1	-650.00	20	7	28
	d=2	1	**-450.00**	571	7	42
Ben-Tal5	d=1	1	**-3,500.00**	145	30	465
DeyGupte4	d=1	1	-4.00	101	11	66
	d=2	0.32	**-1.02**	15,155	11	170
Foulds2	d=1	1	-1,200.00	63	19	190
	d=2	0.02	-1,101.83	5,860	19	358
	d=3	-	-	289,695	19	2,850
Foulds3	d=1	1.60	**-8.00**	604	153	11,781
Foulds4	d=1	1.53	**-8.00**	604	153	11,781
Adhya1	d=1	1	-960.37	138	12	78
	d=2	1.34	-640.19	28,360	12	270
	d=3	-	-	3,056,470	12	2,860
Adhya2	d=1	1	-777.63	198	12	78
	d=2	1.13	-569.55	58,510	12	270
	d=3	-	-	9,036,390	12	4,108
Adhya3	d=1	1	-879.02	283	18	171
	d=2	1.06	-664.39	119,710	18	691
	d=2	-	-	26,402,485	18	13,452
Adhya4	d=1	1	-1,032.50	172	17	153
	d=2	1.58	-948.88	44,119	17	1,038
	d=3	-	-	5,921,616	17	7,089
RT2	d=1	1	-36,542.22	212	15	120
	d=2	0.091	≈-32,739.03	67,099	15	354
	d=3	-	-	11,093,536	15	6,440
sppA0	d=1	1.39	≈-46,636.57	4,705	162	13,203
	d=2	-	-	37,414,021	162	94,812

Table 2.6: Results for computing the lower bounds $\hat{q}_d^1$ for the P-formulation after adding valid inequalities and considering (2.23) and (2.24) as the upper bound on the number of linearly independent constraints.

possible to eliminate some variables and redundant linear constraints in the hierarchy in a systematic way, and this goes some way in improving scalability. More ideas are needed, though, if this approach is to become competitive for medium to larger scale pooling problems.

In the next chapter, we will investigate how one may solve the BSOS hierarchy more efficiently by exploiting sparsity in the data, in particular so-called “chordal sparsity”.

CHAPTER 3

Solving sparse polynomial optimization problems with chordal structure using the sparse-BSOS hierarchy

3.1 Introduction

A PO problem is a mathematical optimization problem in which all constraints and the objective function are multi-variate polynomials. PO problems include non-convex QCQO problems, which were proved to be intractable by Pardalos and Vavasis [105].

Many approaches are available for constructing lower bounds for the optimal value of a PO problem (denoted by f^*). Kim, Kojima and Waki [81] proposed a relaxation of a PO problem using a generalized Lagrangian dual. Lasserre [86] introduced an LO hierarchy that constructs a sequence of lower bounds for f^*. Using the Krivine *positivstellensatz* (Theorem 2.2), Lasserre showed that under some assumptions the sequence converges to f^*. In the hope of getting a tighter lower bound, Lasserre, Toh, and Yang [88] extended the LO hierarchy to an SDO one, called the BSOS hierarchy. The advantage of the BSOS hierarchy is that it contains one semi-definite matrix variable, which has a fixed size that is independent of the level of the hierarchy. A major drawback of the BSOS hierarchy lies in the fact that the number of linear variables grows quickly when the level of the hierarchy increases. Also, for a large problem, the size of the semi-definite matrix variable gets large, which makes the hierarchy inefficient. In an effort to resolve these issues, Weisser, Lasserre and Toh [120] introduced a modification of the BSOS hierarchy, called the sparse-BSOS hierarchy, for PO problems with a particular structural sparsity, which satisfies the running intersection property (RIP).

The RIP is a well-known concept in graph theory. In the literature of positive semi-definite (PSD) matrices and polynomial optimization, exploiting a sparsity that satisfies the RIP is done by studying the corresponding chordal graphs, see [61] for PSD matrices and [119] for polynomial optimization. The results in [120] can be seen as a combination of the results in the papers [88] and [119].

PO problems have many real-life applications. Some of these applications were studied in the recent paper by Ahmadi and Majumdar [3]. In this chapter, we analyze the behavior of the sparse-BSOS hierarchy on a class of bilinear programming problems, called pooling problems, and a class of discrete-time optimal control problems.

Solving the pooling problem is attracting considerable interest due to their applications in many real-life optimization problems, like oil refinery planning, chemical process, and water-waste network design. There are many formulations for the pooling problem. Haverly [70] proposed a formulation, called the P-formulation. It was shown by Alfaki and Haugland [7] that the P-formulation problem is NP-hard. One way of finding a lower bound for the P-formulation problem is by using the McCormick relaxation of each bilinear term, which can be reformulated as a Mixed Integer Linear Programing problem. In order to tighten this relaxation, Tawarmalani and Sahinidis [115] proposed the *PQ-formulation*. Dey and Gupte [51] proved that even for the PQ-formulation, using the McCormick relaxations of the bilinear terms might yield a lower bound that is far from the optimal value of the problem. There are other formulations with different characteristics for the pooling problem, like Q-, TP- formulations. Detailed discussion can be found in the surveys by Misener and Floudas [96] and Gupte et al. [68], and the Ph.D. thesis by Alfaki [6].

Recently, SDO hierarchies have been used to find lower bounds for pooling problems. Frimannslund et al. [60] applied the hierarchy proposed by Lasserre [85] to pooling problems. As they pointed out, the fast increase in the sizes of the semi-definite matrix variables in the problem, which is related to the level of the hierarchy, prevents the hierarchy from being applicable for pooling problems. In Chapter 2, we evaluated the BSOS hierarchy on pooling problems. We found that the BSOS hierarchy is successful in acquiring the optimal values of small-sized instances, but because of the number of variables, the hierarchy does not work well on moderate and large-sized instances. In this chapter, we evaluate the sparse-BSOS hierarchy on the P-formulation of the pooling problem and compare the results with BSOS.

Another class of problem that we consider in this chapter is a class of optimal control problems. A continuous-time optimal control problem finds a control function for a dynamical system such that a certain objective function is optimized. A discrete-time optimal control (DTOC) is a method of solving a continuous one by discritizing the time slot that is considered in it. There are different approaches to solve a DTOC problem using a nonlinear optimization problem. In [59], Friesz provides necessary conditions to make the KKT solutions optimal for the problem. Also, the authors in [45, 52, 119] test their proposed method on a DTOC problem. We refer the reader to the Ph.D. thesis by Nielsen [103] for a more detailed discussion. In this chapter, we show how sparse-BSOS hierarchy works on a DTOC problem.

The BSOS and sparse-BSOS hierarchies are applicable to PO problems that do not contain any equality constraints. However, all pooling problem formulations and DTOC problems contain many equality constraints. The standard way of dealing with equality constraints is elimination, or replacing them with two inequalities. A way of eliminating equality constraints in the P-formulation was proposed in Section 2.3.2; however, the elimination may destroy the sparsity pattern. On the other hand, replacing any equality constraints with two inequalities keeps the sparsity pattern but increases the number of constraints in the problem, which is not desirable in the BSOS and sparse-BSOS hierarchies because it makes each level harder to solve. In this chapter, we show how the hierarchies can be modified to deal with equality constraints directly so that the convergence results remain valid.

The remainder of the paper is organized as follows. In Section 3.2, we describe continuous- and discrete-time optimal control problems. Section 3.3 describes the sparse-BSOS hierarchy proposed in [120]. Section 3.4 demonstrates the link between graph theory, PSD matrices, and PO problems. In particular, in Section 3.4.1 we mention some well-known results in graph theory. Then in Section 3.4.2 we provide the links between chordal graphs and PSD matrices, and in Section 3.4.3 we construct a graph corresponding to a PO problem and exploit a sparsity that satisfies the RIP. In Section 3.5, we show how to modify the BSOS and sparse-BSOS hierarchies to deal with equality constraints directly. A numerical evaluation of the results is provided in Section 3.6.

3.2 Discrete-time optimal control

In this section, we briefly describe continuous- and discrete-time optimal control (DTOC) problems. We borrow the notation from [109]. A continuous-time optimal control is the optimization problem

$$\begin{aligned} \min_{\substack{x(.):\mathbb{R}\to\mathbb{R}^n \\ u(.):\mathbb{R}\to\mathbb{R}^m}} \quad & \int_0^T F[x(t), u(t), t]\, dt + S[x(T), T] \\ \text{s.t.} \quad & \dot{x}(t) = f[x(t), u(t), t], \ \ \forall t \in [0, T] \\ & x(t) \in \mathcal{X}, \ u(t) \in \mathcal{U}, \ \ \forall t \in [0, T] \\ & x(0) = x_0, x(T) = x_T, \end{aligned} \tag{3.1}$$

where $F[.,.,.] : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}$ is a function with continuous partial derivatives with respect to $x(t)$, $S[.,.] : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ is a function with continuous partial derivatives with respect to $x(T)$, $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^m$ is a function, $\mathcal{X} \subseteq \mathbb{R}^m$ and $\mathcal{U} \subseteq \mathbb{R}^n$ are given sets, and x_0, x_T are given vectors.

One of the methods for solving (3.1) is discretizing the problem and consider the discrete-time optimal control

$$
\begin{aligned}
\min_{\substack{x_k\in\mathbb{R}^n, k=1,\ldots,N-1\\ u_k\in\mathbb{R}^m, k=0,\ldots,N-1}} \quad & \frac{1}{N}\sum_{k=0}^{N-1} F\left[x_k, u_k, \frac{kT}{N}\right] + S[x(T),T]\\
\text{s.t.} \quad & x_{k+1} = x_k + \frac{1}{N} f\left[x_k, u_k, \frac{kT}{N}\right], k = 0, \ldots, N-1\\
& x_k \in \mathcal{X}, k = 1, \ldots, N-1,\\
& u_k \in \mathcal{U}, k = 0, \ldots, N-1,
\end{aligned}
\tag{3.2}
$$

where x_0, x_N are given. It is clear that when F and f are polynomials, and the sets $\mathcal{X}$ and $\mathcal{U}$ are semi-algebraic, then (3.2) is a polynomial optimization problem. The size of (3.2) is related to the number of intervals (N), which can be large. In Section 3.6.2, we show how one can apply the sparse-BSOS hierarchy to solve (3.2).

3.3 Sparsity pattern in a polynomial optimization problem

In this section, we briefly describe the sparse-BSOS hierarchy in PO problems introduced, by Weisser, Lasserre and Toh. [120]. In what follows, for an integer $m \geq 0$,

$$
[m] := \begin{cases} \{1, \ldots, m\} & \text{if } m > 0,\\ \emptyset & \text{if } m = 0. \end{cases}
$$

Also, we assume that $x \in \mathbb{R}^n$, and for a given $\mathcal{D} \subseteq [n]$, we denote by $\Sigma[x;\mathcal{D}]_\kappa$, the cone of SOS polynomials of degree at most 2κ, and by $\mathbb{R}[x;\mathcal{D}]$ the ring of all polynomials in the variables $\{x_i : \ i \in \mathcal{D}\}$.

We recall that a general PO problem is the mathematical problem:

$$
\begin{aligned}
f^* = \min_x \quad & f(x)\\
\text{s.t.} \quad & g_j(x) \geq 0, \quad j = 1, \ldots, m,
\end{aligned}
\tag{3.3}
$$

where $x \in \mathbb{R}^n$, $n, m \in \mathbb{N}$ and all $f(x)$ and $g_j(x)$, $j = 1, \ldots, m$ are n-variate polynomials. For (3.3) the *running intersection property* is defined as follows.

Definition 3.1 *Problem (3.3) satisfies the running intersection property if there exists $q \in \mathbb{N}$, $\mathcal{D}_\ell \subseteq [n]$ and $\mathcal{C}_\ell \subseteq [m]$ for all $\ell \in [q]$ such that*

- *$f = \sum_{\ell=1}^q f^\ell$, for some $f^\ell \in \mathbb{R}[x;\mathcal{D}_\ell]$, for all $\ell \in [q]$,*
- *$g_j \in \mathbb{R}[x;\mathcal{D}_\ell]$, for all $j \in \mathcal{C}_\ell$, and $\ell \in [q]$,*

- $\bigcup_{\ell=1}^{q} \mathcal{D}_\ell = [n]$, *and* $\bigcup_{\ell=1}^{q} \mathcal{C}_\ell = [m]$,
- *for all* $\ell \in [q-1]$, *there is an* $s \leq \ell$ *such that* $\left(\mathcal{D}_{\ell+1} \cap \bigcup_{r=1}^{\ell} \mathcal{D}_r\right) \subseteq \mathcal{D}_s$. $\square$

Assume that the *running intersection property* holds for (3.3). Let

$$\hat{\mathbb{N}}_d^\ell := \left\{ (\alpha, \beta) \in \mathbb{N}^{2m} : \quad \alpha_j = \beta_j = 0 \text{ if } j \notin \mathcal{C}_\ell, \ \sum_{j \in [m]} \alpha_j + \beta_j \leq d \right\},$$

and

$$h_{\alpha\beta}^\ell := \prod_{j \in [m]} g_j^{\alpha_j} (1 - g_j)^{\beta_j} \in \mathbb{R}[x, \mathcal{D}_\ell], \ (\alpha, \beta) \in \hat{\mathbb{N}}_d^\ell.$$

Then, we have the following result from [120].

Theorem 3.1 *[120, Theorem 2] Consider the general PO problem (3.3). Suppose that it satisfies the running intersection property, and* $g_j(x) \leq 1$ *for any feasible solution* x, $j \in [m]$. *Also, assume that for all* $\ell \in [q]$, *the ring of* $\mathbb{R}[x; \mathcal{D}_\ell]$ *is generated by* $\{1, (g_j)_{j \in \mathcal{C}_\ell}\}$, *and there exists* $M_\ell > 0$ *and* $j \in \mathcal{C}_\ell$ *such that* $g_j = 1 - \frac{1}{M_\ell}\left(\sum_{i \in \mathcal{D}_\ell} x_i^2\right)$. *Then, for a fixed* $\kappa \in \mathbb{N}$, $\{\tilde{q}_d^\kappa\}$ *is a non-decreasing sequence and* $\tilde{q}_d^\kappa \to f^*$ *as* $d \to +\infty$, *where*

$$\tilde{q}_d^\kappa := \sup \left\{ t : \begin{array}{ll} f^\ell - \sum_{(\alpha,\beta) \in \hat{\mathbb{N}}_d^\ell} \lambda_{\alpha\beta}^\ell h_{\alpha\beta}^\ell \in \Sigma[x; \mathcal{D}_\ell]_\kappa, & \ell \in [q] \\ f - t = \sum_{\ell \in [q]} f^\ell, \ \ \lambda^\ell \geq 0, \ t \in \mathbb{R}, & f^\ell \in \mathbb{R}[x; \mathcal{D}_\ell], \ \ell \in [q] \end{array} \right\}. \quad (3.4)$$

$\square$

Theorem 3.1 introduces a non-decreasing sequence that converges to the optimal value of (3.3) under some assumptions. Instead of (3.4), we consider the following equivalent problem where the f^ℓ, $\ell = 1, ..., q$, have been eliminated,

$$\tilde{q}_d^\kappa = \sup \left\{ t : \begin{array}{c} f - t = \sum_{\ell \in [q]} \sum_{(\alpha,\beta) \in \hat{\mathbb{N}}_d^\ell} \lambda_{\alpha\beta}^\ell h_{\alpha\beta}^\ell + \sum_{\ell \in [q]} \sigma_\ell \\ \sigma_\ell \in \Sigma[x; \mathcal{D}_\ell]_\kappa, \ \ \lambda^\ell \geq 0, \ t \in \mathbb{R}, \ \ell \in [q] \end{array} \right\}. \quad (3.5)$$

The number of scalar variables in each level of (3.5) is smaller than the one in the same level of the BSOS hierarchy (2.5). In the next proposition we show that all constraints in (3.5) are linearly independent, when the degree of the SOS polynomial $\sum_{\ell \in [q]} \sigma_\ell$ equals the degree of the whole equality constraint. To prove this proposition, we need the following remark.

Remark 3.1 *Let* $x \in \mathbb{R}^n$ *and* $p(x) = \sum_{\alpha \in \bar{\mathbb{N}}_{2\omega}^{[n]}} p_\alpha x^\alpha$, *where*

$$\bar{\mathbb{N}}_\kappa^{\mathcal{D}} := \left\{ \alpha \in \mathbb{N}^n : \ \alpha_i = 0 \ \textit{if } i \notin \mathcal{D}, \ \sum_{i \in [n]} \alpha_i \leq \kappa \right\},$$

for a set $\mathcal{D} \subseteq [n]$. If $M \in \mathbb{R}^{\binom{n+\omega}{\omega}\times\binom{n+\omega}{\omega}}$ is a symmetric matrix variable whose rows (columns) are corresponding to the members of $\bar{\mathbb{N}}^{[n]}_{\omega}$, then linear constraints

$$p_\alpha = \sum_{\substack{\beta,\gamma\in\bar{\mathbb{N}}^{[n]}_{\omega}\\ \beta+\gamma=\alpha}} M_{\beta\gamma}, \quad \forall\alpha \in \bar{\mathbb{N}}^{[n]}_{2\omega}, \tag{3.6}$$

are linearly independent. This is because all the constraints in (3.6) involve different variables, i.e., no variable appears in two constraints in (3.6). To see this, let $\beta, \gamma \in \bar{\mathbb{N}}^{[n]}_{\omega}$ be fixed. Due to the construction of the constraints in (3.6), the variable $M_{\beta\gamma}$ appears only in the constraint corresponding to $\alpha = \beta + \gamma$, and no other constraints. □

Proposition 3.1 *Consider problem (3.3). Let d be such that*

$$2\kappa = \max\{d \max_{j=1,\dots,m} (deg(g_j)), deg(f), 2\kappa\}.$$

Then, all equality constraints in (3.5) are linearly independent, if the polynomial equality is modeled by equating the monomials coefficients.

Proof. For each $\ell \in [q]$, set $v^\ell = \left(x^\beta\right)_{\beta\in\bar{\mathbb{N}}^{\mathcal{D}_\ell}_{\kappa}}$. Also, let $\sigma_\ell = {v^\ell}^T W^\ell v^\ell$, for each $\ell \in [q]$, where $W^\ell \in \mathbb{R}^{\binom{n_\ell+\kappa}{\kappa}\times\binom{n_\ell+\kappa}{\kappa}}$ is a PSD matrix variable, and $n_\ell = |\mathcal{D}_\ell|$. So, Remark 3.1 implies that the equality constraints in (3.5) are linearly independent, if the polynomial equality is modeled by equating the monomials coefficients. □

According to the proof of Proposition 3.1, if

$$2\kappa \neq \max\{d \max_{j=1,\dots,m} (\deg(g_j)), \deg(f), 2\kappa\},$$

still the constraints corresponding to the monomials up to degree 2κ are linearly independent. If $\deg(f) > \max\{d \max_{j=1,\dots,m} (\deg(g_j)), 2\kappa\}$, then

$$\deg(f) = \max\{d \max_{j=1,\dots,m} (\deg(g_j)), \deg(f), 2\kappa\},$$

and clearly the dth iteration of the hierarchy is infeasible, because there is no monomial with degree $\deg(f)$ in $\sum_{\ell\in[q]} \sum_{(\alpha,\beta)\in\hat{\mathbb{N}}^{\ell}_{d}} \lambda^\ell_{\alpha\beta} h^\ell_{\alpha\beta} + \sum_{\ell\in[q]} \sigma_\ell$.

The main assumption in Theorem 3.1 is the existence of a splitting that satisfies the *running intersection property*. So, the question is how to exploit such a sparsity for a PO problem. In the next section we answer this question.

3.4 Polynomial optimization and chordal graphs

In this section, we study the relation between graph theory, PSD matrices, and PO problems. Specifically, we mention some results on chordal graphs and their relations to PSD matrices, and use them to exploit sparsity for a PO problem that satisfies the *running intersection property*.

3.4.1 Chordal graph and maximal cliques

In this subsection we recall some well-known results on chordal graphs and maximal cliques. The notation we are using in this section is the same as in [41].

Definition 3.2 *Consider an undirected graph $G = (V, E)$, where V and E denote the sets of vertices and edges, respectively. A chord of a cycle is any edge joining two nonconsecutive vertices of the cycle. A graph G is called chordal, if every cycle of length greater than 3 has a chord.* □

Definition 3.3 *Let $G = (V, E)$ be any graph. A clique of G is any subset of V for which the induced graph is complete in G. A maximal clique is a clique that is not properly contained in another clique. We denote by $\mathcal{K}_G$ the set of all maximal cliques of G.* □

Example 3.1 *Figure 3.1 provides a chordal and a nonchordal graph. The graph in Figure 3.1(a) is not chordal, because the cycle* $(1, 2, 4, 3)$ *has no chord. For this graph the maximal cliques are* $\{1, 2\}$, $\{2, 4\}$, $\{3, 4\}$, *and* $\{1, 3\}$. *The graph in Figure 3.1(b) is chordal with the maximal cliques* $\{1, 2, 3\}$ *and* $\{2, 3, 4\}$.

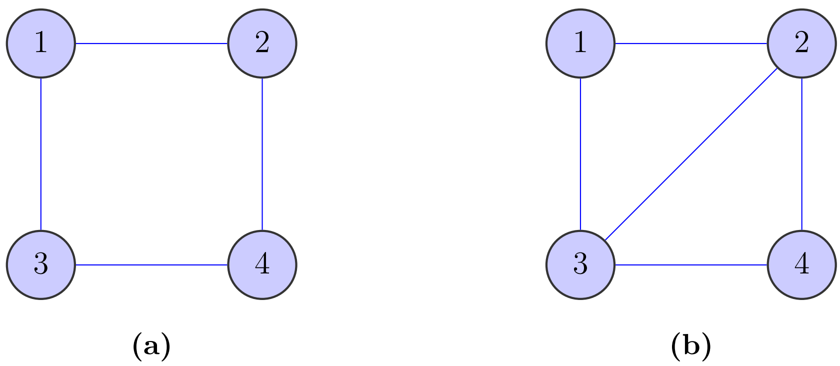


Figure 3.1: Simple examples of chordal and non-chordal graphs: The graph in (a) is not chordal with four maximal cliques. The graph (b), however, is chordal with two maximal cliques.

□

Let adj(v) denote the set of vertices adjacent to a vertex v. A vertex ordering ϕ of graph $G = (V, E)$ with n vertices is a bijection $\phi : V \to [n]$, and it can be denoted by indexing the vertex set, such that $V = \{v_1, ..., v_n\}$, and $\phi(v_i) = i$, for $i \in [n]$. Let $v_1, ..., v_n$ be a vertex ordering of G and set

$$\mathcal{L}_i := \{v_i, ..., v_n\}, \quad i \in [n].$$

Definition 3.4 *A vertex ordering ϕ is a perfect elimination ordering if for any $i \in [n]$ the subgraph of G induced by $\mathcal{L}_i \cap adj(v_i)$ is complete.* □

In words, in a perfect elimination ordering, the induced subgraph on v_i and its neighboring vertices that come after it in the ordering, forms a clique for $i \in [n]$.

Example ***3.1(continued)*** *The labeling of the graph in Figure 3.1(b) is a perfect elimination ordering. However, the graph in Figure 3.1(a) cannot have a perfect elimination ordering, since it is a cycle graph of length 4, and in any labeling $\mathcal{L}_1 \cap adj(v_1)$ is not complete.* $\square$

As we already mentioned, the graph in Figure 3.1(b) is chordal in contrary to the one in Figure 3.1(a). The following theorem states the link between a chordal graph and a perfect elimination ordering.

Theorem 3.2 *(see, e.g., [41, Theorem 2.2]) A graph G is chordal if and only if G has a perfect elimination ordering.* $\square$

Definition 3.5 *Consider a graph G with the set of maximal cliques $\mathcal{K}_G$. A tree with the vertex set $\mathcal{K}_G$, which is called a clique tree of G, satisfies the clique-intersection property, if, for every pair of distinct cliques $\hat{K}, K' \in \mathcal{K}_G$, the set $\hat{K} \cap K'$ is contained in every clique on the path connecting $\hat{K}$ and K' in the tree.* $\square$

Here, we provide an example to illustrate Definition 3.5.

Example 3.2 *Let G be the chordal graph in Figure 1.3 on page 6. For this graph, a tree that satisfies the clique-intersection property is provided in Figure 3.2.*

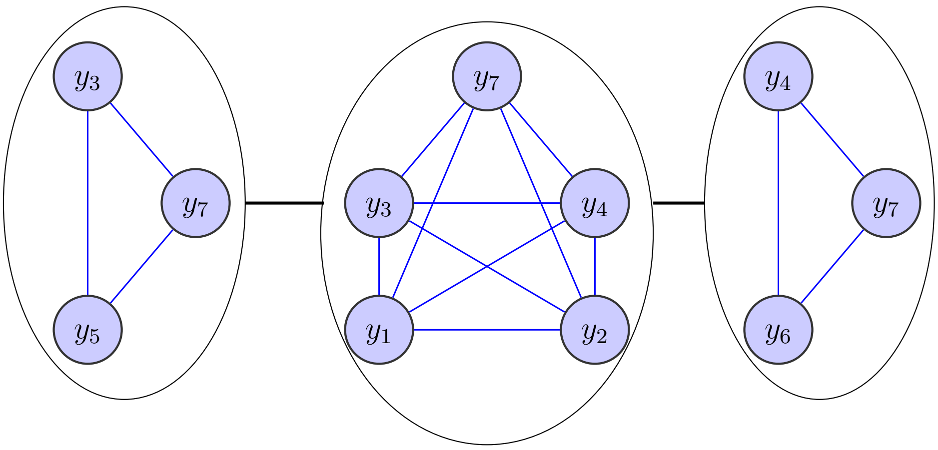


Figure 3.2: A clique tree of the graph G in Figure 1.3 that satisfies the clique-intersection property.

$\square$

In Example 3.2, we considered a chordal graph and provided a clique tree that satisfies the clique-intersection property. The following theorem shows that the existence of the clique tree in Example 3.2 was not by chance.

Theorem 3.3 *(see, e.g., [41, Theorem 3.1]) A connected graph G is chordal if and only if it has a clique tree that satisfies the clique-intersection property.* $\square$

Next definition provides the *running intersection property* in graph theory.

Definition 3.6 *For a graph G with q maximal cliques, a labeling $\mathcal{K}_G = \{K_1, ..., K_q\}$ has the running intersection property if for each clique K_j, $j = 2, ..., q$, there exists a clique K_i, $i = 1, ..., j-1$, such that*

$$K_j \cap (K_1 \cup K_2 \cup ... \cup K_{j-1}) \subset K_i.$$

$\square$

In the following theorem, the link between the running intersection and clique-intersection properties is established.

Theorem 3.4 *(see, e.g., [41, Corollary 1]) For a connected graph G, a clique tree satisfies clique-intersection property if and only if there exists a labeling of $\mathcal{K}_G$ that satisfies the running intersection property.* $\square$

Now, it is time to connect two concepts of chordal graph and the running intersection property by using the following theorem.

Corollary 3.1 *For a graph G, assume that $|\mathcal{K}_G| = q$. Then, G is chordal if and only if there is a labeling $\mathcal{K}_G = \{\mathcal{K}_1, ..., \mathcal{K}_q\}$ such that*

$$\forall \ell \in [q-1],\ \exists s \le \ell : \left(\mathcal{K}_{\ell+1} \cap \bigcup_{r=1}^{\ell} \mathcal{K}_r\right) \subseteq \mathcal{K}_s.$$

Proof. By invoking Theorems 3.3 and 3.4, one can deduce the theorem. $\square$

Corollary 3.1 asserts the link between a chordal graph and the running intersection property in its maximal cliques. Later in Section 3.4.3, we will show that the existence of a labeling on the maximal cliques that satisfies the *running intersection property* is enough for finding the sparsity in a PO problem. Now, the question is how we can find the maximal cliques of a chordal graph. Next theorem answers this question.

Theorem 3.5 *(see, e.g., [41, Lemma 6]) For a graph G with n vertices, let $\{v_1, ..., v_n\}$ be a perfect elimination ordering. Then, the set of maximal cliques of G is given by*

$$\mathcal{K}_i := \{v_i\} \bigcup (\mathcal{L}_i \cap adj(v_i))$$

where there is no vertex v_j, $j < i$, such that $\mathcal{K}_i \subset \mathcal{K}_j$. $\square$

Example ***3.1(continued)*** *To see how Theorem 3.5 works, we apply it to the chordal graph in Figure 3.1(b). As we already mentioned,* $\{1, 2, 3, 4\}$ *is a perfect elimination ordering. It is clear that*

$$\mathcal{K}_1 = \{1\} \cup \{2, 3\},\ \mathcal{K}_2 = \{2\} \cup \{3, 4\},\ \mathcal{K}_3 = \{3\} \cup \{4\}.$$

Since, $\mathcal{K}_3 \subset \mathcal{K}_2$*, the set of maximal cliques is* $\mathcal{K}_G = \{\mathcal{K}_1, \mathcal{K}_2\}$*.* □

For any vertex ordering ϕ of graph $G = (V, E)$, let us add extra edges to G in order to make all $\mathcal{L}_i \cap \mathrm{adj}(v_i)$ complete, $i \in [n]$, and denote by E^*_ϕ the union of E with the extra edges. Then clearly $G^* = (V, E^*_\phi)$ is chordal, because ϕ is a *perfect elimination ordering* for G^*, and one may use Theorem 3.5 to find the maximal cliques of G^*. The graph G^* is called a chordal extension (or triangulation) of G.

3.4.2 Chordal graphs and positive semi-definite matrices

In this section, we briefly mention the known results on the connection of PSD matrices and chordal graphs. We start with the definition of the Laplacian matrix of a graph G, which is known to be PSD.

Definition 3.7 *Laplacian matrix of a graph G with vertices* $\{v_1, ..., v_n\}$ *is defined as follows:*

$$L_{ij} = \begin{cases} deg(v_i) & \text{if } i = j \\ -1 & \text{if } i \neq j, \text{ and } v_j \text{ is adjacent to } v_i \\ 0 & \text{otherwise,} \end{cases}$$

where $deg(v_i)$ *is the degree of the vertex* v_i*,* $i = 1, ..., n$*.* □

It is well-known that the Laplacian matrix of a graph $G = (V, E)$ is PSD. This is because L is a symmetric matrix with positive diagonal entries and

$$L_{ii} = \sum_{\substack{j=1 \\ i \neq j}}^{n} |L_{ij}|,$$

for any $i = 1, ..., n$. Thus L is diagonally dominant, and therefore positive semidefinite (see, e.g., Proposition 1.8 in [31]).

In the following definition, we associate a graph to a matrix.

Definition 3.8 *Consider a symmetric matrix* $A \in \mathbb{R}^{n \times n}$*. The graph associated with* A *is constructed as follows: Set the vertex set* V *to* $[n]$*. Nodes* i *and* j $(i \neq j)$ *are adjacent if* $A_{ij} \neq 0$*.* □

The associated graph of a matrix A is an object that helps us to study the sparsity of the matrix. One of the advantages of studying the associated graph is helping to check whether the matrix is PSD or not, based on the following theorem.

Theorem 3.6 *(see, e.g., [118, Theorem 9.2]) For any* $\mathcal{K} = \{\mathcal{K}(1), ..., \mathcal{K}(r)\} \subseteq \{1, ..., n\}$, *let* $P_{\mathcal{K}}$ *denotes the* $r \times n$ *matrix with entries*

$$(P_{\mathcal{K}})_{ij} = \begin{cases} 1 & \text{if } j = \mathcal{K}(i) \\ 0 & \text{otherwise.} \end{cases}$$

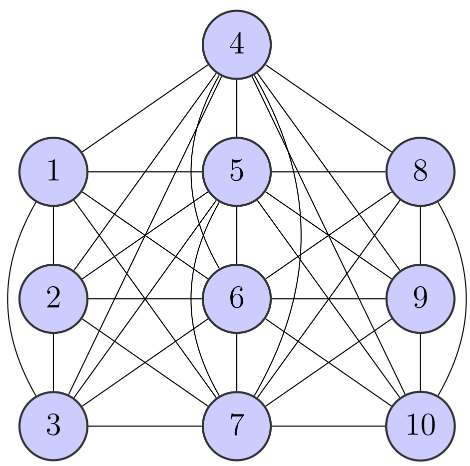
Let $A \in \mathbb{R}^{n\times n}$ *be a symmetric matrix and* G *be its associated graph. Assume that* G *is chordal. Then,* A *is PSD if and only if it can be written as* $A = \sum_{\mathcal{K}\in\mathcal{K}_G} P_{\mathcal{K}}^T H_{\mathcal{K}} P_{\mathcal{K}}$, *where* $H_{\mathcal{K}}$ *is symmetric and PSD.* □

We illustrate Theorem 3.6 in the following example.

Example 3.3 *Consider the PSD matrix*

$$A = \begin{bmatrix} 86 & 51 & 74 & 62 & 50 & 83 & 54 & 0 & 0 & 0 \\ 51 & 44 & 52 & 57 & 47 & 60 & 30 & 0 & 0 & 0 \\ 74 & 52 & 95 & 74 & 57 & 78 & 45 & 0 & 0 & 0 \\ 62 & 57 & 74 & 173 & 152 & 156 & 119 & 48 & 67 & 71 \\ 50 & 47 & 57 & 152 & 169 & 144 & 118 & 54 & 75 & 80 \\ 83 & 60 & 78 & 156 & 144 & 177 & 129 & 52 & 74 & 65 \\ 54 & 30 & 45 & 119 & 118 & 129 & 148 & 58 & 82 & 73 \\ 0 & 0 & 0 & 48 & 54 & 52 & 58 & 44 & 59 & 53 \\ 0 & 0 & 0 & 67 & 75 & 74 & 82 & 59 & 89 & 67 \\ 0 & 0 & 0 & 71 & 80 & 65 & 73 & 53 & 67 & 79 \end{bmatrix}.$$

The associated graph to A is



with maximal cliques $\mathcal{K}_1 = \{1, ..., 7\}$ and $\mathcal{K}_2 = \{4, ..., 10\}$. One can check that A can be written as

$$A = P_{\mathcal{K}_1}^T H_{\mathcal{K}_1} P_{\mathcal{K}_1} + P_{\mathcal{K}_2}^T H_{\mathcal{K}_2} P_{\mathcal{K}_2},$$

where

$$H_{\mathcal{K}_1} = \begin{bmatrix} 86 & 51 & 74 & 62 & 50 & 83 & 54 \\ 51 & 44 & 52 & 57 & 47 & 60 & 30 \\ 74 & 52 & 95 & 74 & 57 & 78 & 45 \\ 62 & 57 & 74 & 86 & 68 & 83 & 42 \\ 50 & 47 & 57 & 68 & 73 & 71 & 42 \\ 83 & 60 & 78 & 83 & 71 & 102 & 60 \\ 54 & 30 & 45 & 42 & 42 & 60 & 45 \end{bmatrix}, H_{\mathcal{K}_2} = \begin{bmatrix} 87 & 84 & 73 & 77 & 48 & 67 & 71 \\ 84 & 96 & 73 & 76 & 54 & 75 & 80 \\ 73 & 73 & 75 & 69 & 52 & 74 & 65 \\ 77 & 76 & 69 & 103 & 58 & 82 & 73 \\ 48 & 54 & 52 & 58 & 44 & 59 & 53 \\ 67 & 75 & 74 & 82 & 59 & 89 & 67 \\ 71 & 80 & 65 & 73 & 53 & 67 & 79 \end{bmatrix}.$$

□

Until now, we mentioned some results on chordal graphs and the connection of sparse PSD matrices with chordal graphs. In the rest of this section, we discuss how one can use these results and connections step by step to find the sparsity pattern of a graph G.

We denote the permuted Laplacian matrix of a graph G according to a vertex ordering ϕ by L_ϕ. As it is proved in [118, Section 9.1], using the Cholesky factorization $L_\phi = R^T R$, the nonzero entries of $R + R^T$ correspond to the edges in E_ϕ^*.

A vertex ordering that minimizes $|E_\phi^* \backslash E|$ over all possible vertex orderings of G is called a minimum ordering. Finding a minimum ordering is known to be intractable [118, Section 6.6]. There are many polynomial-time algorithms to find a "good" ordering, see [118, Section 6.6]. *Minimum degree* ordering is such an algorithm, which finds the vertex v with the least degree, set $\phi(v) = i$ and delete v from the graph G in the ith iteration. In our numerical results, we use the *approximate minimum degree* ordering (AMD) introduced in [9], which is known to have a lower complexity than the *minimum degree* ordering. We do not explain the algorithm here, and in our numerical experiments we use the available package called "CHOLMOD" (`https://github.com/JuliaLang/julia/blob/master/base/sparse/cholmod.jl`) in Julia 0.5 to get the ordering. This package uses AMD algorithm to find the ordering.

3.4.3 Exploiting sparsity in a polynomial optimization problem using chordal graphs

In this subsection, we construct a graph corresponding to problem (3.3) and use the results mentioned in Sections 3.4.1 and 3.4.2 to exploit sparsity that satisfies the *running intersection property*. The graph is essentially the same as the one constructed in [119].

Consider a general PO problem (3.3). A graph $G = (V, E)$ associated to this problem can be constructed as follows:

- the vertex set $V := \{x_1, ..., x_n\}$,
- $E_j := \{(x_i, x_k) : \text{variables } x_i, x_k \text{ are present in the definition of } g_j(x)\}$, $j \in [m]$,
- $E_0 := \{(x_i, x_k) : \text{product } x_i x_k \text{ is present in the definition of } f(x)\}$,
- the edge set $E := \bigcup_{j=0}^{m} E_j$.

Let L be the Laplacian matrix of $G = (V, E)$. Using the results in Sections 3.4.1 and 3.4.2, in order to find the sparsity pattern for the PO problem that satisfies the *running intersection property*, one can use Algorithm 1.

Algorithm 1 Exploit sparsity in the PO (3.3) using chordal graphs

$\phi \leftarrow$ get the ordering from AMD algorithm on graph G
$L_\phi \leftarrow$ reorder Laplacian matrix L according to ϕ
use Cholesky factorization of L_ϕ to construct a chordal extension G^* of G
$\{\mathcal{D}_1, ..., \mathcal{D}_q\} \leftarrow$ the maximal cliques of G^*
for all $l \in [q]$ **do**
  $\mathcal{C}_l \leftarrow$ all g_j in $\mathbb{R}[x, \mathcal{D}_l]$
end for

Theorem 3.7 *There is a bijection $\Gamma : [q] \to [q]$, such that the index blocks $\mathcal{D}_{\Gamma(\ell)}$ and constraint blocks $\mathcal{C}_{\Gamma(\ell)}$, $\ell \in [q]$, constructed by Algorithm 1 satisfy the running intersection property for the PO problem (Definition 3.1).*

Proof. By Corollary 3.1, there is a bijection $\Gamma : [q] \to [q]$ such that

$$\forall \Gamma(\ell) \in [q-1],\ \exists \Gamma(s) \leq \Gamma(\ell) : \left(\mathcal{D}_{\Gamma(\ell)+1} \cap \bigcup_{r=1}^{\ell} \mathcal{D}_{\Gamma(r)} \right) \subseteq \mathcal{D}_{\Gamma(s)}.$$

Now, we show that $f = \sum_{\ell=1}^{q} f^\ell$, where $f^\ell \in \mathbb{R}[x; \mathcal{D}_{\Gamma(\ell)}]$, for all $\ell \in [q]$. Because f is a polynomial, it is sufficient to show that for each monomial in the definition of f there is an ℓ such that the monomial belongs to $\mathbb{R}[x; \mathcal{D}_\ell]$. Let x^β be a monomial in the definition of f, where $\beta \in \mathbb{N}^n$. Due to the structure of E_0, the graph induced by the vertices corresponding to x^β is complete and hence contained in one of the maximal cliques of G^*. Therefore, there is an $\ell \in [q]$ such that $x^\beta \in \mathbb{R}[x; \mathcal{D}_\ell]$.

By the construction of $\mathcal{C}_l$, it is clear that $g_j \in \mathbb{R}[x, \mathcal{D}_\ell]$ for all $j \in \mathcal{C}_l$, $l \in [q]$, and $\bigcup_{\ell=1}^{q} \mathcal{C}_\ell = [m]$. Also, $\bigcup_{\ell=1}^{q} \mathcal{D}_\ell = [n]$, because $\{\mathcal{D}_1, ..., \mathcal{D}_q\}$ is the set of all maximal cliques of G^*. □

In Theorem 3.7, we showed that there is an ordering of $[q]$ with which the blocks $\mathcal{D}_\ell$ and $\mathcal{C}_\ell$, $\ell \in [q]$ satisfy the *running intersection property.* In the following lemma, we show that we do not need to know the ordering to solve problem (3.5).

Lemma 3.1 *Let $\Gamma : [q] \to [q]$ be a bijection for some $q \in \mathbb{N}$, $\mathcal{D}_\ell \subseteq [n]$ and $\mathcal{C}_\ell \subseteq [m]$ for all $\ell \in [q]$. Then for each $\kappa, d \in \mathbb{N}$, $\tilde{q}_d^\kappa$ in (3.5) is given by*

$$\tilde{q}_d^\kappa = \sup \left\{ t : \begin{array}{c} f - t = \sum_{\ell\in[q]} \sum_{(\alpha,\beta)\in\mathbb{N}_d^{\Gamma(\ell)}} \lambda_{\alpha\beta}^{\Gamma(\ell)} h_{\alpha\beta}^{\Gamma(\ell)} + \sum_{\ell\in[q]} \sigma_{\Gamma(\ell)} \\ \sigma_\ell \in \Sigma[x;\mathcal{D}_\ell]_\kappa, \;\; \lambda^\ell \geq 0, \; t\in\mathbb{R}, \; \ell\in[q] \end{array} \right\}. \tag{3.7}$$

Proof. The summations in (3.7) are over $\ell \in [q]$, and may change the order of summations. In other words:

$$\begin{aligned} &\sum_{\ell\in[q]} \sum_{(\alpha,\beta)\in\mathbb{N}_d^{\Gamma(\ell)}} \lambda_{\alpha\beta}^{\Gamma(\ell)} h_{\alpha\beta}^{\Gamma(\ell)} = \sum_{\ell\in[q]} \sum_{(\alpha,\beta)\in\mathbb{N}_d^{\ell}} \lambda_{\alpha\beta}^{\ell} h_{\alpha\beta}^{\ell}, \\ &\sum_{\ell\in[q]} \sigma_{\Gamma(\ell)} = \sum_{\ell\in[q]} \sigma_\ell. \end{aligned}$$

□

Theorems 3.1, 3.7 and Lemma 3.1 show that if:

- $g_j(x) \leq 1$ for any feasible solution x, $j \in [m]$,
- for all $\ell \in [q]$, the ring of $\mathbb{R}[x;\mathcal{D}_\ell]$ is generated by $\{1, (g_j)_{j\in\mathcal{C}_\ell}\}$,
- there exists $M_\ell > 0$ and $j \in \mathcal{C}_\ell$ such that $g_j = 1 - \frac{1}{M_\ell}\left(\sum_{i\in\mathcal{D}_\ell} x_i^2\right)$,

then for a fixed $\kappa \in \mathbb{N}$, $\{\tilde{q}_d^\kappa\}$ is a non-decreasing sequence that converges to the optimal value of (3.3), when $\mathcal{D}_\ell$ and $\mathcal{C}_\ell$ are the outputs of Algorithm 1.

The result of this section can be applied to the P-formulation (2.11) by elimination of equality constraints proposed in Section 2.3.2. The following example shows how Algorithm 1 works for "Haverly1" and "Adhya1", two pooling problem instances, after elimination of the equality constraints.

Example 3.4 *"Haverly1" is a pooling problem instance with 3 inputs, 2 outputs and 1 pool, where the inputs are characterized with only 1 specification. Recall from Example 2.1 on page 33 that the formulation of this instance after elimination of the equality constraints, proposed in Section 2.3.2, is*

$$\begin{aligned} \min \quad & -200x_2(15x_1 - 12) - 200x_3(15x_1 - 6) + 200x_4 - 1000x_5 \\ s.t. \quad & 1 \geq -\frac{3}{4}(x_1 - 1)(x_2 + x_3) \geq 0 \end{aligned} \tag{3.8a}$$

$$1 \geq \frac{1}{4}(3x_1 - 1)(x_2 + x_3) \geq 0 \tag{3.8b}$$
$$1 \geq 1 - 2(x_2 + x_4) \geq 0 \tag{3.8c}$$
$$1 \geq 1 - (x_3 + x_5) \geq 0 \tag{3.8d}$$
$$1 \geq \frac{1}{2}(x_4 + x_2) - \frac{2}{5}x_4 - \frac{3}{5}x_1x_2 \geq 0 \tag{3.8e}$$
$$1 \geq \frac{1}{2}(x_5 + x_3) - \frac{2}{3}x_5 - x_1x_3 \geq 0 \tag{3.8f}$$
$$1 \geq x_i \geq 0, \quad i = 1, ..., 5. \tag{3.8g}$$

For this problem, the Laplacian matrix corresponding to its graph G (Figure 3.3(a)) is

$$L = \begin{bmatrix} 4 & -1 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 & 0 \\ -1 & -1 & 3 & 0 & -1 \\ -1 & -1 & 0 & 2 & 0 \\ -1 & 0 & -1 & 0 & 2 \end{bmatrix}.$$

The output of the AMD algorithm is $\phi(\{x_1, x_2, x_3, x_4, x_5\}) = [5, 3, 1, 4, 2]$. Using the Cholesky factorization, one finds out that there is no need to add any extra edge, so G is chordal with the maximal cliques $\mathcal{D}_1 = \{x_1, x_2, x_3\}$, $\mathcal{D}_2 = \{x_1, x_2, x_4\}$, and $\mathcal{D}_3 = \{x_1, x_3, x_5\}$. Hence,

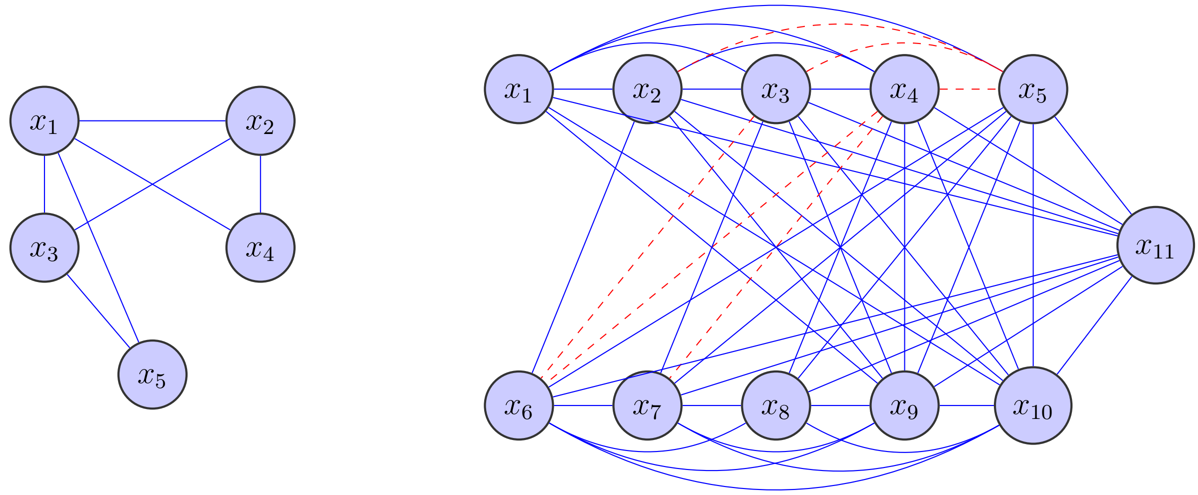
$$\begin{aligned} \mathcal{C}_1 &= \{(3.8a), (3.8b), (3.8g)_1, (3.8g)_2, (3.8g)_3\}, \\ \mathcal{C}_2 &= \{(3.8c), (3.8e), (3.8g)_1, (3.8g)_2, (3.8g)_4\}, \\ \mathcal{C}_3 &= \{(3.8d), (3.8f), (3.8g)_1, (3.8g)_3, (3.8g)_5\}, \end{aligned}$$

where $(3.8g)_i$ is the constraint $(3.8g)$ for x_i, $i = 1, ..., 5$.

"Adhya1" is a pooling problem instance that has 5 inputs, 4 outputs, 2 pools where the inputs are characterized with 4 specifications. After elimination of the equality constraints, the problem contains 11 variables and 41 constraints. The graph in Figure 3.3(b) shows G where the red dashed arcs are corresponding to the nonzero entries of $R+R^T$ that are zeros in L_ϕ. This means that the red dashed arcs are added to make the graph chordal. For G^*, the maximal cliques are

$$\begin{aligned} \mathcal{D}_1 &= \{x_1, x_2, x_3, x_4, x_5, x_9, x_{10}, x_{11}\}, \\ \mathcal{D}_2 &= \{x_2, x_3, x_4, x_5, x_6, x_9, x_{10}, x_{11}\}, \\ \mathcal{D}_3 &= \{x_3, x_4, x_5, x_6, x_7, x_9, x_{10}, x_{11}\}, \\ \mathcal{D}_4 &= \{x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}\}. \end{aligned} \tag{3.9}$$

Elimination of the equality constraints may destroy the sparsity pattern of a general PO problem. So, in the following section we study algebraic sets with equality constraints and prove a *Positivstellensatz* that deals with equality constraints.



(a) Haverly1 **(b)** Adhya1

Figure 3.3: The graphs corresponding to “Haverly1” and “Adhya1” after elimination of the equality constraints. The red dashed arcs are added to make the graph chordal.

3.5 Problems with equality constraints

Consider the following algebraic set:

$$\mathcal{F} := \{x \in \mathbb{R}^n : \ \ e_t(x) = 0, \ t \in [T], \ \ g_i(x) \geq 0, \ i \in [m]\}, \tag{3.10}$$

where $e_t, g_i \in \mathbb{R}[x, [n]]$, $t \in [T]$, $i \in [m]$. In the next theorem we show that one can slightly change the Krivine's *Positivstellensatz* [84] in order to handle algebraic set $\mathcal{F}$.

Theorem 3.8 *Assume that $g_i(x) \leq 1$, for all $x \in \mathcal{F}$, $i \in [m]$. If a polynomial $f(x)$ is positive on $\mathcal{F}$ and the ring of polynomials $\mathbb{R}[x, [n]]$ is generated by $\{1, e_t(x), g_i(x)\}_{\substack{i \in [m] \\ t \in [T]}}$, then there is an integer d such that*

$$f(x) = \sum_{(\gamma,\alpha,\theta,\beta) \in \mathbb{N}_d^{2m+2T}} \lambda_{\gamma,\alpha,\theta,\beta} \prod_{t \in [T]} e_t(x)^{\gamma_t} \left(1 - e_t(x)\right)^{\theta_t} \prod_{i \in [m]} g_i(x)^{\alpha_i} \left(1 - g_i(x)\right)^{\beta_i},$$

for some λ that

$$\begin{cases} \lambda_{\gamma,\alpha,\theta,\beta} \geq 0 & \text{if } \gamma = 0 \text{ or } T = 0, \\ \lambda_{\gamma,\alpha,\theta,\beta} \in \mathbb{R} & \text{otherwise.} \end{cases}$$

Proof. We prove this theorem by induction on T, the number of equality constraints. If $T = 0$, there is no equality constraints in $\mathcal{F}$. So, the result follows directly from Krivine's *Positivstellensatz* [84]. Now, assume that the result holds for all sets in the form of (3.10) with T equality constraints and we prove it for a set $\mathcal{F}$ with $T+1$ equality constraints.

Setting $g(x) := e_{T+1}(x)$, we can write $\mathcal{F}$ as follows:

$$\{x: \ \ g(x) \geq 0, \ \ -g(x) \geq 0, \ \ g_i(x) \geq 0, \ i = 1, ..., m, \ \ e_t(x) = 0, \ t = 1, ..., T\}.$$

So, by the induction hypothesis, there is an integer d such that

$$f(x) = \sum_{\substack{\bar\alpha = (\alpha_0, \alpha_{00}, \alpha, \gamma) \\ \bar\beta = (\beta_0, \beta_{00}, \beta, \theta) \\ \lambda_{\bar\alpha, \bar\beta} \in \mathbb{N}_d^{2m+2T+4}}} \lambda_{\bar\alpha,\bar\beta}\, g(x)^{\alpha_0} \left(-g(x)\right)^{\alpha_{00}} \left(1 + g(x)\right)^{\beta_{00}} h_{\beta_0\alpha\beta\gamma\theta}(x),$$

where,

$$h_{\beta_0\alpha\beta\gamma\theta}(x) = \left(1 - g(x)\right)^{\beta_0} \prod_{i=1}^{m} g_i(x)^{\alpha_i} \left(1 - g_i(x)\right)^{\beta_i} \prod_{t=1}^{T} e_t(x)^{\gamma_t} \left(1 - e_t(x)\right)^{\theta_t},$$

and

$$\begin{cases} \lambda_{\bar\alpha,\bar\beta} \geq 0 & \gamma = 0 \text{ or } T = 0, \\ \lambda_{\bar\alpha,\bar\beta} \in \mathbb{R} & \text{otherwise.} \end{cases}$$

So, $f(x)$ can be written as

$$\sum_{\substack{\bar\alpha = (\alpha_0, \alpha_{00}, \alpha, \gamma) \\ \bar\beta = (\beta_0, \beta_{00}, \beta, \theta) \\ \lambda_{\bar\alpha, \bar\beta} \in \mathbb{N}_d^{2m+2T+4}}} (-1)^{\alpha_{00}} \lambda_{\bar\alpha,\bar\beta}\, g(x)^{\alpha_0 + \alpha_{00}} \left(1 + g(x)\right)^{\beta_{00}} h_{\beta_0\alpha\beta\gamma\theta}(x).$$

By using binomial theorem for $(1 + g(x))^{\beta_{00}}$, we have

$$f(x) = \sum_{\substack{\bar\alpha = (\alpha_0, \alpha_{00}, \alpha, \gamma) \\ \bar\beta = (\beta_0, \beta_{00}, \beta, \theta) \\ \lambda_{\bar\alpha, \bar\beta} \in \mathbb{N}_d^{2m+2T+4}}} (-1)^{\alpha_{00}} \lambda_{\bar\alpha,\bar\beta}\, g(x)^{\alpha_0 + \alpha_{00}} \left(\sum_{j=0}^{\beta_{00}} a_j g(x)^j\right) h_{\beta_0\alpha\beta\gamma\theta}(x),$$

where a_j, $j = 0, ..., \beta_{00}$, are the binomial coefficients and therefore positive. This means that

$$f(x) = \sum_{\substack{\bar\alpha = (\alpha_0, \alpha_{00}, \alpha, \gamma) \\ \bar\beta = (\beta_0, \beta_{00}, \beta, \theta) \\ \lambda_{\bar\alpha, \bar\beta} \in \mathbb{N}_d^{2m+2T+4}}} \sum_{j=0}^{\beta_{00}} (-1)^{\alpha_{00}} a_j \lambda_{\bar\alpha,\bar\beta}\, g(x)^{\alpha_0 + \alpha_{00} + j} h_{\beta_0\alpha\beta\gamma\theta}(x).$$

Let us fix $\beta_0, \alpha, \beta, \gamma, \theta$ such that $k := \beta_0 + |\alpha| + |\beta| + |\gamma| + |\theta| \leq d$, and set

$$\chi_{\beta_0,\alpha,\beta,\gamma,\theta}(x) := \sum_{(\alpha_0,\alpha_{00},\beta_{00}) \in \mathbb{N}_{d-k}^3} \sum_{j=0}^{\beta_{00}} (-1)^{\alpha_{00}} a_j \lambda_{\bar\alpha,\bar\beta}\, g(x)^{\alpha_0 + \alpha_{00} + j}.$$

The coefficient of $g(x)^l$, $l = 0, ..., d-k$, in $\chi_{\beta_0,\alpha,\beta,\gamma,\theta}(x)$ is the summation of some $a_j\lambda_{\bar{\alpha},\bar{\beta}}$ and $-a_j\lambda_{\bar{\alpha},\bar{\beta}}$ corresponding to different $\bar{\beta}$ and j. If $l = 0$ and $\gamma = 0$, or $l = 0$ and $T = 0$, then, $\alpha_0 = \alpha_{00} = j = 0$, which means the coefficient of $g(x)^0$ is nonnegative. Hence,

$$\begin{aligned} f(x) &= \sum_{\substack{(\beta_0,\alpha,\beta,\gamma,\theta) \\ k\leq d}} \chi_{\beta_0,\alpha,\beta,\gamma,\theta}(x) h_{\beta_0,\alpha,\beta,\gamma,\theta}(x) \\ &= \sum_{\substack{(\beta_0,\alpha,\beta,\gamma,\theta) \\ k\leq d}} \sum_{l=0}^{d-k} \bar{\lambda}_{l,\beta_0,\alpha,\beta,\gamma,\theta}\, g(x)^l h_{\beta_0,\alpha,\beta,\gamma,\theta}(x), \end{aligned}$$

for some $\bar{\lambda}$ with real components such that $\bar{\lambda}_{l,\beta_0,\alpha,\beta,\gamma,\theta}$ is nonnegative if $l = 0$ and $\gamma = 0$ or $l = 0$ and $T = 0$. So, combining the two summations completes the proof. □

Theorem 3.8 asserts that the coefficients corresponding to the polynomial-multiplications

$$\prod_{t\in[T]} e_t(x)^{\gamma_t} (1 - e_t(x))^{\theta_t} \prod_{i\in[m]} g_i(x)^{\alpha_i} (1 - g_i(x))^{\beta_i},$$

with $\gamma \neq 0$, are unrestricted.

Remark 3.2 *Applying Theorem 3.8 to [120, Theorem 1], if a PO problem with feasible region (3.10) satisfies the assumptions of Theorem 3.1, then the part of linear variable λ^ℓ in (3.4) associated with the polynomial-multiplications containing equality constraints is unrestricted, and all of the convergence results in [120] are valid.* □

Remark 3.3 *Considering Theorem 3.8 and Remark 3.2, one can easily construct the corresponding graph to any PO problem with some equality constraints, and exploit the sparsity that satisfies the running intersection property, as described in Section 3.4.* □

For a pooling problem, let G be the graph of the P-formulation (2.11) on page 26 that is constructed with the procedure in Section 3.4.3. All nodes in G are corresponding to a variable in the P-formulation (2.11). Because of the constraint (2.11c), nodes corresponding to y_{il}, y_{ij}, $(i,l), (i,j) \in \mathcal{A}$ are connected in G, for each $i \in \mathcal{I}$. We denote by $\mathcal{K}_i$, $i \in \mathcal{I}$, this type of cliques. The nodes corresponding to y_{ij}, y_{lj}, $(i,j), (l,j) \in \mathcal{A}$, for each $j \in \mathcal{J}$ are connected because of (2.11e), and we denote the cliques by $\mathcal{K}_j$, $j \in \mathcal{J}$. In the same way because of (2.11d), the nodes y_{il}, y_{lj}, $(i,l), (l,j) \in \mathcal{A}$, make the cliques $\mathcal{K}_l$ for each $l \in \mathcal{L}$. If there is an arc in $\mathcal{A}$ between two units, then their corresponding cliques have a node in common. This means that the overlaps between the cliques in G are related to the arcs in $\mathcal{A}$. Let $\bar{G}$ be the network (Figure 2.1 on page 25) of the pooling problem. The latter discussion shows that the more sparse is $\bar{G}$, the fewer overlaps are between the cliques in G. This means that if $\bar{G}$ is sparse then the possibility that in the sparse-BSOS hierarchy the matrix variables have fewer overlaps is high and therefore in this case each level of the sparse-BSOS hierarchy can be solved faster than the same level of the BSOS one.

3.6 Numerical result

The results in Sections 3.4 and 3.5 have been implemented in a Julia 0.5 package called "*Polyopt*", available on `https://github.com/MOSEK/Polyopt.jl`.

In the implementation of the BSOS hierarchy, we model the polynomial equality by equating the monomials coefficients. To construct problems that satisfy the assumptions of Theorem 3.1, we add to the problems the constraints

$$1 - \frac{1}{M_\ell}\left(\sum_{i \in \mathcal{D}_\ell} x_i^2\right) \geq 0 \quad \ell \in [q].$$

The rest of this section is split into two parts, each of which presents the evaluation of our results on a class of optimization problems.

3.6.1 The evaluation on the pooling problems using the P-formulation

In this subsection, we present the numerical evaluation of the sparse-BSOS hierarchy on the P-formulation of the pooling problem instances and compare it with the BSOS hierarchy (Section 2.2). In all tables, #var., #const., and **bold** numbers mean the number of linear variables, the number of constraints, and **the optimal value of the instance**, respectively.

In the numerical experiments, we consider $\kappa = 1$ in (3.5). We compare the results of applying the BSOS and sparse-BSOS hierarchies to this formulation when the equality constraints are eliminated (Section 2.3.2), and when they are handled directly using Theorem 3.8. The time in the tables contains the time of constructing the level of the hierarchy and solving it by Mosek 8.0 [12] in seconds.

Table 3.1 presents the results of solving different pooling problem instances with the sparse-BSOS hierarchy (3.5) and BSOS hierarchy [88].

The comparison has been made in two ways: the columns that are denoted by "with elimination" contain the result of applying the corresponding hierarchy to the pooling problem instances using the elimination method proposed in Section 2.3.2. In the other columns, we use Theorem 3.8 to handle the equality constraints directly. For a few instances, such as sppA0, the time that is mentioned in Table 3.1 is larger for the sparse-BSOS hierarchy than the one for the BSOS hierarchy. This is due to the overlap of the matrix variables in the sparse-BSOS hierarchy. The dash "-" in Table 3.1 means we cannot solve the corresponding level of the hierarchy, due to the size of the problem.

After elimination of equality constraints, the constraints and variables are reduced. This means that in this case, applying Theorem 3.8 is not worthwhile with respect to the time, because the solver needs to solve a larger problem. Comparing the columns in Tables 3.1 shows that using Theorem 3.8 does not necessarily result in better or worse lower bounds. For Adhya4, applying this theorem results in better lower bounds both in the sparse-BSOS and BSOS hierarchies, but this is not the case for Adhya1.

As it can be seen in Tables 3.1, the sparse-BSOS hierarchy may construct worse lower bounds compared to the BSOS hierarchy, which can be seen on the second level of the sparse-BSOS hierarchy of Haverly1-3, Ben-Tal4, Adhya1-3. The advantage of the sparse-BSOS hierarchy is that each level of the hierarchy can be solved relatively faster than the BSOS one, if the problem is sparse. For Foulds2, the lower bounds from the BSOS and sparse-BSOS hierarchies are close but the time in the sparse-BSOS hierarchy is much less than in the BSOS one.

The intuition behind the sparse-BSOS hierarchy is to split the variables into some blocks that contain only some (and not all) variables. If the blocks have many variables in common, then it does not matter much if we merge them together. In Table 3.2 we present the results of solving the P-formulation of the pooling problem instances when two blocks are merged if the number of variables in their intersection is greater than 75% of the size of the smallest one.

Table 3.1: Comparing the BSOS hierarchy and sparse-BSOS hierarchies with $\kappa = 1$ for the P-formulation. The columns denoting by "with elimination" apply the hierarchy after elimination of equality constraints. The number of maximal cliques (q) in each level is presented between parentheses. The time of the model construction and the solution time is presented between the brackets. Boldfaced entries indicate the (approximate) optimal value.

	iteration	BSOS with elimination	BSOS	SBSOS with elimination	SBSOS
Haverly1	d=1	−600.00 [0.02]	−600.00 [0.03]	−600.00(3) [0.01]	−600.00(3) [0.03]
	d=2	−417.20 [0.04]	−417.20 [0.05]	−509.09(3) [0.02]	−505.00 [0.04]
	d=3	**-400.00** [0.10]	**-400.00** [0.15]	**-400.00**(3) [0.06]	**-400.00**(3) [0.07]

Haverly2	d=1	−1200.00 [0.03]	−1200.00 [0.03]	−1200.00(3) [0.02]	−1200.00(3) [0.03]
	d=2	−601.67 [0.05]	**-600.00** [0.05]	−1,054.55(3) [0.05]	−1,054.55(3) [0.05]
	d=3	**-600.00** [0.16]	**-600.00** [0.21]	−634.65(3) [0.08]	−628.98(3) [0.09]
	d=4	**-600.00** [2.63]	**-600.00** [3.28]	−601.09(3) [0.44]	−600.51(3) [0.52]
Haverly3	d=1	−875.00 [0.03]	−875.00 [0.03]	−875.00(3) [0.03]	−875.00(3) [0.03]
	d=2	**-750.00** [0.04]	**-750.00** [0.05]	−811.98(3) [0.04]	−810.80(3) [0.05]
	d=3	**-750.00** [0.10]	**-750.00** [0.16]	−756.95(3) [0.06]	−754.01(3) [0.06]
	d=4	**-750.00** [1.12]	**-750.00** [1.91]	**-750.00**(3) [0.20]	**-750.00**(3) [0.31]
Ben-Tal4	d=1	−650.00 [0.03]	−650.00 [0.03]	−650.00(3) [0.03]	−650.00(3) [0.03]
	d=2	−467.20 [0.05]	−467.20 [0.05]	−558.29(3) [0.04]	−540.81(3) [0.04]
	d=3	**-450.00** [0.18]	**-450.00** [0.22]	**-450.00** (3) [0.08]	**-450.00** (3) [0.09]
Ben-Tal5	d=1	**-3500.00** [0.06]	**-3500.00** [0.15]	**-3500.00**(9) [0.05]	**-3500.00**(13) [0.09]
DeyGupte4	d=1	−4.00 [0.03]	−1.33 [0.04]	−4.00(1) [0.04]	−1.33(6) [0.05]
	d=2	−3.86 [0.19]	−1.33 [0.20]	−3.86(1) [0.19]	−1.33(6) [0.18]
	d=3	**≈-0.99** [22.92]	**-1.00** [36.77]	**≈-0.99**(1) [23.15]	≈−1.30(6) [7.29]
Foulds2	d=1	−1,200.00 [0.03]	−1,200.00 [0.07]	−1,200.00(6) [0.05]	−1,200.00(8) [0.05]
	d=2	−1,191.30 [0.22]	−1,182.80 [0.34]	−1,193.92(6) [0.15]	−1,182.80(8) [0.16]
	d=3	−1,103.10 [26.45]	≈−1,102.34 [101.43]	−1,104.53(6) [3.79]	−1,103.96(8) [3.64]
Foulds3	d=1	**-8.00** [62.26]	**-8** [138.25]	**-8**(20) [95.30]	**-8**(33) [136.94]
Foulds4	d=1	**-8.00** [61.77]	**-8** [126.13]	**-8**(22) [194.33]	**-8**(33) [137.44]
Adhya1	d=1	−999.32 [0.0.3]	−999.32 [0.05]	−999.32(4) [0.04]	−999.32(4) [0.05]
	d=2	−721.12 [0.19]	−997.63 [0.40]	−957.02(4) [0.13]	≈−998.43(4) [0.22]
	d=3	−578.27 [36.50]	≈−669.86 [210.43]	−778.72(4) [3.42]	≈−860.91(4) [17.16]
Adhya2	d=1	−798.29 [0.01]	−854.10 [0.05]	−798.29(4) [0.01]	−854.10(4) [0.04]
	d=2	−577.00 [0.24]	−853.82 [0.89]	−686.60(4) [0.12]	−854.10(4) [0.44]
	d=3	−566.52 [39.02]	≈−575.81 [1,163.57]	−573.31(4) [8.42]	−749.44(4) [99.10]
Adhya3	d=1	−882.84 [0.02]	−882.84 [0.18]	−882.84(5) [0.03]	−882.84(7) [0.13]
	d=2	−805.08 [0.64]	−882.73 [9.28]	−870.55(5) [0.31]	−882.84(7) [3.20]
Adhya4	d=1	−1055.00 [0.01]	−1003.33 [0.05]	−1055.00(5) [0.02]	−1003.33(7) [0.05]
	d=2	−1,040.00 [0.33]	−1003.33 [0.90]	−1,040.00(5) [0.20]	−1003.33(7) [0.38]
	d=3	≈−908.13 [317.94]	−893.68 [1,343.85]	−1,012.16(5) [17.46]	−982.42(7) [40.42]
RT2	d=1	−45,420.50 [0.02]	−22,578.70 [0.20]	−45,420.49 [0.02]	−22,578.70(13) [0.22]
	d=2	≈−39,287.34 [0.54]	−22,153.45 [3.42]	≈−39,291.55(2) [0.56]	−22,153.48(13) [1.66]
sppA0	d=1	−47,675.00 [193.38]	—	−47,675.00(27) [413.88]	—

For Haverly1-3, and Ben-Tal4, there is no block merging. For DeyGupte4, Foulds3-4, Adhya2-3, RT2 and sppA0, all of the blocks get merged, which results in the BSOS hierarchy. Hence, we present the results for the rest of the instances in Table 3.2. In this table, we present the number of linear variables (the number of λ^ℓ, $\ell \in [q]$), the number of constraints, and the size of the semi-definite variables (`size of PSDs`) as well as the lower bounds that we get in each level of the hierarchies. As one can see, in all the instances in Table 3.2, the elimination of the equality constraints results in the high overlap in the maximal cliques of the associated graph to the P-formulation, and therefore we have merged all the maximal cliques. For the instances Foulds2 and Adhya4, even though the number of constraints in the first level of the sparse-BSOS hierarchy is less when the equality constraints are eliminated compared to the one when the equalities are handled directly, it is other way around in the third iteration. This is because, for each $\ell \in [q]$, the constraints in $\mathbb{R}[x; \mathcal{D}_\ell]$ are fewer than the ones in $\mathbb{R}[y]$, where x is the vector of variables in the P-formulation (2.11) and y is the vector of variables in the P-formulation after eliminating equality constraints.

According to the discussion in Section 3.5, the overlaps in the matrix variables of the sparse-BSOS hierarchy corresponding to the P-formulation (2.11) is related to the sparsity of the network of the pooling problem, Figure 2.1, and the number of specifications. As one can see, for the four instances in Table 3.2, the networks of the instances are highly sparse, and therefore the possibility that each level of the sparse-BSOS hierarchy can be solved faster than the same level in the BSOS hierarchy is high. For Adhya2, the network is the same as Adhya1, but because the number of specifications is much higher in Adhya2, the overlaps of the matrix variables in the sparse-BSOS hierarchy are much higher and more than 75% of their sizes.

Table 3.2: The result of solving the P-formulation of pooling problem instances with sparse-BSOS hierarchy when two maximal cliques are merged if the intersection size is larger than 75% of the size of smallest maximal clique (the size(s) of the positive semi-definite matrix variable(s) is (are) presented in the second row of each iteration).

	iteration	SBSOS with elimination			SBSOS		
		Value ([`size of PSDs`]) [time]	#const.	#var.	Value ([`size of PSDs`]) [time]	#const.	#var.
Ben-Tal5	d=1	-3,500.00 ([30]) [0.06]	465	112	-3,500.00 ([12, 22, 29]) [0.07]	601	169

Foulds2	d=1	-1,200.00 ([19]) [0.02]	190	80	-1,200.00 ($[2\times 8, 19]$) [0.02]	220	112
	d=2	-1,191.30 ([19]) [0.17]	973	3,161	-1,182.80 ($[2\times 8, 19]$) [0.16]	1,109	2,665
	d=3	-1,103.14 ([19]) [25.97]	13,601	85,321	≈-1,102.74 ($[2\times 8, 19]$) [20.61]	14,609	49,529
Adhya1	d=1	-999.32 ([12]) [0.01]	78	86	-999.32 ($[12, 19]$) [0.04]	223	114
	d=2	-721.12 ([12]) [0.13]	543	3,656	≈-998.09 ($[12, 19]$) [0.42]	2,317	4,035
	d=3	-578.27 ([12]) [38.18]	5,105	105,996	≈-776.44 ($[12, 19]$) [71.97]	41,127	107,300
Adhya4	d=1	-1,055.00 ([17]) [0.01]	153	106	-1,003.33 ($[2\times 14, 19]$) [0.03]	290	156
	d=2	-1,040.00 ([17]) [0.33]	1,730	5,566	-1,003.33 ($[2\times 14, 19]$) [0.31]	3,253	5,194
	d=3	≈-908.13 ([17]) [360.27]	27,922	198,486	-974.55 ($[2\times 14, 19]$) [62.54]	61,401	135,764

3.6.2 The evaluation on DTOC problems

Consider a DTOC problem (3.2). If F and f are polynomials, and $\mathcal{X}$,$\mathcal{U}$ are semi-algebraic sets, then it is easy to see that (3.2) satisfies the RIP with the maximal cliques

$$\mathcal{D}_k = \{x_{k+1}, x_k, u_k\}\,,\ k = 0, ..., N-1.$$

In the numerical experiments, we consider the following DTOC problem:

$$\begin{aligned}
\min_{\substack{x_k\in\mathbb{R}, k=1,\ldots,N-1\\ u_k\in\mathbb{R}, k=0,\ldots,N-1}} \quad & \frac{1}{N-1}\sum_{k=0}^{N-1}\left(x_k^2+u_k^2\right)\\
\text{s.t.} \quad & x_{k+1} = x_k + \frac{1}{N}\left(x_k^2 - u_k\right),\ k=0,...,N-1, \\
& x^\ell \le x_k \le x^u,\ k=1,...,N-1,\\
& u^\ell \le u_k \le u^u,\ k=0,...,N-1,
\end{aligned} \tag{3.11}$$

for given $x^\ell, x^u, x_0, x_N, u^\ell, u^u$. Special cases of (3.11) were considered in [45, 52, 119].

In our numerical experiments, we compare the sparse-BSOS hierarchy, when the equality constraints are replaced by two inequalities (which we will call SBSOS without Theorem 3.8), with the sparse-BSOS hierarchy after applying Theorem 3.8 (which we will call SBSOS with Theorem 3.8). To evaluate the lower bounds that we get from the SBSOS with and without Theorem 3.8, we compare them with the bounds that we get from the global optimization solver BARON [111], and the objective value of the local solution obtained from CONOPT [53]. To pass the problems to the solvers, we use AIMMS 4.39 [40]. For a fair comparison, we report the times that MOSEK 8.0 needs to solve the level of the hierarchies, as well as the times taken by BARON and CONOPT to solve the DTOC problems in seconds. We put a maximum time limit of 120 seconds for all methods. We emphasize that we do not report the result of using the BSOS hierarchy for this problem, since the number of variables in (3.11) is large and the size of the semi-definite matrix variable in the BSOS hierarchy makes the hierarchy inefficient for this type of problems.

Table 3.3: Numerical experiments on DTOC problems (3.11) with the input data $\left(x^\ell, x^u, x_0, x_N, u^\ell, u^u\right) = (0.9, 5, 1, 1, -10, 10)$, and different N, for $\kappa = 2$, $d = 2$, . The upper bounds obtained by BARON in all cases are the same as the one obtained by CONOPT.

N	SBSOS with Theorem 3.8		SBSOS without Theorem 3.8		BARON [111]		CONOPT [53]	
	value	time	value	time	lower bound	time	upper bound	time
50	**1.6600**	0.14	**1.6600**	0.19	**1.6600**	1.06	**1.6600**	0.02
250	**1.6569**	0.78	**1.6569**	1.06	1.6544	120.00	**1.6569**	0.08
500	**1.6566**	1.65	**1.6566**	2.28	1.6201	120.00	**1.6566**	0.19
1000	**1.6564**	3.77	**1.6564**	4.88	1.5932	120.00	**1.6564**	0.80
2000	**1.6563**	9.14	**1.6563**	12.68	1.5613	120.00	**1.6563**	2.17
3000	**1.6563**	14.54	**1.6563**	22.14	1.5611	120.00	**1.6563**	5.23
4000	≈1.6563	25.12	≈1.6563	31.14	0.8118	120.00	1.6562	8.77
5000	≈1.6562	36.33	≈1.6562	54.83	0.8118	120.00	1.6562	42.20

Table 3.3 presents the results of applying the SBSOS with and without Theorem 3.8 to DTOC problems for fixed

$$\left(x^\ell, x^u, x_0, x_N, u^\ell, u^u\right) = (0.9, 5, 1, 1, -10, 10)$$

and different N, and demonstrates the advantage of applying Theorem 3.8 to the sparse-BSOS hierarchy. As one can see in the table, the lower bounds that we obtain from the SBSOS with and without Theorem 3.8 are the same but applying this theorem helps us to get the lower bounds much faster. Moreover, the lower bounds are equal to the objective values of the solutions found by both BARON and CONOPT (the upper bounds obtained by BARON in all cases are the same as the one obtained by CONOPT). Therefore, even though the solvers can not guarantee optimality of the solutions, the lower bounds from the hierarchies assure us that the solutions are globally optimal for $N = 250, 1000, 2000, 3000$. For $N = 4000, 5000$ the lower bounds that we get from the hierarchies are not precise and hence we cannot deduce that the solution is optimal; however the lower bounds are close to the objective values of the solutions.

3.7 Conclusion

In this chapter, we studied the sparse-BSOS hierarchy introduced in [120]. We first showed how to find a splitting of variables for a general polynomial optimization problem that satisfies the *running intersection property*. Then, we modified this hierarchy to handle the problems with equality constraints. The results in this chapter has been implemented in a Julia 0.5 package “Polyopt” to solve a polynomial optimization problem.

In the numerical results we compared the sparse-BSOS hierarchy with the BSOS one, when Theorem 3.8 is applied. For the P-formulation, the problems in each level of the sparse-BSOS hierarchy could be solved faster than the BSOS one if the network of the pooling problem is sparse enough and the number of specifications is small. This is the case for example in the Foulds2 instance. The quality of the lower bounds we got from the sparse-BSOS hierarchy could sometimes be worse than the BSOS hierarchy. The modification we proposed to the BSOS and sparse-BSOS hierarchies to handle equality constraints could sometimes yield much better lower bounds than the original hierarchies, like the first and second levels of the hierarchies in the DeyGupte4 instance.

Applying the sparse-BSOS hierarchy to the discrete-time optimal control problems showed that the lower bounds we obtained from the sparse-BSOS hierarchy and the modified one using Theorem 3.8 were the same, but each level of the modified hierarchy could be solved faster. Also, the lower bounds guaranteed optimality of the feasible solutions found by the solvers BARON and CONOPT.

Part II

Convex Quadratic problems with Uncertainty

CHAPTER 4

Extending the scope of robust quadratic optimization

4.1 Introduction

Many real-life optimization problems have parameters whose values are not exactly known. One way to deal with parameter uncertainty is Static Robust Optimization (SRO), which enforces the constraints to hold for all uncertain parameter values in a user specified uncertainty set. This leads to a semi-infinite optimization problem, called the static robust counterpart (SRC), which is generally (computationally) intractable. A challenge in SRO is to find a tractable reformulation of the SRC. Tractability depends not only on the functions defining the problem, but also on the uncertainty set. For a linear optimization problem with linear uncertainty, there is a broad range of uncertainty sets for which the SRC has a tractable reformulation, see [24, 66].

A natural way of extending the results for linear to quadratic optimization problems, is by keeping the functions defining the optimization problem concave in the uncertain parameters, and using convex quadratic constraints in the variable $y \in \mathbb{R}^n$ of the form

$$y^T A(\Delta) y + b(\Delta)^T y + c \leq 0, \ \ \forall \Delta \in \mathcal{Z}, \tag{4.1a}$$

where $\Delta \in \mathbb{R}^{n\times n}$ is the uncertain parameter belonging to the convex compact uncertainty set $\mathcal{Z} \subset \mathbb{R}^{n\times n}$, and where $A(\Delta) \in \mathbb{R}^{n\times n}$ and $b(\Delta) \in \mathbb{R}^n$ are affine in Δ, and $c \in \mathbb{R}$ is not uncertain.

There are many real-life problems having constraints in the form (4.1a), one of which is a *portfolio choice problem*, which tries to find a combination of asset allocations that trade off the lowest risk against the highest return. One can formulate a portfolio choice problem using the form (4.1a), where $A(\Delta)$ and $b(\Delta)$ are the covariance matrix and minus the mean vector return (possibly with a weight), respectively, both of which are affine in the uncertain parameter Δ. In this chapter, we construct a convex compact uncertainty set for Δ, using statistical tools.

In addition to the constraints in the form (4.1a), we consider conic quadratic constraints with concave uncertainty in the form

$$\sqrt{y^T A(\Delta) y} + b(\Delta)^T y + c \leq 0, \ \ \forall \Delta \in \mathcal{Z}. \tag{4.1b}$$

To the best of our knowledge, there are only a few papers treating the constraints in the forms (4.1). Moreover, the matrix $A(\Delta)$ typically is given as an uncertain linear combination of some primitive matrices with vector uncertainty. For example, the authors in [63] study constraints in the form (4.1a), where $\Delta \in \mathcal{Z} \subseteq \mathbb{R}^t$ is a vector, and $A(\Delta) = \sum_{i=1}^t \Delta_i A_i$, for given A_i, $i = 1, ..., m$. They provide exact tractable reformulations of SRCs for polyhedral and ellipsoidal uncertainty sets. The uncertainty set $\mathcal{Z}$ that we consider in this chapter is a matrix valued one, which cannot be covered by the results in that paper. In a more general setting, the authors in [65] introduce a dual problem to a general convex nonlinear robust optimization problem with concave uncertainty, and provide conditions under which strong duality holds. Furthermore, the results in [76] are similar to the results in [65].

Except for the aforementioned papers, the focus in the literature remarkably is on the constraints in the variable $y \in \mathbb{R}^n$ in the forms

$$y^T A(\Delta)^T A(\Delta) y + b(\Delta)^T y + c \leq 0, \ \ \forall \Delta \in \mathcal{Z}, \tag{4.2a}$$
$$\sqrt{y^T A(\Delta)^T A(\Delta) y} + b(\Delta)^T y + c \leq 0, \ \ \forall \Delta \in \mathcal{Z}, \tag{4.2b}$$

where $A(\Delta)$ and $b(\Delta)$ are affine in $\Delta \in \mathcal{Z}$. For example, the book [24] and papers [54] and [29] treat the constraints in the forms (4.2). The drawback of (4.2) is that the SRC is, in general, (computationally) intractable, because the functions defining the constraints are convex in the uncertain parameter Δ.

Albeit there are applications in which constraints in the forms (4.2) with convex compact uncertainty sets make sense (notably least-squares problems), there are many applications, other than portfolio choice problems, where constraints in the forms (4.1) with convex compact uncertainty sets seem quite natural:

Electrical Network Design [24, Example 8.2.6]: Electrical network design is a problem of finding the best currents and potentials in an electrical circuit with the lowest absorbed power. The absorbed power can be formulated as a quadratic function of the potentials in the form (4.1a), in which A is a function of the temperature and thus uncertain.

Chance Constraint [19, Chapter 1]: Consider a normally distributed random vector $a \in \mathbb{R}^n$. Let $y \in \mathbb{R}^n$ and $b \in \mathbb{R}$ be the vector of decision variables and a constant scalar, respectively. The *chance constraint* $Prob(a^T y + b \geq 0) \geq \alpha$ is equivalent to $0 \geq z_\alpha \sqrt{y^T \Sigma y} - y^T \mu - b$, where $\alpha \in (0, 1)$, z_α is the α percentile of the standard normal distribution, μ and Σ are the mean vector and covariance matrix of a, respectively. The usual way of acquiring μ and Σ is estimation based on historical data, resulting in estimation inaccuracy.

Quadratic Approximations: Many optimization methods, like (quasi) Newton and Sequential Quadratic Programming, use quadratic approximations of objective and constraint functions. For a twice differentiable function, this approximation can be taken using the second order truncated Taylor series. However, often the calculated gradients and Hessians are inaccurate.

It is worth mentioning that the key characteristic of the constraints in the forms (4.1) is having concave uncertainty with a *convex* uncertainty set. Constraints in the forms (4.2) can easily be formulated in the forms (4.1), but the uncertainty set is not convex anymore.

The contribution of this chapter is threefold. First, we derive tractable formulations of the SRCs of uncertain constraints in the forms (4.1) with a general convex compact matrix valued uncertainty set $\mathcal{Z}$, given in terms of its support function. We derive explicit formulas for support functions of many choices of $\mathcal{Z}$, mostly of those given in terms of matrix norms and cones. This contribution extends the results of [22] from vector to matrix uncertainty.

The second contribution is finding inner and outer tractable approximations of the SRCs of constraints in the forms (4.2) with a general compact convex uncertainty set. We do this by substituting the quadratic term in the uncertain parameter with proper upper and lower bounds that are linear in the uncertain parameter and hence are in the forms (4.1).

Thirdly, we show how to construct a natural uncertainty set consisting of the mean and (vectorized) covariance matrix by using historical data and probabilistic confidence sets. We prove for this case that the support function is semi-definite representable.

The remainder of the chapter is organized as follows. Section 4.2 introduces notations and definitions that are used throughout the chapter. In Section 4.3, we derive an exact tractable formulation for the SRC of constraints in the forms (4.1) with a wide range of uncertainty sets. In Section 4.4, we study constraints in the forms (4.2) with a general bounded uncertainty set, and provide inner and outer approximations of the SRCs. We show that the approximations are tight for some uncertain constraints. Section 4.5 is about constructing an uncertainty set using historical information and confidence sets. In Section 4.6, we apply the results of this chapter to a portfolio choice, a norm approximation, and a regression line problem. All proofs in this chapter are presented in the Appendices.

4.2 Preliminaries

In this section, we introduce the notations and definitions we use throughout the chapter. A matrix $A \in \mathbb{R}^{n\times n}$ (not necessarily symmetric) is PSD if $x^T Ax \geq 0$, for any $x \in \mathbb{R}^n$. We denote by S_n the set of all $n \times n$ symmetric matrices, and by S_n^+ its subset of all PSD matrices. For $A, B \in \mathbb{R}^{n\times n}$, the notations $A \succeq B$ and $A \succ B$ are used when $A - B \in S_n^+$ and $A - B \in int(S_n^+)$, respectively, where $int(S_n^+)$ denotes the interior of S_n^+. We denote by trace (A) the trace of A. For $A, B \in \mathbb{R}^{n\times m}$, we set $\text{vec}(A) := [A_{11}, ..., A_{1m}, ..., A_{n1}, ..., A_{nm}]^T$, and hence, $\text{trace}\left(AB^T\right) = \text{vec}(A)^T\text{vec}(B)$. For symmetric matrices $A, B \in S_n$, we set

$$\text{svec}(A) := [A_{11}, \sqrt{2}A_{12}, ..., \sqrt{2}A_{1n}, A_{22}, ..., \sqrt{2}A_{(n-1)n}, A_{nn}]^T,$$

and hence, $\text{trace}\,(AB) = \text{svec}(A)^T\text{svec}(B)$. Additionally, to represent a vector $d \in \mathbb{R}^n$ by its components, we use $[d_i]_{i=1,...,n}$. Also, we denote the zero matrix in $\mathbb{R}^{n\times m}$ and identity matrix in S_n by $0_{n\times m}$ and I_n, respectively.

We denote the singular values of a matrix $A \in \mathbb{R}^{m\times n}$ with rank r by $\sigma_1(A) \geq ... \geq \sigma_r(A) > 0$. For a vector $x \in \mathbb{R}^n$, the Euclidean norm is denoted by $\|x\|_2$. We use the following matrix norms in this chapter:

Frobenius norm: $\|A\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n A_{ij}^2}$;

l_1 **norm:** $\|A\|_1 = \sum_{i=1}^m \sum_{j=1}^n |A_{ij}|$;

l_∞ **norm:** $\|A\|_\infty = \max_{\substack{1\leq i\leq m\\ 1\leq j\leq n}} |A_{ij}|$;

spectral norm: $\|A\|_{2,2} = \sup_{\|x\|_2=1} \|Ax\|_2$;

dual norm: For a general matrix norm $\|.\|$, its dual norm is defined as $\|A\|^* = \max_{\|B\|=1} \text{trace}\left(B^T A\right)$;

trace (nuclear) norm: $\|A\|_\Sigma = \sigma_1(A) + ... + \sigma_r(A)$.

Remark 4.1 *Let $\|.\|$ be a general vector norm. Then a corresponding matrix norm can be defined as $\|vec(A)\|$ for a matrix $A \in \mathbb{R}^{m\times n}$. Frobenius, l_1, and l_∞ norms are examples of this type of matrix norms.* □

The following lemma provides the exact formulations of the dual norms corresponding to the matrix norms defined above.

Lemma 4.1 *[72, Section 5.6]*

(a) $\|A\|_F^* = \|A\|_F = \|vec(A)\|_2$.

(b) $\|A\|_1^* = \|A\|_\infty$.

(c) $\|A\|_\Sigma^* = \|A\|_{2,2} = \sigma_1(A)$. □

In the rest of this section, we recall some definitions related to optimization.

Definition 4.1 *Let $\mathcal{Y}$ be a set determined by constraints in a variable y. A set $\mathcal{S}$ determined by constraints in the variable y and additional variable x, is an inner approximation of $\mathcal{Y}$, if*

$$(x, y) \in \mathcal{S} \;\Rightarrow\; y \in \mathcal{Y}.$$

A set $\mathcal{S}$ is an outer approximation if

$$y \in \mathcal{Y} \;\Rightarrow\; \exists x : (x, y) \in \mathcal{S}.$$ □

In [24] the inner approximation is called safe approximation.

Definition 4.2 *For a convex set $\mathcal{Z}$, the support function $\delta^*_{\mathcal{Z}}(.)$ is defined as follows:*

$$\begin{aligned}
&\text{if } \mathcal{Z} \subseteq \mathbb{R}^n, && \delta^*_{\mathcal{Z}}(u) := \sup_{b \in \mathcal{Z}} \left\{ u^T b \right\}, \\
&\text{if } \mathcal{Z} \subseteq \mathbb{R}^{m\times n}, && \delta^*_{\mathcal{Z}}(W) := \sup_{A \in \mathcal{Z}} \left\{ trace\left(A W^T \right) \right\}, \\
&\text{if } \mathcal{Z} \subseteq \mathbb{R}^{m\times n} \times \mathbb{R}^n, && \delta^*_{\mathcal{Z}}(W, u) := \sup_{(A,b) \in \mathcal{Z}} \left\{ trace\left(A W^T \right) + u^T b \right\},
\end{aligned}$$

where $W \in \mathbb{R}^{m\times n}$, $u \in \mathbb{R}^n$. □

4.3 Tractable reformulation for constraints with concave uncertainty

In this section, we derive tractable formulations of the SRCs in the forms (4.1).

4.3.1 Main results

In the first theorem we provide reformulations of the SRCs for uncertain constraints in the forms (4.1).

Theorem 4.1 *Let $\mathcal{Z} \subset \mathbb{R}^{n\times n}$ be a convex, compact set. Also, let $A \in \mathbb{R}^{n\times n}$, $a, b \in \mathbb{R}^n$, and $c \in \mathbb{R}$ be given. For any $\Delta \in \mathcal{Z}$, let $A(\Delta) = A + \Delta$, $b(\Delta) = b + \Delta a$. Assume that $A(\Delta)$ is positive semi-definite (PSD), for all $\Delta \in \mathcal{Z}$, and there exists Δ in the relative interior of $\mathcal{Z}$ such that $A(\Delta)$ is positive definite. Then:*

(I) $y \in \mathbb{R}^n$ *satisfies (4.1a) if and only if there exists* $W \in \mathbb{R}^{n\times n}$ *satisfying the convex system*

$$\begin{cases} trace\,(AW) + b^T y + c + \delta^*_{\mathcal{Z}}(W + ya^T) \le 0, \\ \begin{bmatrix} W & y \\ y^T & 1 \end{bmatrix} \succeq 0_{n+1\times n+1}. \end{cases} \tag{4.3}$$

(II) $y \in \mathbb{R}^n$ *satisfies (4.1b) if and only if there exist* $W \in \mathbb{R}^{n\times n}$ *and* $\eta \in \mathbb{R}$ *satisfying the convex system*

$$\begin{cases} trace\,(AW) + b^T y + c + \delta^*_{\mathcal{Z}}(W + ya^T) + \frac{\eta}{4} \le 0, \\ \begin{bmatrix} W & y \\ y^T & \eta \end{bmatrix} \succeq 0_{n+1\times n+1}. \end{cases} \tag{4.4}$$

Proof. Appendix 4.B.1. □

Remark 4.2 *One of the assumptions in Theorem 4.1 is that* $A(\Delta)$ *is positive semi-definite. This assumption is needed to guarantee convexity of the constraint and holds in many applications. An example is when* $A(\Delta)$ *is a covariance matrix, which is estimated, e.g., by using historical data. In this case,* $A(\Delta)$ *is positive semi-definite, for all possible values of the uncertain parameter* Δ*.*

4.3.2 Support functions

According to Theorem 4.1, tractability of the SRCs in the forms (4.1) depends on the tractability of the support function of the uncertainty set $\mathcal{Z}$. In this subsection, we derive such tractable reformulations for several typical uncertainty sets.

In the following lemma we provide equivalent formulations of the support functions of the uncertainty sets constructed using standard composition rules.

Lemma 4.2 *Let* $U \in \mathbb{R}^{n\times n}$*.*

(i) Let $\mathcal{Z} = \left\{\Delta \in \mathbb{R}^{n\times n} :\ vec(\Delta) \in \mathcal{U} \subset \mathbb{R}^{n^2}\right\}$*. Then* $\delta^*_{\mathcal{Z}}(U) = \delta^*_{\mathcal{U}}(vec(U))$*.*

(ii) Let $\Delta^1, ..., \Delta^k \in \mathbb{R}^{n\times n}$ *be given. Also, let* $\mathcal{Z} = \left\{\sum_{i=1}^k \zeta_i \Delta^i :\ \zeta \in \mathcal{U} \subset \mathbb{R}^k\right\}$*. Then,* $\delta^*_{\mathcal{Z}}(U) = \delta^*_{\mathcal{U}}\left(\left[trace\left(\Delta^i U^T\right)\right]_{i=1,...,k}\right)$*.*

(iii) Let $L \in \mathbb{R}^{n\times t}$ *and* $R \in \mathbb{R}^{s\times n}$ *be given, and* $\mathcal{Z} = \{L\Delta R : \Delta \in \mathcal{U} \subseteq \mathbb{R}^{t\times s}\}$*. Then* $\delta^*_{\mathcal{Z}}(U) = \delta^*_{\mathcal{U}}\left(L^T U R^T\right)$*.*

(iv) Let $\mathcal{Z}_i \subseteq \mathbb{R}^{n\times n}$*,* $i = 1, ..., k$*, and let* $\mathcal{Z} = \sum_{i=1}^k \mathcal{Z}_i$ *be the Minkowski sum. Then* $\delta^*_{\mathcal{Z}}(U) = \sum_{i=1}^k \delta^*_{\mathcal{Z}_i}(U)$*.*

(v) *Let* $\mathcal{Z}_i \subseteq \mathbb{R}^{n\times n}$, $i = 1, ..., k$, *such that* $\bigcap_{i=1}^{k} ri(\mathcal{Z}_i) \neq \emptyset$, *where* $ri()$ *denotes the relative interior. Also, let* $\mathcal{Z} = \bigcap_{i=1}^{k} \mathcal{Z}_i$. *Then*

$$\delta^*_{\mathcal{Z}}(U) = \min_{\substack{U^i \in \mathbb{R}^{n\times n}\\ i=1,\dots,k}} \left\{ \sum_{i=1}^{k} \delta^*_{\mathcal{Z}_i}(U^i) : \ \sum_{i=1}^{k} U^i = U \right\}.$$

(vi) *Let* $\mathcal{Z}_i \subseteq \mathbb{R}^{n_i\times n_i}$, $U_i \in \mathbb{R}^{n_i\times n_i}$, $i = 1, ..., k$, *and*

$$\mathcal{Z} = \left\{ \Delta = (\Delta_1, ..., \Delta_k) : \ \ \Delta_i \in \mathcal{Z}_i, i = 1, ..., k \right\}.$$

Then we have $\delta^*_{\mathcal{Z}}\left((U_1, ..., U_k)\right) = \sum_{i=1}^{k} \delta^*_{\mathcal{Z}_i}(U_i)$.

(vii) *Let* $\mathcal{Z}_i \subseteq \mathbb{R}^{n\times n}$, $i = 1, ..., k$, *be convex and* $\mathcal{Z} = conv(\bigcup_{i=1}^{k} \mathcal{Z}_i)$ *be the convex hull. Then* $\delta^*_{\mathcal{Z}}(U) = \max_{i=1,\dots,k} \delta^*_{\mathcal{Z}_i}(U)$.

Proof. Appendix 4.B.2. □

The support functions derived in [22], which are for vector uncertainty sets, can directly be used for the support functions $\delta^*_{\mathcal{Z}}(.)$ in (4.3) or (4.4) in cases similar to the ones considered in Lemmas 4.2(i) and 4.2(ii). In the next lemma we derive tractable reformulations of matrix uncertainty sets.

Lemma 4.3 *Let* $U \in \mathbb{R}^{n\times n}$.

(a) *Let* $\mathcal{Z} = \left\{ \Delta \in \mathbb{R}^{n\times n} : \ \|\Delta\| \leq \rho \right\}$, *where* $\|.\|$ *is a general matrix norm. Then* $\delta^*_{\mathcal{Z}}(U) = \rho\|U\|^*$.

(b) *Let* $\mathcal{Z} = \{\Delta : \ \ \Delta^l \preceq \Delta \preceq \Delta^u\}$, *where* Δ^l, $\Delta^u \in S_n$ *are given such that* $\Delta^u - \Delta^l \succ 0_{n\times n}$. *Then*

$$\delta^*_{\mathcal{Z}}(U) = \min_{\Lambda_1,\Lambda_2} \left\{ trace\left(\Delta^u\Lambda_2\right) - trace\left(\Delta^l\Lambda_1\right) : \ \Lambda_2 - \Lambda_1 = \frac{U + U^T}{2}, \ \Lambda_1, \Lambda_2 \succeq 0_{n\times n} \right\}.$$

Proof. (a) It follows directly from the definition of the dual norm.
(b) Appendix 4.B.3. □

Special cases of the uncertainty sets studied in Lemma 4.3 have been considered in the literature. The uncertainty set constructed using the *Frobenius norm* is considered in [54] for the constraints in the form (4.2b). Also, the authors in [112] construct an uncertainty set for the covariance matrix using the *Frobenius norm.* The constraints in the forms (4.2) with uncertainty set defined by the *spectral norm* is treated in Chapter 6 of [24]. Furthermore, the uncertainty set that we considered in Lemma 4.3(b) is constructed in [110] for covariance matrices. Besides, the authors in [50] construct an uncertainty set for the mean vector and covariance matrix, which can be formulated as an intersection of two sets that are considered in Lemma 4.3(b).

It is known that the l_1 and l_∞ *norms* are linear representable and the *Frobenius norm* is conic quadratic representable [22]. The following lemma shows that the *spectral* and *trace norms* are semi-definite representable.

Lemma 4.4 *Let $U \in \mathbb{R}^{n\times n}$ and $\rho \geq 0$.*

(i) $\|U\|_\Sigma \leq \rho$ if and only if there exist matrices $Y \in \mathbb{R}^{n\times n}$ and $Z \in \mathbb{R}^{n\times n}$ such that

$$\begin{bmatrix} Y & U \\ U^T & Z \end{bmatrix} \succeq 0_{2n\times 2n}, \quad \mathit{trace}\,(Y) + \mathit{trace}\,(Z) \leq 2\rho.$$

(ii) $\|U\|_{2,2} \leq \rho$ if and only if

$$\begin{bmatrix} \rho^2 I_n & U \\ U^T & I_n \end{bmatrix} \succeq 0_{2n\times 2n}.$$

Proof. (i) See, e.g., Lemma 1 in [56].
(ii) See, e.g., Example 8 in [22], or Appendix 4.B.4. □

4.3.3 Illustrative examples

In this section we derive tractable reformulations of SRCs for some natural uncertain convex quadratic and conic quadratic constraints.

Example 4.1 *Let $\mathcal{Z} = \{\Delta \in \mathbb{R}^{n\times n} : \|\Delta\|_F \leq 1\}$ and let the assumptions of Theorem 4.1 hold. Then, using Theorem 4.1(I), Lemma 4.1(a), and Lemma 4.3(a), y satisfies (4.1a) if and only if there exists $W \in \mathbb{R}^{n\times n}$ such that*

$$\begin{cases} \mathit{trace}\,(AW) + b^T y + c + \left\| W + ya^T \right\|_F \leq 0, \\ \begin{bmatrix} W & y \\ y^T & 1 \end{bmatrix} \succeq 0_{n+1\times n+1}. \end{cases}$$

□

In the next example, we derive a tractable reformulation of the SRC in the form (4.1b) with the uncertainty set similar to the one proposed in [50].

Example 4.2 *Consider the constraint*

$$\sqrt{y^T A(\Delta) y} + b(\zeta)^T y + c \leq 0 \ \ \forall (\zeta, \Delta) \in \mathcal{Z}, \tag{4.5}$$

where $\zeta \in \mathbb{R}^n$, $\Delta \in \mathbb{R}^{n\times n}$ *are uncertain parameters,* $A(\Delta) = A + \Delta$, $b(\zeta) = b + D\zeta$, $A, D \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$, *and* $\mathcal{Z} = \mathcal{Z}_1 \cap \mathcal{Z}_2$,

$$\mathcal{Z}_1 = \left\{(\zeta,\Delta) : \begin{bmatrix} 1 & \zeta^T \\ \zeta & \Delta \end{bmatrix} \succeq 0_{n+1\times n+1}\right\},\ \ \mathcal{Z}_2 = \left\{(\zeta,\Delta) : \Delta^l \preceq \Delta \preceq \Delta^u\right\},$$

with given Δ^l *and* Δ^u *such that* $\Delta^u - \Delta^l \succ 0_{n\times n}$*. Also, assume that the assumptions of Theorem 4.1 hold. By Lemma 4.2(v),*

$$\delta^*_{\mathcal{Z}}(U,v) = \min_{\substack{U^1,U^2\in\mathbb{R}^{n\times n}\\ v^1,v^2\in\mathbb{R}^n}} \left\{\delta^*_{\mathcal{Z}_1}(U^1,v^1) + \delta^*_{\mathcal{Z}_2}(U^2,v^2) :\ \ U^1 + U^2 = U,\ v^1 + v^2 = v\right\}.$$

Following a similar line of reasoning as in the proof of Theorem 4.1(II), $y \in \mathbb{R}^n$ *satisfies (4.5) if and only if there exist* $W, U^1, U^2 \in \mathbb{R}^{n\times n}$, $v^1, v^2 \in \mathbb{R}^n$ *and* $\eta \in \mathbb{R}$ *such that*

$$\begin{cases} \mathit{trace}\,(AW) + b^Ty + c + \delta^*_{\mathcal{Z}_1}(v^1,U^1) + \delta^*_{\mathcal{Z}_2}(v^2,U^2) + \frac{\eta}{4} \le 0, \\ \begin{bmatrix} W & y \\ y^T & \eta \end{bmatrix} \succeq 0_{n+1\times n+1},\ U^1 + U^2 = W,\ v^1 + v^2 = D^Ty. \end{cases} \tag{4.6}$$

Using Lemma 4.3(b), (4.6) is equivalent to

$$\begin{cases} \mathit{trace}\,(AW) + b^Ty + c + \mathit{trace}\,(\Delta^u\Lambda_2) - \mathit{trace}\left(\Delta^l\Lambda_1\right) + \dfrac{\eta}{4} + \gamma \le 0, \\ U^1 + U^2 = W,\ \ \Lambda_2 - \Lambda_1 = \dfrac{U^2 + {U^2}^T}{2},\ \ \Lambda_1, \Lambda_2 \succeq 0_{n\times n}, \\ \begin{bmatrix} W & y \\ y^T & \eta \end{bmatrix} \succeq 0_{n+1\times n+1},\ \ \begin{bmatrix} \frac{U^1 + {U^1}^T}{2} & \frac{1}{2}D^Ty \\ \frac{1}{2}y^TD & -\gamma \end{bmatrix} \preceq 0_{n+1\times n+1}, \end{cases}$$

for some $\Lambda_1, \Lambda_2 \in S_n$ *and* $\gamma \in \mathbb{R}$. □

The following example is for constraints in the form (4.1a) with vector uncertainty.

Example 4.3 *Consider*

$$y^TA(\zeta)y + b(\zeta)^Ty + c \le 0,\ \ \forall \zeta \in \mathcal{Z}, \tag{4.7}$$

where $A(\zeta) = A + \sum_{i=1}^t \zeta_i A^i$, $b(\zeta) = b + \sum_{i=1}^t \zeta_i b^i$, $(A^i, b^i) \in \mathbb{R}^{n\times n} \times \mathbb{R}^n$ *is given,* $i = 1, ..., t$*. This constraint is considered in [63], where the uncertainty in* A *and* b *are independent with polyhedral or ellipsoidal, and where* A^i *is positive semi-definite,* $i = 1, ..., t$ *. In this example we show how using the results of [22] and Section 4.3.1 can extend the results of [63] for general uncertainty sets, where* $A(\zeta)$ *is positive semi-definite for all* $\zeta \in \mathcal{Z}$*.*

First, we assume that $\mathcal{Z} \subset \mathbb{R}^t_+$*, where* $\mathbb{R}^t_+$ *denotes the nonnegative orthant of* $\mathbb{R}^t$*. Also, we assume that* A_i *is positive semi-definite,* $i = 1, ..., t$*. In this case,* $y \in \mathbb{R}^n$ *satisfies (4.7) if and only if there exists* $v \in \mathbb{R}^t$ *such that*

$$y^TAy + \delta^*_{\mathcal{Z}}(v) + b^Ty + c \le 0,\ v \ge \left[y^TA^iy + {b^i}^Ty\right]_{i=1,\dots,t}, \tag{4.8}$$

where the proof can be found in Appendix 4.B.5. It is clear that in this case $A(\zeta)$ *is positive semi-definite for all* $\zeta \in \mathcal{Z}$. *Moreover, as it is mentioned in Remark 2 of [63], if* $b^i = 0$, $i = 1, ..., t$, *then, for a general uncertainty set* $\mathcal{Z}$, $y \in \mathbb{R}^n$ *satisfies (4.7) if and only if*

$$y^T \left(A + \sum_{i=1}^{t} \zeta_i A_i \right) y + b^T y + c \leq 0, \;\; \forall \zeta \in \bar{\mathcal{Z}},$$

where $\bar{\mathcal{Z}} = \mathcal{Z} \cap \{\zeta : \zeta \geq 0_{t\times 1}\}$ *and* $\bar{\mathcal{Z}} \neq \emptyset$. *Hence,* $y \in \mathbb{R}^n$ *satisfies (4.7) if and only if*

$$y^T A y + \delta^*_{\bar{\mathcal{Z}}}(v) + b^T y + c \leq 0, \;\; v \geq \left[y^T A^i y \right]_{i=1,\dots,t},$$

which is an extension of the results of [63], since there is a broad range of uncertainty sets for which the support functions have tractable reformulations.

Now, for a general case with dependent uncertainty in A *and* b, *if* $A(\zeta)$ *is positive semi-definite for all* $\zeta \in \mathcal{Z}$, *and positive definite for a* ζ *in its relative interior, then by Theorem 4.1(I),* y *satisfies (4.7) if and only if there exists* $W \in \mathbb{R}^{n\times n}$ *such that*

$$\begin{cases} trace\,(AW) + b^T y + \delta^*_{\mathcal{Z}}\left(\left[trace\,(A^i W) + b^{i^T} y \right]_{i=1,\dots,t} \right) + c \leq 0, \\ \qquad\qquad\qquad \begin{bmatrix} W & y \\ y^T & 1 \end{bmatrix} \succeq 0_{n+1\times n+1}. \end{cases} \tag{4.9}$$

□

In the last example we derive a tractable reformulation of a quadratic constraint with an uncertainty set similar to the one constructed in Section 4.5, and used in the numerical experiments (Section 4.6.1).

Example 4.4 *Consider the uncertain quadratic constraint*

$$y^T A y + b^T y + c \leq 0, \;\; \forall \begin{pmatrix} b \\ svec(A) \end{pmatrix} \in \mathcal{Z}$$

where $\mathcal{Z} = \mathcal{Z}_1 \cap \mathcal{Z}_2$, *and*

$$\mathcal{Z}_1 = \left\{ \begin{pmatrix} b \\ svec(A) \end{pmatrix} = B\nu : \; \|\nu\|_2 \leq \rho, \; \nu \in \mathbb{R}^{\frac{n^2+3n}{2}} \right\},$$

$$\mathcal{Z}_2 = \left\{ \begin{pmatrix} b \\ svec(A) \end{pmatrix} : \; b \in \mathbb{R}^n, \; A \in S_n^+ \right\},$$

for some invertible $B \in \mathbb{R}^{\frac{n^2+3n}{2} \times \frac{n^2+3n}{2}}$, $\rho > 0$. *For a fixed* $W \in S_n$, *by Lemma 4.2(v), and Example 4 in [22],*

$$\delta^*_{\mathcal{Z}} \begin{pmatrix} u \\ svec(W) \end{pmatrix} = \begin{cases} \min\limits_{\substack{u^1,u^2 \\ W^1,W^2}} \; \rho \left\| B^T \begin{pmatrix} u^1 \\ svec(W^1) \end{pmatrix} \right\|_2 + \delta^*_{\mathcal{Z}_2} \begin{pmatrix} u^2 \\ svec(W^2) \end{pmatrix} \\ \qquad s.t. \;\; u^1 + u^2 = u, \quad W^1 + W^2 = W, \quad W^1, W^2 \in S_n. \end{cases}$$

Similar to the proofs of Lemmas 4.2(i) and 4.2(vi), we have

$$\delta^*_{\mathcal{Z}}\begin{pmatrix} u \\ svec(W) \end{pmatrix} = \min_{W^1}\left\{\rho \left\| B^T \begin{pmatrix} u \\ svec(W^1) \end{pmatrix} \right\|_2 : W^1 \succeq W \right\}. \tag{4.10}$$

Since B is invertible, it is easy to show that there exist a positive definite A and a vector b such that $\binom{b}{svec(A)}$ is in the relative interior of $\mathcal{Z}$. Hence, $y \in \mathbb{R}^n$ satisfies (4.4) if and only if there exists $W \in S_n^+$ that satisfies

$$\rho \left\| B^T \begin{pmatrix} u \\ svec\,(W) \end{pmatrix} \right\|_2 + c \le 0, \quad \begin{bmatrix} W & y \\ y^T & 1 \end{bmatrix} \succeq 0_{n+1\times n+1}.$$

□

4.4 Inner and outer approximations for constraints with convex uncertainty

In this section, we provide inner and outer approximations of the SRCs of constraints in the forms (4.2) by replacing the quadratic term in the uncertain parameter with suitable upper and lower bounds. We assume that $A(\Delta) = A + \Delta$, $b(\Delta) = b + D\Delta a$, for given $A \in \mathbb{R}^{m\times n}$, $D \in \mathbb{R}^{n\times m}$, $b \in \mathbb{R}^n$, $a \in \mathbb{R}^n$, $c \in \mathbb{R}$, and $\Delta \in \mathcal{Z} \subseteq \mathbb{R}^{m\times n}$, $\mathcal{Z}$ is convex and compact. Here, we list all assumptions on the constraints in the forms (4.2a) and (4.2b) that we will make in this section, and use some of them in each theorem.

Assumption:

(A) $0_{m\times n}$ is in the relative interior of $\mathcal{Z}$.

(B) there exists $\Omega > 0$ such that $\|\Delta\|_{2,2} \le \Omega$ for all $\Delta \in \mathcal{Z}$.

(C) $A^TA + 2\Delta^TA$ is positive semi-definite for all $\Delta \in \mathcal{Z}$.

(D) $A^TA + 2\Delta^TA$ is positive definite for some Δ in the relative interior of $\mathcal{Z}$.

The following theorem provides tractable inner approximations of the constraints in the forms (4.2) by replacing the quadratic term in the uncertain parameter with a linear upper bound.

Theorem 4.2 *Let Assumptions (A) and (B) hold. Then:*

(I) $y \in \mathbb{R}^n$ satisfies (4.2a) if there exists $W \in \mathbb{R}^{n\times n}$ satisfying the convex system

$$\begin{cases} trace\left((A^TA + \Omega^2 I_n)W\right) + \delta^*_{\mathcal{Z}}(2AW + D^Tya^T) + b^Ty + c \le 0, \\ \begin{bmatrix} W & y \\ y^T & 1 \end{bmatrix} \succeq 0_{n+1\times n+1}. \end{cases} \tag{4.11}$$

(II) $y \in \mathbb{R}^n$ *satisfies (4.2b) if there exist* $W \in \mathbb{R}^{n\times n}$ *and* $\eta \in \mathbb{R}$ *satisfying the convex system*

$$\begin{cases} \mathit{trace}\left((A^TA + \Omega^2 I_n)W\right) + \delta^*_{\mathcal{Z}}(2AW + D^T y a^T) + b^T y + c + \frac{\eta}{4} \le 0, \\ \begin{bmatrix} W & y \\ y^T & \eta \end{bmatrix} \succeq 0_{n+1\times n+1}. \end{cases} \tag{4.12}$$

Proof. Appendix 4.B.6. □

The inner approximations proposed in Theorem 4.2 can be tight. As a simple example, if $A = 0_{m\times n}$, $D = 0_{n\times m}$, and the uncertainty set is $\{\Delta : \|\Delta\|_{2,2} \le \Omega\}$, then y and W satisfy (4.11) if and only if y satisfies (4.2a). This result holds because in this case $\max_{\Delta\in\mathcal{Z}} \|\Delta y\|_2 = \Omega \|y\|_2$.

In the next theorem we derive tractable outer approximations of the constraints in the forms (4.2).

Theorem 4.3 *Let Assumptions (C) and (D) hold. Then:*

(I) if y *satisfies (4.2a), then there exists* $W \in \mathbb{R}^{n\times n}$ *satisfying the convex system*

$$\begin{cases} \mathit{trace}\left(A^TAW\right) + \delta^*_{\mathcal{Z}}(2AW + D^T y a^T) + b^T y + c \le 0, \\ \begin{bmatrix} W & y \\ y^T & 1 \end{bmatrix} \succeq 0_{n+1\times n+1}. \end{cases} \tag{4.13}$$

(II) if $y \in \mathbb{R}^n$ *satisfies (4.2b), then there exist* $W \in \mathbb{R}^{n\times n}$ *and* $\eta \in \mathbb{R}$ *satisfying the convex system*

$$\begin{cases} \mathit{trace}\left(A^TAW\right) + \delta^*_{\mathcal{Z}}(2AW + D^T y a^T) + b^T y + c + \frac{\eta}{4} \le 0, \\ \begin{bmatrix} W & y \\ y^T & \eta \end{bmatrix} \succeq 0_{n+1\times n+1}. \end{cases} \tag{4.14}$$

Proof. Appendix 4.B.7. □

In the next theorem we provide an upper bound on the violation errors of (4.2a) and (4.2b) for the solutions that satisfy the outer approximations (4.13) and (4.14), respectively.

Theorem 4.4 *Let Assumptions (B), (C), and (D) hold. Then,*

(I) if $y \in \mathbb{R}^n$ *and* $W \in \mathbb{R}^{n\times n}$ *satisfy (4.13), then* y *violates (4.2a) by at most* $\Omega^2 \|y\|_2^2$.

(II) if $y \in \mathbb{R}^n$ *and* $W \in \mathbb{R}^{n\times n}$ *satisfy (4.14), then* y *violates (4.2b) by at most* $\Omega \|y\|_2$.

Proof. Appendix 4.B.8. □

In Theorems 4.2 and 4.4 we need Assumption (B), which states that there exists an upper bound Ω for $\sup_{\Delta\in\mathcal{Z}} \|\Delta\|_{2,2}$. Notice that $\|.\|_{2,2}$ is a convex function and the maximization of a convex function over a set, in general, is intractable. In the cases for which $\sup_{\Delta\in\mathcal{Z}} \|\Delta\|_{2,2}$ cannot be computed efficiently, one may use an upper bound for $\|.\|_{2,2}$ to calculate Ω.

Furthermore, it is mentioned in Section 8.2 of [24] that finding a robust solution to an uncertain LMI, in general, is *NP-hard*. Hence, there is no efficient way to check Assumption (C) exactly. In the following proposition, we provide equivalent statements to Assumption (C).

Proposition 4.1 *Let* $B(\Delta) := A^TA + \Delta^TA + A^T\Delta$*. Then, the following statements are equivalent:*

(i) Assumption (C) holds.

(ii) $B(\Delta) \succeq 0_{n\times n}$*, for all* $\Delta \in \mathcal{Z}$*.*

(iii) $\|B(\Delta)\|_\Sigma - \mathit{trace}(B(\Delta)) \leq 0$*, for all* $\Delta \in \mathcal{Z}$*.*

Proof. Appendix 4.B.9. □

Using Proposition 4.1, depending on the structure of $\mathcal{Z}$, to check Assumption (C) one may use an upper bound for $\|B(\Delta)\|_\Sigma$ for which the robust counterpart has a tractable reformulation.

4.5 Data-driven uncertainty set

A usual way of constructing an uncertainty set is by using historical data and statistical tools, such as a hypothesis testing [36], or asymptotic confidence sets [21]. In this section, we use the latter to design an uncertainty set for a vector consisting of the mean and vectorized covariance matrix.

For notational simplicity, we explain how to construct an uncertainty set for the two dimensional case; the extension to higher dimensions is straightforward. For the two dimensional case, assume that $\binom{x}{z}$ is a random vector with components x, z, and set $\mu_x = \mathbb{E}(x)$, $\mu_z = \mathbb{E}(z)$, $\sigma_x^2 = \mathbb{E}(x-\mu_x)^2$, $\sigma_z^2 = \mathbb{E}(z-\mu_z)^2$, $\sigma_{xz} = \mathbb{E}(x-\mu_x)(z-\mu_z)$, and $\mu_{kl} = \mathbb{E}(x-\mu_x)^k(z-\mu_z)^l$, $k, l = 0, 1, 2, ...$ Assume that the fourth moments exist, which means that μ_{kl} exists when $k + l \leq 4$, $k, l = 0, 1, 2, 3, 4$. This assumption can be tested using the result in [116]. Now, consider a random sample of size n, $\binom{x_i}{z_i}$, $i = 1, ..., n$, that are independent and identically distributed. Set

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \quad \bar{z} = \frac{1}{n}\sum_{i=1}^{n} z_i, \quad S_x^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2,$$

$$S_z^2 = \frac{1}{n}\sum_{i=1}^{n}(z_i - \bar{z})^2, \quad S_{xz} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(z_i - \bar{z}).$$

Using the Central Limit Theorem (Example 2.18 in [117]) and the Delta Method (Theorem 3.1 in [117]), and setting

$$\begin{aligned} \mathcal{Y} &= \left(\mu_x,\ \mu_z,\ \mathbb{E}(x^2),\ \mathbb{E}(xz),\ \mathbb{E}(z^2)\right)^T, \\ Y_n &= \left(\bar{x},\ \bar{z},\ \frac{1}{n}\sum_{i=1}^{n} x_i^2,\ \frac{1}{n}\sum_{i=1}^{n} x_i z_i,\ \frac{1}{n}\sum_{i=1}^{n} z_i^2\right)^T, \end{aligned}$$

it follows for any differentiable function $\phi : \mathbb{R}^5 \to \mathbb{R}^m$ that $\sqrt{n}\,(\phi(Y_n) - \phi(\mathcal{Y}))$ converges in distribution to the normal distribution $N(0, \nabla\phi(\theta)\Sigma\nabla\phi(\theta)^T)$, where Σ and $\nabla\phi$ are the covariance matrix of $(x, z, x^2, xz, z^2)^T - \mathcal{Y}$ and the Jacobian matrix of ϕ, respectively. Letting

$$\phi(x_1, ..., x_5) = \left(x_1,\ x_2,\ x_3 - x_1^2,\ x_4 - x_1 x_2,\ x_5 - x_2^2\right)^T,$$

it is easy to show, similar to Example 3.2 in [117], that

$$\sqrt{n}\,(T_n - \theta) \xrightarrow[n\to\infty]{d} N\left(\begin{pmatrix}0\\0\\0\\0\\0\end{pmatrix}, \underbrace{\begin{pmatrix} \mu_{20} & \mu_{11} & \mu_{30} & \mu_{21} & \mu_{12} \\ \mu_{11} & \mu_{02} & \mu_{21} & \mu_{12} & \mu_{03} \\ \mu_{30} & \mu_{21} & \mu_{40} - \mu_{20}^2 & \mu_{31} - \mu_{11}\mu_{20} & \mu_{22} - \mu_{20}\mu_{02} \\ \mu_{21} & \mu_{12} & \mu_{31} - \mu_{11}\mu_{20} & \mu_{22} - \mu_{11}^2 & \mu_{13} - \mu_{11}\mu_{02} \\ \mu_{12} & \mu_{03} & \mu_{22} - \mu_{20}\mu_{02} & \mu_{13} - \mu_{11}\mu_{02} & \mu_{04} - \mu_{02}^2 \end{pmatrix}}_{V}\right), \tag{4.15}$$

where

$$\theta = \phi(\mathcal{Y}) = \left(\mu_x,\ \mu_z,\ \sigma_x^2,\ \sigma_{xz},\ \sigma_z^2\right)^T, \quad T_n = \phi(\mathcal{Y}_n) = \left(\bar{x},\ \bar{z},\ S_x^2,\ S_{xz},\ S_z^2\right)^T,$$

and $\xrightarrow[n\to\infty]{d}$ means convergence in distribution when the size of the random sample goes to infinity (for a precise definition of the multivariate normal distribution $N(\mu, \Sigma)$ with mean μ and covariance matrix Σ see, e.g., Section 45.2 in [83]).

Let $\hat{V}$ and $\hat{\theta}$ be consistent estimates of V and θ defined in (4.15), respectively. Then, asymptotically with $(1-\alpha)\%$ confidence, θ belongs to the following ellipsoid (see, e.g., Section 45.9 in [83]):

$$\mathcal{U} := \left\{\theta :\ n\left(\hat{\theta}-\theta\right)^T \hat{V}^{-1}\left(\hat{\theta}-\theta\right) \le \chi^2_{\text{rank}(V),1-\alpha}\right\},$$

where $\chi^2_{d,1-\alpha}$ denotes the $(1-\alpha)$ percentile of the Chi-square distribution with d degrees of freedom(for a precise definition of the univariate χ^2 distribution see, e.g., Section 18.2 in [78]).

To use the results of Section 4.3, we reformulate the uncertainty set $\mathcal{U}$. Setting

$$\begin{aligned}
&A = \begin{bmatrix} A_\mu \\ A_\Sigma \end{bmatrix},\ \ A_\mu = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix},\ \ A_\Sigma = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \sqrt{2} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \\
&\mu = [\mu_x\ \mu_z]^T,\ \ \Sigma = \begin{bmatrix} \sigma_x^2 & \sigma_{xz} \\ \sigma_{xz} & \sigma_z^2, \end{bmatrix},
\end{aligned} \tag{4.16}$$

due to positive semi-definiteness of Σ, with $(1-\alpha)\%$ confidence

$$\begin{pmatrix} \mu \\ \text{svec}(\Sigma) \end{pmatrix} \in \bar{\mathcal{U}} := \hat{\mathcal{U}} \cap \left\{\gamma :\ n\left(A\hat{\theta}-\gamma\right)^T A^{-1}\hat{V}^{-1}A^{-1}\left(A\hat{\theta}-\gamma\right) \le \chi^2_{\text{rank}(V),1-\alpha}\right\},$$

where $\hat{\mathcal{U}} = \left\{\binom{\gamma_\mu}{\gamma_\Sigma} :\ \gamma_\Sigma = \text{svec}(M),\ M \succeq 0_{n\times n}\right\}$. Letting $R^T R$ be the Cholesky factorization of $\hat{V}^{-1}$, i.e., $\hat{V}^{-1} = R^T R$, $\bar{\mathcal{U}}$ can be rewritten as

$$\begin{aligned}
\bar{\mathcal{U}} &= \hat{\mathcal{U}} \cap \left\{\gamma :\ \left\|RA^{-1}\left(\gamma - A\hat{\theta}\right)\right\|_2 \le \sqrt{\frac{\chi^2_{\text{rank}(V),1-\alpha}}{n}}\right\} \\
&= \hat{\mathcal{U}} \cap \left\{AR^{-1}\nu + A\hat{\theta} :\ \|\nu\|_2 \le \sqrt{\frac{\chi^2_{\text{rank}(V),1-\alpha}}{n}}\right\}.
\end{aligned}$$

Hence, by letting the estimated mean vector and covariance matrix based on the random sample be $\hat{\mu}$ and $\hat{\Sigma}$, respectively, we have

$$\bar{\mathcal{U}} = \hat{\mathcal{U}} \cap \left\{AR^{-1}\nu + \begin{pmatrix} \hat{\mu} \\ \text{svec}(\hat{\Sigma}) \end{pmatrix} :\ \|\nu\|_2 \le \sqrt{\frac{\chi^2_{\text{rank}(V),1-\alpha}}{n}}\right\}. \tag{4.17}$$

Remark 4.3 *If V is not invertible, then one can use a generalized inverse, such as the Moore-Penrose inverse.* □

Remark 4.4 *The construction of the uncertainty set can straightforwardly be extended to higher dimensions using suitable ϕ, A, and V. Details are omitted for brevity of exposition.* □

Now, consider a convex quadratic constraint

$$y^T \Sigma y + \mu^T y + c \leq 0, \tag{4.18}$$

where μ and Σ are the mean vector and covariance matrix of a random vector. By using the uncertainty set $\bar{\mathcal{U}}$ in (4.17) and Example 4.4, the SRC of (4.18) is

$$\begin{cases} \hat{\mu}^T y + \text{trace}\left(\hat{\Sigma} W\right) + \rho \left\| \left(AR^{-1}\right)^T \begin{pmatrix} y \\ \text{svec}(W) \end{pmatrix} \right\|_2 + c \leq 0, \\ \begin{bmatrix} W & y \\ y^T & 1 \end{bmatrix} \succeq 0_{n+1 \times n+1}, \end{cases} \tag{4.19}$$

where $\rho = \sqrt{\frac{\chi^2_{\text{rank}(V),1-\alpha}}{n}}$. If y and W satisfy (4.19), then y satisfies (4.18) for all $\binom{\mu}{\text{svec}(\Sigma)} \in \bar{\mathcal{U}}$, which asymptotically contains the population mean vector and covariance matrix with confidence level $(1-\alpha)\%$. This means that y satisfies (4.18) with *at least* confidence level $(1-\alpha)\%$ for the actual population mean vector and covariance matrix μ and Σ. Therefore, y that satisfies (4.19) is probably also immunized against some extra μ and Σ and hence conservative. In order to reduce conservativeness, we make use of the (asymptotic) chance constraint $Prob\left(y^T \Sigma y + \mu^T y + c \leq 0\right) \geq 1 - \alpha$, where $\alpha > 0$ is close to 0. Consider a solution $\bar{y}$ that satisfies (4.19) with a desired confidence level. $Prob\left(\bar{y}^T \Sigma \bar{y} + \mu^T \bar{y} + c \leq 0\right)$ is expected to be larger than the desired confidence level. If so, then by decreasing the confidence level that is used in the construction of the uncertainty set and considering $\tilde{y}$ that satisfies (4.19), $Prob\left(\tilde{y}^T \Sigma \tilde{y} + \mu^T \tilde{y} + c \leq 0\right)$ gets closer to the desired confidence level. This procedure of finding a suitable confidence level used in the uncertainty set for which the chance constraint has the desired confidence level may be time consuming. Therefore, we propose a reformulation of the chance constraint $Prob\left(y^T \Sigma y + \mu^T y + c \leq 0\right) \geq 1 - \alpha$.

For any vector β, (4.15) implies that

$$\sqrt{n}\left(\beta^T T_n - \beta^T \theta\right) \xrightarrow[n \to \infty]{d} N\left(0, \beta^T V \beta\right). \tag{4.20}$$

By setting $\beta = A\binom{y}{\text{svec}(yy^T)}$, it follows straightforwardly that the (asymptotic) chance constraint with probability of $1-\alpha$ is equivalent to

$$\frac{z_{1-\alpha}}{\sqrt{n}} \sqrt{\beta^T \hat{V} \beta} + \hat{\mu}^T y + y^T \hat{\Sigma} y + c \leq 0, \tag{4.21}$$

where $z_{1-\alpha}$ is the $1-\alpha$ percentile of the standard normal distribution. A solution that satisfies constraint (4.21) is less conservative than a solution that satisfies (4.19). However, due to the term $\text{svec}(yy^T)$ in β, $\sqrt{\beta^T \hat{V} \beta}$ is not convex and therefore the computational complexity of (4.21) is much higher than (4.19). So, we find a tractable relaxation of it.

Clearly (4.21) is equivalent to the set of constraints

$$\frac{z_{1-\alpha}}{\sqrt{n}}\|R^{-1}\beta\| + \hat{\mu}^T y + y^T\hat{\Sigma}y + c \leq 0, \quad \beta = A\begin{pmatrix} y \\ \text{svec}(W)\end{pmatrix}, \quad W = yy^T,$$

where R is the Cholesky factorization of $\hat{V}^{-1}$. The constraint $W = yy^T$ is nonconvex, so we relax it to $W \succeq yy^T$, which is a semi-definite representable constraint. Hence,

$$\begin{cases} \frac{z_{1-\alpha}}{\sqrt{n}}\|R^{-1}\beta\| + \hat{\mu}^T y + y^T\hat{\Sigma}y + c \leq 0, \\ \begin{bmatrix} W & y \\ y^T & 1\end{bmatrix} \succeq 0_{n+1\times n+1}, \quad \beta = A\begin{pmatrix} y \\ \text{svec}(W)\end{pmatrix}, \end{cases} \tag{4.22}$$

is a relaxation of (4.21). After solving the problem containing (4.22), it can be checked whether the constraint $W = yy^T$ is satisfied in the solution, which surprisingly happens to be the case in all our numerical results.

4.6 Applications

In this section, we apply the results of the previous sections to a robust portfolio choice, norm approximation, and regression line problem.

4.6.1 Mean-Variance portfolio problem

In this subsection, we describe a formulation for a mean-variance portfolio problem (Chapter 2 of [55]), and use the results of Section 4.5 to construct an uncertainty set and to derive a tractable reformulation of the robust counterpart.

Problem formulation: We consider a mean-variance portfolio problem with n assets. Let μ and Σ be the expectation and covariance matrix of the return vector $r = (r_1, ..., r_n)$, respectively. One formulation of a mean-variance portfolio problem is to model the trade-off between the risk and mean return in the objective function using a risk-aversion coefficient λ:

$$\max_{\omega}\left\{\mu^T\omega - \lambda\omega^T\Sigma\omega \;:\; \mathbb{1}^T\omega = 1, \;\; \omega \geq 0\right\}, \tag{4.23}$$

where $\mathbb{1} = [1, 1, ..., 1]^T$. The risk aversion coefficient is determined by the decision maker. When it is small, it means that the mean return is more important than the corresponding risk and it leads to a more risky portfolio than when the risk-aversion coefficient is large.

In practice μ and Σ are typically estimated from a set of historical data, which makes them sensitive to sampling inaccuracy. There are several ways of defining uncertainty sets for the expected return vector and asset return covariance matrix, e.g., see Chapter 12 of [55]. In this section, we use $\bar{\mathcal{U}}$ defined in (4.17), i.e., the uncertainty set constructed for $\left(\begin{smallmatrix}\mu\\ \mathrm{svec}(\Sigma)\end{smallmatrix}\right)$. Using (4.19), the robust counterpart of (4.23) with uncertainty set $\bar{\mathcal{U}}$ reads

$$\begin{aligned}\max_{\omega,W}\ & \hat{\mu}^T\omega-\lambda tr(\hat{\Sigma}W)-\rho\left\|\left(AR^{-1}\right)^T\begin{pmatrix}-\omega\\ \lambda\mathrm{svec}(W)\end{pmatrix}\right\|_2\\ \text{s.t.}\ & \begin{bmatrix}W & \omega\\ \omega^T & 1\end{bmatrix}\succeq 0_{n+1\times n+1},\ \ \mathbb{1}^T\omega=1,\ \ \omega\geq 0,\end{aligned}\tag{4.24}$$

where $\rho=\sqrt{\frac{\chi^2_{\mathrm{rank}(V),1-\alpha}}{n}}$, $\hat{\mu}$, $\hat{\Sigma}$, and $\hat{V}$ are consistent estimates of μ, Σ, and V, with V and A as in (4.15) and (4.16), respectively, but formulated for the higher dimensional case, and R is the Cholesky factorization of $\hat{V}^{-1}$.

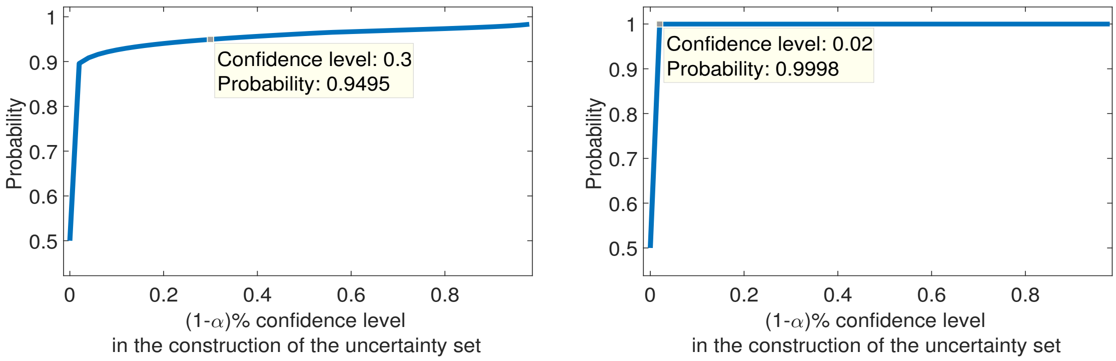
Furthermore, by setting $\beta=A\left(\begin{smallmatrix}\omega\\ -\lambda\mathrm{svec}(W)\end{smallmatrix}\right)$, and using the relaxed chance constraint (4.22), the robust counterpart of problem (4.23) with confidence $(1-\alpha)\%$ is approximated by

$$\begin{aligned}\max_{\omega}\ \ & \hat{\mu}^T\omega-\lambda\omega^T\hat{\Sigma}\omega-\frac{z_{1-\alpha}}{\sqrt{n}}\left\|R^{-1}\beta\right\|_2\\ \text{s.t.}\ \ & \begin{bmatrix}W & \omega\\ \omega^T & 1\end{bmatrix}\succeq 0_{n+1\times n+1},\ \ \mathbb{1}^T\omega=1,\ \ \omega\geq 0.\end{aligned}\tag{4.25}$$

Numerical evaluation: To evaluate the above robust counterparts, we use the monthly average value weighted return of 5 and 30 industries from 1956 until 2015, obtained from "Industry Portfolios" data on the website http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html. The data are monthly returns, but to present the results, we report the annualized returns (obtained by multiplying the expected monthly return by 12) and the annualized risk (multiplication of the standard deviation by $\sqrt{12}$). Furthermore, we set the risk aversion coefficient λ to 3.

We have solved the following three problems: (4.23) with nominal values for μ and Σ estimated from the data, which we call *Nominal problem*; (4.24), which we call *Robust problem*; and (4.25), which we call *Chance problem*, due to the chance constraint.

We first check the behavior of $Prob(\mu^T\omega^*-\lambda{\omega^*}^T\Sigma\omega^*\geq z^*)$ as a function of the confidence level used to construct the uncertainty set, where ω^* and z^* are the robust solution and corresponding robust objective value, respectively.



(a) 5 industries **(b)** 30 industries

Figure 4.1: The horizontal axis presents the value of $(1-\alpha)\%$, the confidence level used in the uncertainty set for data with 5 and 30 industries. The vertical axis presents the value of $Prob(\mu^T\omega^* - 3{\omega^*}^T\Sigma\omega^* \geq z^*)$, where ω^*, z^* are the robust solution and corresponding objective value for (4.24), respectively, with the uncertainty set (4.17) and different α. The plots are constructed by considering multiplications of 0.02 in $[0, 1]$ as values of α.

As shown in Figure 4.1, in order to be sure that the constraint $\mu^T\omega^* - \lambda{\omega^*}^T\Sigma\omega^* \geq z^*$ is satisfied with probability of at least 95%, one can reduce the confidence level used in the construction of the uncertainty set from 95% to 30% for the 5 industries case, and to 2% for the 30 industries case.

We considered both data sets with 5 and 30 industries in our numerical experiments; however, due to similarity in the results, we present the results of considering only data set with 30 industries.

After solving the *Nominal problem*, the *Robust problem* considering the uncertainty set with 95% confidence level, the *Robust problem* considering the uncertainty set with 2% confidence level, and the *Chance problem* with 95% confidence level, we compare the solutions in three ways:

(i) evaluating the solutions with respect to the nominal values;

(ii) evaluating the solutions with respect to their worst-case scenarios in the uncertainty set constructed with 95% confidence level;

(iii) evaluating the solutions with respect to their worst-case scenarios in the uncertainty set constructed with 2% confidence level.

Table 4.1 presents the evaluations of the solutions. In the first block row (with results), the evaluation is done using the nominal scenario. The objective value of the *Nominal problem* is the highest. The worst objective value in this row is corresponding to the solution of the *Robust problem* considering the uncertainty set with 95% confidence level. This solution is immunized against more scenarios than the others.

The second block row is the evaluation of the solutions considering their worst-case scenario in the uncertainty set constructed by 95% confidence level. This implies that the solution of the *Robust problem* with this uncertainty set has the highest objective value, because the solution is immunized against all scenarios in the uncertainty set; however, other solutions are immunized against all scenarios in a subset of the uncertainty set. The third block row has the same interpretation, where the scenario is chosen in the uncertainty set with confidence level 2%.

Table 4.1 shows that even though all solutions have close annualized returns and risks in the nominal scenario, the solutions of (4.24) have extremely better returns and risks in the included worst-case scenarios.

The solution of (4.25) is much worse than the robust solutions, because all solutions of (4.25) are immunized against 95% of possible scenarios and the worst-case scenarios lie in the other 5% part.

We emphasize that even though the confidence level of 2% seems to make the uncertainty set much smaller than the one corresponding to the 95%, this is not the case because for the 30 industries case we have $\sqrt{\frac{\chi^2_{\text{rank}(V),0.95}}{n}} = 0.8723$ and $\sqrt{\frac{\chi^2_{\text{rank}(V),0.02}}{n}} = 0.7751$.

4.6.2 Least-squares problems with uncertainties

This subsection contains applications of the results of Section 4.4 to two well-known problems, namely a norm approximation and a linear regression problem.

Norm approximation with uncertainty in the coefficients The norm approximation $\min_{y\in\mathbb{R}^n} \|Ay-b\|_2$ tries to find the closest point to $b \in \mathbb{R}^m$ in the range of the linear function Ay. The solution to this problem can be sensitive even to small errors in A or b. To detect this, one can analyze the condition number of the matrix A and check the sensitivity of the nominal solution to a perturbation in A, see, e.g., Chapter 7 of [71]. If the condition number is large, then the solution might be sensitive to a small error in A or b and not reliable. In this subsection we are using the results of Section 4.4 to deal with this problem.

Consider the norm approximation $\min_y \|(A+\Delta)y-b\|_2$, where $\Delta \in \mathcal{Z} \subseteq \mathbb{R}^{m\times n}$ reflects the uncertainty in A. This problem is equivalent to $\min_y y^T (A+\Delta)^T (A+\Delta) y + 2b^T (A+\Delta) y + b^T b$. Now using the results of Section 4.4, upper and lower bounds

		solution of *Nominal problem* (4.23)	solution of *Robust problem* (4.24) with confidence level 95%	solution of *Robust problem* (4.24) with confidence level 2%	solution of *Chance problem* (4.25)
Nominal case	Obj. value	**-35.57**	-38.10	-37.82	-35.59
	Ann. risk	*12.03*	*12.44*	*12.40*	*12.03*
	Ann. return	*7.02*	*7.41*	*7.36*	*7.00*
Worst-case with confidence level 95%	Obj. value	-57.24	**-48.65**	-48.67	-55.83
	Ann. risk	*14.00*	*13.12*	*13.10*	*13.82*
	Ann. return	*-99.03*	*-67.27*	*-69.20*	*-97.24*
Worst-case with confidence level 2%	Obj. value	-55.79	-47.70	**-47.70**	-54.44
	Ann. risk	*13.88*	*13.06*	*13.04*	*13.71*
	Ann. return	*-91.16*	*-60.80*	*-62.64*	*-89.46*

Table 4.1: Comparison among the solutions of the *nominal problem* (4.23), the *Robust problem* (4.24) considering the uncertainty set with 95% confidence level, the *Robust problem* (4.24) considering the uncertainty set with 2% confidence level, and (4.25) in three ways: The first block-row with results is the nominal evaluation of the solutions. The second and third block-rows are the evaluation of the solutions with respect to their worst-case scenarios in uncertainty sets 95%, and 2% confidence level, respectively. The results are by considering the data for 30 industries. The **bold** number in each block-row shows the best objective value corresponding to that scenario. The annualized return and risk are in *italics* and not individually optimized.

on the robust optimal value of this problem are obtained by solving

$$\begin{aligned} \min_{W,y} \;& \operatorname{trace}\Big((A^TA + \Omega^2 I_n)W\Big) + \delta^*_{\mathcal{Z}}(2WA^T - 2by^T) - 2b^TAy + \|b\|_2^2 \\ & \begin{bmatrix} W & y \\ y^T & 1 \end{bmatrix} \succeq 0_{n+1\times n+1}, \end{aligned} \tag{4.26}$$

and

$$\begin{aligned} \min_{W,y} \;& \operatorname{trace}\Big(A^TAW\Big) + \delta^*_{\mathcal{Z}}(2WA^T - 2by^T) - 2b^TAy + \|b\|_2^2 \\ & \begin{bmatrix} W & y \\ y^T & 1 \end{bmatrix} \succeq 0_{n+1\times n+1}, \end{aligned} \tag{4.27}$$

respectively. As an example, let

$$A = \begin{bmatrix} 16.0283 & 2.0422 & 3.0204 & 13.0173 \\ 5.0000 & 11.0271 & 10.0230 & 7.9977 \\ 8.9510 & 7.0000 & 5.9724 & 12.0124 \\ 4.0343 & 13.9878 & 15.0000 & 0.9736 \end{bmatrix}, \quad b = \begin{bmatrix} 34 \\ 34 \\ 34 \\ 34 \end{bmatrix}.$$

The matrix A is invertible. Hence, the nominal solution is $y^N = A^{-1}b$, which may be sensitive to a small perturbation in A, since the condition number of A, calculated by $\|A\|_{2,2}\|A^{-1}\|_{2,2}$, is 2.2×10^4, which is rather high.

We consider the uncertainty set $\mathcal{Z} = \{\Delta \in \mathbb{R}^{n\times n} : \ \|\Delta\|_\infty \leq \rho\}$, where $n = 4$. In our numerical experiments, we use $\Omega = n\rho$. This is the exact value of $\sup_{\Delta\in\mathcal{Z}} \|\Delta\|_{2,2}$, which is shown in the following proposition.

Proposition 4.2 *Let* $\mathcal{Z} = \{\Delta \in \mathbb{R}^{n\times n} : \ \|\Delta\|_\infty \leq \rho\}$. *Then,* $\sup_{\Delta\in\mathcal{Z}} \|\Delta\|_{2,2} = n\rho$.

Proof. Appendix 4.B.10. □

Let y^I and y^O be solutions to (4.26) and (4.27), respectively. Figure 4.2 presents the objective values of y^N, y^I, and y^O, for different scenarios and values of $\rho \in [0, 0.1]$.

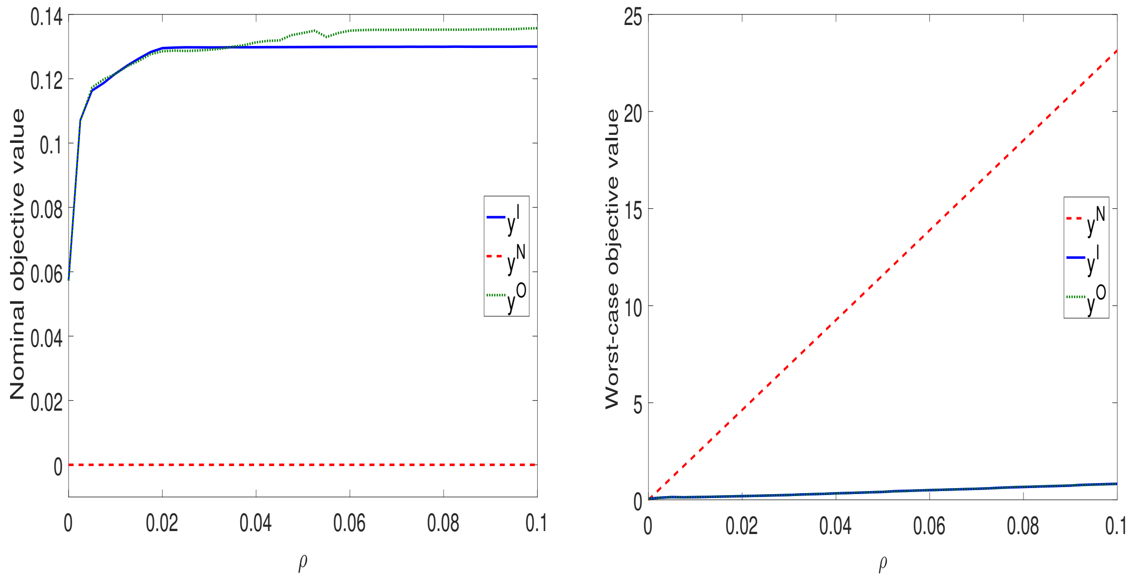
By checking all the corner points in the uncertainty set, we find that Assumption (C) holds for all $\rho \in [0, 0.1]$. Moreover, since A is invertible and $0_{n\times n}$ is in the relative interior of $\mathcal{Z}$, Assumption (D) holds. To find the worst-case scenario, we check all the corner points as well. As one can see in Figure 4.2, even though the nominal objective values of y^N, y^I, and y^O are not much different, the objective value of y^N in the worst-case scenario is significantly worse than the other two.

Robust linear regression with data inaccuracy Another application of the results of this chapter is finding a robust linear regression of a dependent variable Y and a vector of independent variables X that are highly collinear. For a data set with n linearly independent variables and m data points, a mathematical formulation of finding the regression line is

$$\begin{aligned} \min_{w,c,b} \quad & \|w\|_2\,, \\ \text{s.t.} \quad & w_i = \sum_{j=1}^{n} X_{ij}c_j + b - Y_i, \ \forall i = 1, ..., m, \end{aligned} \tag{4.28}$$

where X_{ij} is the i-th observed value of the j-th independent variable and Y_i is the value of the dependent variable in the i-th observation.

For our numerical experiment, we use the data presented in Table 7.1 of [101]. The data, consisting of twenty data points ($m = 20$), corresponds to studying the linear regression of the amount of body fat (Y) on three different possible independent



(a) **(b)**

Figure 4.2: Two different evaluations of y^I, y^O and y^N related to the norm approximation problem. (a) The nominal objective value is computed by $\|Ay - b\|_2$. (b) The worst-case objective value is computed by $\|(A + \Delta^*)y - b\|_2$, where Δ^* is the worst-case scenario corresponding to y^N, y^I or y^O. The solid blue, red dashed, and green dot curves correspond to y^I, y^N, and y^O, respectively. The objective worst-case evaluation of y^I and y^O are close, with the largest difference of 0.0145 that occurs for $\rho = 0.0525$.

variables $(X_1,\ X_2,\ X_3)$, namely triceps skinfold thickness, thigh circumference, and midarm circumference. For this data, as it is stated in Table 9.5 of the book [101], the maximum variance inflation factor, which qualifies the strength of the linear relation between the variables X_1, X_2, X_3, is 708.84 with a mean of 459.26, which implies that, for example, X_3 is strongly correlated to X_1 and X_2.

To reformulate (4.28) into the form (4.1b), let $B \in \mathbb{R}^{m\times(n+2)}$ be a matrix whose collection of the first $n = 3$ columns is the matrix X consisting of the data points corresponding to (X_1, X_2, X_3), the $(n + 1) = 4$th column is $[Y_i]_{i=1,\ldots,m}$, and the components of the last column are all ones. Then, problem (4.28) is equivalent to $\min_{y\in\mathbb{R}^{(n+2)}}\{\|By\|_2 : \ y_{n+1} = -1\}$. Solving this problem results in the nominal solution $y^N = [c^{N^T}, -1, b^N]^T = [4.334, -2.857, -2.186, -1, 117.082]^T$. The condition number of B is 1.26×10^4. This means that the nominal solution might be sensitive to an error in B. Let us assume that the maximal inaccuracy in the coefficients of the first $(n + 1)$ columns of B is 0.5%. Hence, we consider the following uncertainty

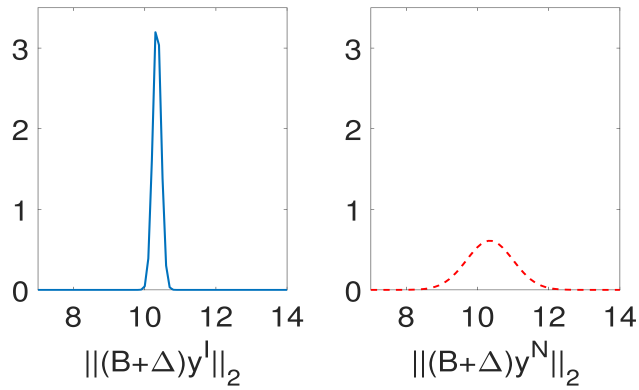


Figure 4.3: The distributions of $\left\|(B+\Delta)y^I\right\|_2$ and $\left\|(B+\Delta)y^N\right\|_2$ for 50,000 sample points Δ from the uncertainty set $\mathcal{Z}$.

set:

$$\mathcal{Z} = \{\Delta \in \mathbb{R}^{m\times(n+2)} : \ |\Delta_{ij}| \leq 0.005 \max_{i=1,\dots,m} |B_{ij}|, \ \Delta_{i(n+2)} = 0, i = 1,\dots,m, j = 1,\dots,n{+}1\}.$$

Similar to the proof of Proposition 4.2, $\Omega = 0.005\sqrt{(n+2)m}\|B\|_\infty$ is an upper bound on $\sup_{\Delta\in\mathcal{Z}} \|\Delta\|_{2,2}$. For this instance, Assumption (C) does not hold. Therefore, we only consider the inner approximation

$$\begin{aligned} \min_{\substack{W \in \mathbb{R}^{(n+2)\times(n+2)} \\ y \in \mathbb{R}^{(n+2)} \\ \eta \in \mathbb{R}}} \quad & \text{trace}\left((B^T B + \Omega^2 I_{n+2})W\right) + \delta^*_{\mathcal{Z}}(2BW) + \frac{\eta}{4} \\ & \begin{bmatrix} W & y \\ y^T & \eta \end{bmatrix} \succeq 0_{n+1\times n+1}, \quad y_{n+1} = -1. \end{aligned}$$

Solving the above inner approximation results in the solution $y^I = [c^{I^T}, -1, b^I]^T = [0.839, 0.131, -0.331, -1, 1.370]^T$. To evaluate y^I and y^N, we find their corresponding worst-case scenarios by using YALMIP and global solver BMIBNB. The nominal objective values of y^I and y^N are 10.331 and 9.920, and the worst-case objective values are 11.733 and 18.394, respectively. Therefore, the robust solution performs more efficiently than the nominal solution in the worst-case scenario, considering only 0.5% inaccuracy in the data.

To have a more elegant way of comparing the solutions, we derive the distributions of $\left\|(B+\Delta)y^I\right\|_2$ and $\left\|(B+\Delta)y^N\right\|_2$ when Δ is drawn uniformly from $\mathcal{Z}$ with 50,000 sample points. Figure 4.3 shows the probability density functions of $\left\|(B+\Delta)y^I\right\|_2$ and $\left\|(B+\Delta)y^N\right\|_2$. Here, $\left\|(B+\Delta)y^I\right\|_2$ has a mean and standard deviation of 10.3430 and 0.1166, respectively. For $\left\|(B+\Delta)y^N\right\|_2$, these numbers are 10.3440 and 0.6541, respectively. This shows that for this sample set, y^I and y^N have close means; however, the standard deviation of y^N is much higher than y^I.

As it was pointed out, X_1,X_2, and X_3 are highly correlated. By excluding X_3 from the data set, the performance of the solution to the inner approximation problem becomes similar to the performance of the nominal solution.

Appendix

We provide in this section all the proofs of the results of the chapter, but first we present one lemma from the literature.

4.A Essential lemma

Lemma 4.5 *(**Schur complement lemma**, see, e.g., Appendix A.5.5 of [44]) A symmetric block matrix* $A = \begin{bmatrix} P & Q^T \\ Q & R \end{bmatrix}$*, where* $P \in S_n$*,* $Q \in \mathbb{R}^{m \times n}$*, and* $R \succ 0_{m\times m}$*, is positive (semi-) definite if and only if the matrix* $P - Q^T R^{-1} Q$ *is positive (semi-) definite.*

4.B Proofs

4.B.1 Proof of Theorem 4.1

To prove this theorem we use the same line of reasoning as in Theorem 2 in [22]. For any $\Delta \in \mathcal{Z}$, it is clear that $\frac{A(\Delta)+A(\Delta)^T}{2} \succeq 0_{n\times n}$ due to positive semi-definiteness of $A(\Delta)$. Also, $y^T A(\Delta) y = y^T \frac{A(\Delta)+A(\Delta)^T}{2} y$ for any $y \in \mathbb{R}^n$, and $\Delta \in \mathcal{Z}$. We substitute $y^T A(\Delta) y$ by $y^T \frac{A(\Delta)+A(\Delta)^T}{2} y$ in constraints (4.1).
(I) Let $\mathcal{U} = \left\{ \left(\frac{A(\Delta)+A(\Delta)^T}{2}, b(\Delta) \right) : \ \Delta \in \mathcal{Z} \right\}$. It is clear that $y \in \mathbb{R}^n$ satisfies (4.1a) if and only if $F(y) := \max_{(B,b)\in\mathcal{U}} \left\{ y^T B y + b^T y + c \right\} \leq 0$. Setting

$$\delta_{\mathcal{U}}(B,b) = \begin{cases} 0 & \text{if } (B,b) \in \mathcal{U}, \\ +\infty & \text{otherwise,} \end{cases}$$

we have

$$F(y) = \max_{\substack{B \succeq 0_{n\times n} \\ b \in \mathbb{R}^n}} \left\{ y^T B y + b^T y + c - \delta_{\mathcal{U}}(B,b) \right\}.$$

Since $B \succeq 0_{n\times n}$, for all $B \in \mathcal{U}$ and there exists a positive definite B in the relative interior of $\mathcal{U}$, by Theorem 6.1 in [22], $F(y) \leq 0$ is equivalent to the existence of

$W \in \mathbb{R}^{n\times n}$ and $u \in \mathbb{R}^n$, such that

$$\delta^*_{\mathcal{U}}(W,u) - \inf_{\substack{A \succeq 0_{n\times n}\\ b\in\mathbb{R}^n}} \left\{\text{trace}\left(AW^T\right) + u^Tb - \left(y^TAy + b^Ty + c\right)\right\} \le 0, \tag{4.29}$$

where $\delta^*_{\mathcal{U}}(.)$ is the support function of the set $\mathcal{U}$. It follows from the definition of the support function that

$$\begin{aligned}
\delta^*_{\mathcal{U}}(W,u) &= \sup_{(B,b)\in\mathcal{U}} \text{trace}\left(BW^T\right) + u^Tb\\
&= \sup_{\Delta\in\mathcal{Z}} \text{trace}\left(\frac{A(\Delta)+A(\Delta)^T}{2}W^T\right) + u^Tb(\Delta)\\
&= \sup_{\Delta\in\mathcal{Z}} \text{trace}\left((A+\Delta)\left(\frac{W+W^T}{2}\right)\right) + u^T(b+\Delta a)\\
&= \text{trace}\left(A\left(\frac{W+W^T}{2}\right)\right) + u^Tb + \sup_{\Delta\in\mathcal{Z}} \text{trace}\left(\Delta\left(\frac{W+W^T}{2}\right)\right) + u^T\Delta a\\
&= \text{trace}\left(A\left(\frac{W+W^T}{2}\right)\right) + u^Tb + \sup_{\Delta\in\mathcal{Z}} \text{trace}\left(\Delta\left(\left(\frac{W+W^T}{2}\right) + au^T\right)\right)\\
&= \text{trace}\left(A\left(\frac{W+W^T}{2}\right)\right) + u^Tb + \delta^*_{\mathcal{Z}}\left(\left(\frac{W+W^T}{2}\right) + ua^T\right).
\end{aligned} \tag{4.30}$$

Also, we have

$$\begin{aligned}
&\inf_{\substack{A \succeq 0_{n\times n}\\ b\in\mathbb{R}^n}} \left\{\text{trace}\left(AW^T\right) + u^Tb - \left(y^TAy + b^Ty + c\right)\right\}\\
=&\inf_{\substack{A \succeq 0_{n\times n}\\ b\in\mathbb{R}^n}} \left\{\text{trace}\left(AW\right) + u^Tb - \left(y^TAy + b^Ty + c\right)\right\}\\
=&-c + \inf_{\substack{A \succeq 0_{n\times n}\\ b\in\mathbb{R}^n}} \left\{\text{trace}\left(A\left(W - yy^T\right)\right) + b^T(u-y)\right\}\\
=&\begin{cases} -c & W - yy^T \succeq 0_{n\times n},\ u = y,\\ -\infty & \text{otherwise.}\end{cases}
\end{aligned} \tag{4.31}$$

So, the fact that $W \succeq 0_{n\times n}$ implies $\frac{W+W^T}{2} = W$, together with Lemma 4.5, (4.30), and (4.31) result in (4.3).

(II) Similar to the proof of part (I) we have $y \in \mathbb{R}^n$ satisfies (4.1b) if and only if there exists $W \in \mathbb{R}^{n\times n}$ such that

$$\delta^*_{\mathcal{U}}(W,u) - \inf_{\substack{A \succeq 0_{n\times n}\\ b\in\mathbb{R}^n}} \left\{\text{trace}\left(AW^T\right) + u^Tb - \left(\sqrt{y^TAy} + b^Ty + c\right)\right\} \le 0. \tag{4.32}$$

Analogous to the result in Section 3.4 of [65],

$$\inf_{\substack{A \succeq 0_{n\times n}\\ b\in\mathbb{R}^n}} \left\{\text{trace}\left(AW^T\right) + u^Tb - \left(\sqrt{y^TAy} + b^Ty + c\right)\right\}$$

$$= \quad -c - \inf_{\eta} \left\{ \frac{\eta}{4} : \ u = y, \ \begin{bmatrix} W & y \\ y^T & \eta \end{bmatrix} \succeq 0_{n+1 \times n+1} \right\}. \tag{4.33}$$

So, (4.32) is equivalent to

$$\delta^*_{\mathcal{U}}(W, u) + c + \inf_{\eta \in \mathbb{R}} \left\{ \frac{\eta}{4} : \ \begin{bmatrix} W & y \\ y^T & \eta \end{bmatrix} \succeq 0_{n+1 \times n+1} \right\} \leq 0. \tag{4.34}$$

In (4.34), the infimum is taken over a closed lower bounded set, since $\eta \geq 0$. Hence, $W \in \mathbb{R}^{n \times n}$ and $y \in \mathbb{R}^n$ satisfies (4.34) if and only if there exists $\eta \in \mathbb{R}$ such that

$$\begin{cases} \operatorname{trace}\left(WA^T\right) + b^T y + \delta^*_{\mathcal{Z}}(W + ya^T) + c + \dfrac{\eta}{4} \leq 0, \\ \begin{bmatrix} W & y \\ y^T & \eta \end{bmatrix} \succeq 0_{n+1 \times n+1}, \end{cases}$$

which completes the proof. $\square$

4.B.2 Proof of Lemma 4.2

(i)

$$\begin{aligned} \delta^*_{\mathcal{Z}}(U) &= \sup_{\Delta \in \mathbb{R}^{n \times n}} \left\{ \operatorname{trace}\left(\Delta U^T\right) : \ \operatorname{vec}(\Delta) \in \mathcal{U} \right\} \\ &= \sup_{\Delta \in \mathbb{R}^{n \times n}} \left\{ \operatorname{vec}(U)^T \operatorname{vec}(\Delta) : \ \operatorname{vec}(\Delta) \in \mathcal{U} \right\} = \delta^*_{\mathcal{U}}(\operatorname{vec}(U)). \end{aligned}$$

(ii)

$$\begin{aligned} \delta^*_{\mathcal{Z}}(U) &= \sup_{\Delta \in \mathcal{Z}} \left\{ \operatorname{trace}\left(\Delta U^T\right) \right\} = \sup_{\zeta \in \mathcal{U}} \left\{ \sum_{i=1}^k \operatorname{trace}\left(\zeta_i \Delta^i U^T\right) \right\} \\ &= \sup_{\zeta \in \mathcal{U}} \left\{ \zeta^T \left[\operatorname{trace}\left(\Delta^i U^T\right) \right]_{i=1,\dots,k} \right\} = \delta^*_{\mathcal{U}} \left(\left[\operatorname{trace}\left(\Delta^i U^T\right) \right]_{i=1,\dots,k} \right). \end{aligned}$$

(iii)

$$\begin{aligned} \delta^*_{\mathcal{Z}}(U) &= \sup_{\Delta \in \mathcal{Z}} \left\{ \operatorname{trace}\left(\Delta U^T\right) \right\} = \sup_{\Theta \in \mathcal{U}} \left\{ \operatorname{trace}\left(L \Theta R U^T\right) \right\} \\ &= \sup_{\Theta \in \mathcal{U}} \left\{ \operatorname{trace}\left(\Theta R U^T L\right) \right\} = \delta^*_{\mathcal{U}}(L^T U R^T). \end{aligned}$$

(iv)

$$\begin{aligned} \delta^*_{\mathcal{Z}}(U) &= \sup_{\Delta \in \mathcal{Z}} \operatorname{trace}\left(\Delta U^T\right) = \sup_{\substack{\Delta_i \in \mathcal{Z}^i \\ i=1,\dots,k}} \sum_{i=1}^k \operatorname{trace}\left(\Delta^i U^T\right) \\ &= \sum_{i=1}^k \sup_{\Delta^i \in \mathcal{Z}_i} \operatorname{trace}\left(\Delta^i U^T\right) = \sum_{i=1}^k \delta^*_{\mathcal{Z}_i}(U). \end{aligned}$$

(v) Similar to the proof of Lemma 9 in [22].
(vi)

$$\begin{aligned}
&\delta^*_{\mathcal{Z}}\left((U_1,...,U_k)\right) = \sup_{\Delta\in\mathcal{Z}} \text{trace}\left(\Delta(U_1,...,U_k)^T\right) \\
&= \sup_{\substack{\Delta_i\in\mathcal{Z}_i\\ i=1,...,k}} \text{trace}\left((\Delta_1,...,\Delta_k)(U_1,...,U_k)^T\right) = \sup_{\substack{\Delta_i\in\mathcal{Z}_i\\ i=1,...,k}} \text{trace}\left(\sum_{i=1}^k \Delta_i U_i^T\right) \\
&= \sum_{i=1}^k \sup_{\Delta_i\in\mathcal{Z}_i} \text{trace}\left(\Delta_i U_i^T\right) = \sum_{i=1}^k \delta^*_{\mathcal{Z}_i}(U_i).
\end{aligned}$$

(vii)

$$\begin{aligned}
&\delta^*_{\mathcal{Z}}(U) = \sup_{\Delta\in\mathcal{Z}} \text{trace}\left(\Delta U^T\right) = \sup_{\substack{\Delta^i\,\in\,\mathcal{Z}_i\\ \lambda_i\geq 0\\ i\,=\,1,...,k}} \left\{\sum_{i=1}^k \lambda_i \text{trace}\left(\Delta^i U^T\right) : \sum_{i=1}^k \lambda_i = 1\right\} \\
&= \max_{i=1,...,k} \sup_{\Delta^i\in\mathcal{Z}_i} \text{trace}\left(\Delta^i U^T\right) = \max_{i=1,...,k} \delta^*_{\mathcal{Z}_i}(U).
\end{aligned}$$

$\square$

4.B.3 Proof of Lemma 4.3(b)

The assumptions imply that

$$\begin{aligned}
\delta^*_{\mathcal{Z}}(U) &= \sup_{\Delta}\left\{\text{trace}\left(\Delta U^T\right):\ \Delta^l \preceq \Delta \preceq \Delta^u\right\} \\
&= \max_{\Delta}\left\{\text{trace}\left(\frac{U+U^T}{2}\Delta\right):\ \Delta^l \preceq \Delta \preceq \Delta^u\right\} \\
&= \min_{\Lambda_1,\Lambda_2}\left\{\text{trace}\left(\Delta^u\Lambda_2\right) - \text{trace}\left(\Delta^l\Lambda_1\right):\ \Lambda_2-\Lambda_1 = \frac{U+U^T}{2},\ \Lambda_1,\Lambda_2 \succeq 0_{n\times n}\right\},
\end{aligned}$$

where the last equality holds because of conic duality theorem (Theorem 2.3) (both problems are strictly feasible). $\square$

4.B.4 Proof of Lemma 4.4(ii)

Lemma 4.1(d) implies that $\|U\|^2_{2,2}$ is the largest eigenvalue of UU^T. Hence, $\|U\|^2_{2,2} \leq \rho^2$ can be reformulated as $UU^T \preceq \rho^2 I_n$, which by using Lemma 4.5 is equivalent to

$$\begin{bmatrix}\rho^2 I_n & U\\ U^T & I_n\end{bmatrix} \succeq 0_{2n\times 2n}.$$

$\square$

4.B.5 Proof of the statement in Example 4.3

$y \in \mathbb{R}^n$ satisfies (4.7) if and only if

$$y^T Ay + \sup_{\zeta \in \mathcal{Z}} \left\{ \zeta^T \left[y^T A^i y + b^{i^T} y \right]_{i=1,\dots,t} \right\} + b^T y + c \leq 0. \tag{4.35}$$

Now, we show that $y \in \mathbb{R}^n$ satisfies (4.35) if and only if there exists $v \in \mathbb{R}^t$ such that

$$y^T Ay + \sup_{\zeta \in \mathcal{Z}} \left\{ \zeta^T v \right\} + b^T y + c \leq 0,\ v \geq \left[y^T A^i y + b^{i^T} y \right]_{i=1,\dots,t}. \tag{4.36}$$

It is clear that if $y \in \mathbb{R}^n$ and $v \in \mathbb{R}^t$ satisfy (4.36) then due to nonnegativity of $\zeta \in \mathcal{Z}$,

$$\zeta^T v \geq \zeta^T \left[y^T A^i y + b^{i^T} y \right]_{i=1,\dots,t},$$

which implies $y \in \mathbb{R}^n$ satisfies (4.35). Now let $y \in \mathbb{R}^n$ satisfies (4.35). Then setting $v = \left[y^T A^i y + b^{i^T} y \right]_{i=1,\dots,t}$, $y \in \mathbb{R}^n$ and $v \in \mathbb{R}^t$ satisfy (4.36). Now clearly (4.36) can be reformulated as (4.8). □

4.B.6 Proof of Theorem 4.2

(I) $y \in \mathbb{R}^n$ satisfies (4.2a) if and only if

$$y^T A^T Ay + 2y^T A^T \Delta y + \|\Delta y\|_2^2 + (D\Delta a)^T y + b^T y + c \leq 0, \quad \forall \Delta \in \mathcal{Z}. \tag{4.37}$$

Replacing $\|\Delta y\|_2^2$ by its upper bound $\Omega^2 \|y\|_2^2$ implies that $y \in \mathbb{R}^n$ satisfies (4.37) if it satisfies

$$y^T A^T Ay + 2y^T A^T \Delta y + \Omega^2 y^T y + (D\Delta a)^T y + b^T y + c \leq 0, \quad \forall \Delta \in \mathcal{Z}. \tag{4.38}$$

Setting $\mathcal{U} = \{(A^T A + 2A^T \Delta + \Omega^2 I_n, D\Delta a) : \ \Delta \in \mathcal{Z}\}$, (4.38) is equivalent to

$$y^T By + (b + d)^T y + c \leq 0 \quad \forall (B, d) \in \mathcal{U}. \tag{4.39}$$

For any $(B, d) \in \mathcal{U}$, B is positive semidefinite since

$$B = (A + \Delta)^T (A + \Delta) + \Omega^2 I_n - \Delta^T \Delta \succeq 0_{n \times n}.$$

So, by applying Theorem 4.1(I) and Lemma 4.2(iii), $y \in \mathbb{R}^n$ satisfies (4.37) if there exists $W \in \mathbb{R}^{n \times n}$ such that y and W satisfy (4.11).
(II) The proof is similar to part (I). □

4.B.7 Proof of Theorem 4.3

(I) It is clear that y satisfies (4.2a) if and only if y satisfies (4.37). Replacing $\|\Delta y\|_2^2$ with its lower bound 0 implies that if $y \in \mathbb{R}^n$ satisfies (4.37) then

$$y^T A^T Ay + 2y^T A^T \Delta y + (D\Delta a)^T y + b^T y + c \leq 0, \quad \forall \Delta \in \mathcal{Z}. \tag{4.40}$$

Setting $\mathcal{U} = \{(A^T A + 2A^T \Delta, D\Delta a) : \ \Delta \in \mathcal{Z}\}$, and using Theorem 4.1(I) and Lemma 4.2(iii) completes the proof.
(II) The proof is similar to the previous part. □

4.B.8 Proof of Theorem 4.4

(I) Let $y \in \mathbb{R}^n$ and $W \in \mathbb{R}^{n\times n}$ satisfy (4.13). Then, y satisfies (4.40). Therefore,

$$\max_{\Delta \in \mathcal{Z}} \{y^T A^T Ay + 2y^T A^T \Delta y + (D\Delta a)^T y + b^T y + c\} \leq 0. \tag{4.41}$$

As it is mentioned in the proof of Theorem 4.2(I), (4.2a) is equivalent to (4.37). Therefore, we have

$$\begin{aligned}
&\max_{\Delta \in \mathcal{Z}} \{y^T A^T Ay + 2y^T A^T \Delta y + \|\Delta y\|_2^2 + (D\Delta a)^T y + b^T y + c\} \\
&\leq y^T A^T Ay + b^T y + c + \max_{\Delta \in \mathcal{Z}} \{2y^T A^T \Delta y + (D\Delta a)^T y\} + \max_{\Delta \in \mathcal{Z}} \|\Delta y\|_2^2 \\
&\leq y^T A^T Ay + b^T y + c + \max_{\Delta \in \mathcal{Z}} \{2y^T A^T \Delta y + (D\Delta a)^T y\} + \Omega^2 \|y\|_2^2 \\
&\leq \Omega^2 \|y\|_2^2,
\end{aligned}$$

where the last inequality follows from (4.41).
(II) It is clear that (4.2b) is equivalent to

$$\sqrt{y^T A^T Ay + 2y^T A^T \Delta y + \|\Delta y\|_2^2 + (D\Delta a)^T y} + b^T y + c \leq 0, \quad \forall \Delta \in \mathcal{Z}.$$

Similar to the previous part, if y comes from the outer approximation (4.14), then we have

$$\begin{aligned}
&\sqrt{\max_{\Delta \in \mathcal{Z}} \{y^T A^T Ay + 2y^T A^T \Delta y + \|\Delta y\|_2^2 + (D\Delta a)^T y\}} + b^T y + c \\
&\leq \sqrt{y^T A^T Ay + \max_{\Delta \in \mathcal{Z}} \{2y^T A^T \Delta y + (D\Delta a)^T y\}} + \max_{\Delta \in \mathcal{Z}} \|\Delta y\|_2 + b^T y + c \\
&\leq \sqrt{y^T A^T Ay + \max_{\Delta \in \mathcal{Z}} \{2y^T A^T \Delta y + (D\Delta a)^T y\}} + \Omega \|y\|_2 + b^T y + c \\
&\leq \Omega \|y\|_2,
\end{aligned}$$

where the first inequality holds because of the fact that $\sqrt{f+g} \leq \sqrt{f} + \sqrt{g}$ for any $f, g \geq 0$.

□

4.B.9 Proof of Proposition 4.1

$(i) \Leftrightarrow (ii)$: This equivalence holds, because, in a general case, $A \in \mathbb{R}^{n\times n}$ is positive semi-definite if and only $\frac{A+A^T}{2} \succeq 0_{n\times n}$.
$(ii) \Leftrightarrow (iii)$: We first show that for a given $A \in S_n$, we have $A \succeq 0_{n\times n}$ if and only if $\|A\|_\Sigma - \mathrm{trace}\,(A) = 0$. Since A is symmetric, all of its eigenvalues are real (Theorem 7.4 in [14]). Let $\lambda_1, ..., \lambda_r$ be the nonzero eigenvalues of A, where $r = \mathrm{rank}(A)$. So, $|\lambda_i| = \sigma_i(A)$, $i = 1, ..., r$. We know $A \succeq 0_{n\times n}$ if and only if all of the eigenvalues are nonnegative. This means that $A \succeq 0_{n\times n}$ if and only if $\lambda_i = \sigma_i(A)$, $i = 1, ..., r$. Therefore, if $A \succeq 0_{n\times n}$, then

$$\|A\|_\Sigma = \sum_{i=1}^{r} \sigma_i(A) = \sum_{i=1}^{r} \lambda_i = \mathrm{trace}\,(A)\,.$$

For the other direction, let $\|A\|_\Sigma - \mathrm{trace}\,(A) = 0$. This implies that $\sum_{i=1}^{r} (|\lambda_i| - \lambda_i)$ is zero. Therefore, all of the eigenvalues are nonnegative and hence $A \succeq 0_{n\times n}$.
Now since $\|A\|_\Sigma - \mathrm{trace}\,(A)$ cannot be negative, $\|A\|_\Sigma - \mathrm{trace}\,(A) = 0$ is equivalent to $\|A\|_\Sigma - \mathrm{trace}\,(A) \leq 0$. □

4.B.10 Proof of Proposition 4.2

$$\begin{aligned}
\sup_{\Delta\in\mathcal{Z}} \|\Delta\|_{2,2} &= \sup_{\substack{\Delta:\\ |\Delta_{ij}| \leq \rho\\ i,j=1,...,n.}} \|\Delta\|_{2,2} = \sup_{\substack{\Delta:\\ |\Delta_{ij}| \leq \rho\\ i,j=1,...,n.}} \sup_{x:\|x\|_2=1} \|\Delta x\|_2 \\
&= \sup_{x:\|x\|_2=1} \sup_{\substack{\Delta:\\ |\Delta_{ij}| \leq \rho\\ i,j=1,...,n.}} \|\Delta x\|_2 = \sup_{x:\|x\|_2=1} n\rho \,\|x\|_2 = n\rho.
\end{aligned}$$
□

CHAPTER 5

When are static and adjustable robust optimization problems with constraint-wise uncertainty equivalent?

5.1 Introduction

Many real-life optimization problems have parameters that are not exact. One way to deal with parameter uncertainty is Static Robust Optimization (SRO), which enforces the constraints to hold for all uncertain parameter values in a user-specified uncertainty region. In SRO, all decision variables represent “here and now” decisions, which means they should obtain specific numerical values as a result of the problem being solved before the actual uncertain parameter values “reveal themselves”.

An extension to SRO is Adjustable Robust Optimization (ARO) introduced in [25]. In ARO, some of the decision variables are “here and now”, while others represent “wait and see” decisions, which are assigned numerical values once some of the uncertain parameters have become known.

The advantage of using ARO lies in the fact that its worst-case objective value is no worse, and indeed usually better, than the corresponding static SRO. In [27] the authors prove for linear problems with linear uncertainty and convex uncertainty set that if the uncertainty is constraint-wise, and under a few more assumptions, SRO and ARO have the same optimal objective value. It is shown in [35] that the same result holds even for specific non-constraint-wise uncertainty. The conservativeness of the SRO solution for the ARO problem is studied for some classes of problems in [34] and [35].

Solving an ARO problem can be intractable even for linear cases [57]. There are accordingly many methods in use for finding a good approximation for an ARO problem. Using affine decision rules, [25], for “wait and see” variables appears to be effective for many ARO problems. For linear ARO problems with fixed recourse, using affine decision rules leads to a robust linear problem that is computationally tractable for many types of uncertainty sets. This is not the case for problems with non-fixed recourse.

An important line of research in ARO is finding classes of problems for which the affine decision rules are optimal. In [33], the authors prove that affine decision rules are optimal for linear ARO problems with right-hand side uncertainty and simplex uncertainty sets. A similar result is proven in [37] for ARO problems with a specific objective function that is convex in the uncertain parameters and adjustable variables, box constraints for the variables and a box uncertainty set. Also, in [74], the optimality of the affine decision rules is proven for unconstrained multi-stage ARO problems under some structural assumptions on the uncertainty set and objective function.

In [32], a bound is derived for the gap between the objective value of the problem that results from using affine decision rules and that of the ARO problem.

Although substituting "wait and see" decision variables with affine functions would appear to be highly effective, the method needs introducing many new variables. This is because for a problem with n adjustable variables and l uncertain parameters, applying affine decision rules means substituting n adjustable variables with $n(l+1)$ non-adjustable variables.

The contribution of this chapter is twofold:

1. We prove for a class of problems containing convex problems with concave uncertainty, which also satisfy a set of other conditions, that the objective values of the corresponding SRO and ARO problems are equal. This is an extension of the result in [27, Proposition 2.1], which is only for problems that are linear in the variables and uncertain parameters.

2. We study uncertain nonlinear problems in which some of the uncertain parameters are constraint-wise and the rest are not. In particular, we prove that for an ARO problem, under a set of conditions similar to the pure constraint-wise cases, there is an optimal decision rule that depends only on the non-constraint-wise uncertain parameters. Moreover, we show that for a specific class of problems, there is an affine decision rule that is only a function of the non-constraint-wise uncertain parameters and that yields the same objective value as using an affine decision rule that is a function of all uncertain parameters.

The first contribution means that for this class of problems, there is no need to solve ARO ones. This has two outstanding merits: first, solving an SRO problem is computationally much easier than solving an ARO one; and second, since ARO is an online approach, parts of the solution for a problem can only be implemented once the values of the uncertain parameters are known. The SRO approach is an offline one, however, so all preparations for implementing the solution can start immediately upon solving the SRO problem (for further discussion about online and offline approaches see [95]).

The merit of the second contribution is that it reduces the number of variables in the approximation problem that is using affine decision rules, since we know beforehand that the coefficients of the constraint-wise uncertain parameters are zero.

In the last part of the chapter, we apply our theoretical results to important classes of problems. We show that our contributions are applicable to convex quadratic and/or conic quadratic problems, which can arise in multi-stage portfolio optimization, for example. Moreover, for the facility location problem, we discuss two formulations that are equivalent in the deterministic case, and show, by using the first contribution, that the robust optimal value of one is better than the other. Also, we show that one can apply this contribution to an inventory system problem with demand and cost uncertainty to approximate the ARO problem by using affine decision rules and reach to a tractable formulation. Besides, for a specific class of two-stage LO problems we show that a part of the results in [35, Section 4] can be derived easily using the second contribution of this chapter.

We emphasize that the results obtained in this chapter concern the worst-case objective value of an ARO problem. We provide conditions under which the optimal SRO solutions are also optimal for the ARO problem. However, in such cases, another ARO optimal solution may yield a better average-case objective value [75].

The rest of the chapter is organized as follows: Section 2 presents our main results. We provide sets of conditions under which constraint-wise SRO and ARO problems have the same optimal objective values. Moreover, for problems in which just some of the uncertain parameters are constraint-wise and not all, we show that under similar sets of conditions, there is an optimal decision rule that is independent of the constraint-wise uncertain parameters. In Section 3, we apply our results to convex quadratic and conic quadratic problems, inventory system problems, and a specific class of two-stage LO problems.

5.2 Main results

In this section, we derive the main results presented in the chapter. The section starts by introducing some definitions and preliminaries in Section 5.2.1. In Section 5.2.2, we provide sets of conditions for problems with constraint-wise uncertainty under which adjustable and static robust optimization produces the same optimal values. In Section 5.2.3 we study problems in which only some of the uncertain parameters are constraint-wise and the rest are not.

5.2.1 Preliminaries

Consider the following uncertain nonlinear optimization problem

$$\begin{aligned} &\inf_{x\in\mathcal{X}} \inf_{y\in\mathcal{Y}(x)} f(\zeta,x,y) \\ &\qquad \text{s.t.} \quad g_i(\zeta,x,y)\le 0, \qquad i=1,...,m, \end{aligned} \tag{5.1}$$

where $\zeta \in \mathcal{Z} \subseteq \mathbb{R}^l$ is an uncertain parameter and $\mathcal{Z}$ is a nonempty uncertainty set, $x \in \mathcal{X} \subseteq \mathbb{R}^r$ is a non-adjustable variable, and $\mathcal{X}$ is a nonempty set defined by constraints that depend only on x, $y \in \mathcal{Y}(x) \subseteq \mathbb{R}^n$ is an adjustable variable and $\mathcal{Y}(x)$ is defined by constraints independent of ζ. Also, we assume that $f(\zeta,x,y)$ and $g_i(\zeta,x,y)$, $i = 1,...,m$, are continuous.

We can define static and adjustable robust optimization problems corresponding to uncertain problem (5.1).

Definition 5.1 ***(Static Robust Optimization)*** *For problem (5.1), the SRC is defined by*

$$\begin{aligned} (SRC) \quad &\inf_{x\in\mathcal{X}} \inf_{y\in\mathcal{Y}(x),\ t} \quad t \\ &\qquad s.t.\ \ f(\zeta,x,y)\le t \qquad \forall\zeta\in\mathcal{Z}, \\ &\qquad\qquad g_i(\zeta,x,y)\le 0, \quad \forall\zeta\in\mathcal{Z},\ i=1,...,m. \end{aligned}$$ □

Definition 5.2 ***(Adjustable Robust Optimization)*** *For problem (5.1), there are two different definitions for the adjustable robust counterpart (ARC):*

$$\inf\left\{ t \;\middle|\; \exists x\in\mathcal{X}\ \ \forall\zeta\in\mathcal{Z}\ \ \exists y\in\mathcal{Y}(x): \begin{array}{l} f(\zeta,x,y)\le t, \\ g_i(\zeta,x,y)\le 0, \quad i=1,...,m, \end{array} \right\} \tag{5.2}$$

and

$$\begin{aligned} (ARC) \quad &\inf_{x\in\mathcal{X}} \sup_{\zeta\in\mathcal{Z}} \inf_{\substack{y(\zeta)\in\mathcal{Y}(x)\\ t(\zeta)}} \quad t(\zeta) \\ &\qquad s.t. \qquad f(\zeta,x,y(\zeta))\le t(\zeta), \\ &\qquad\qquad\quad g_i(\zeta,x,y(\zeta))\le 0, \qquad i=1,...,m. \end{aligned}$$ □

The equivalence of problems (5.2) and (ARC) is proved in [114]. We denote the objective values of problems (SRC) and (ARC) by $Opt(SRC)$ and $Opt(ARC)$, respectively.

We extend the definition of (ARC) with fixed recourse for a linear problem with linear uncertainty in [25] to the nonlinear case (nonlinear problem with nonlinear uncertainty) in the following definition.

Definition 5.3 ***(Fixed Recourse Problem)*** (ARC) *has fixed recourse when there are continuous functions* $\tilde{f}, \tilde{g}_i : \mathbb{R}^{n+r} \to \mathbb{R}$, $\bar{f}, \bar{g}_i : \mathbb{R}^{l+r} \to \mathbb{R}$, *for* $i = 1, ..., m$, *such that for all* $\zeta \in \mathcal{Z} \subset \mathbb{R}^l$, $x \in \mathcal{X} \subset \mathbb{R}^r$, *and* $y \in \mathcal{Y}(x) \subset \mathbb{R}^n$,

$$\begin{aligned} f(\zeta, x, y) = & \quad \tilde{f}(x, y) + \bar{f}(\zeta, x), \\ g_i(\zeta, x, y) = & \quad \tilde{g}_i(x, y) + \bar{g}_i(\zeta, x), \quad i = 1, ..., m. \end{aligned}$$ □

In this chapter, we work primarily with constraint-wise uncertainty, which is defined as follows.

Definition 5.4 ***(Constraint-wise Uncertainty*** *[25]) For problem (5.1), the uncertainty is constraint-wise when each uncertain parameter* ζ *can be split into blocks* $\zeta = [\zeta_0, ..., \zeta_m] \in \mathbb{R}^l$ *such that the data of the objective depends only on* $\zeta_0 \in \mathbb{R}^{l_0}$, *the data of the i-th constraint depends solely on* $\zeta_i \in \mathbb{R}^{l_i}$, *and the uncertainty set* $\mathcal{Z} = \mathcal{Z}_0 \times \mathcal{Z}_1 \times ... \times \mathcal{Z}_m$, *where* $\mathcal{Z}_i \subseteq \mathbb{R}^{l_i}$ *is the uncertainty region for* ζ_i, *for some integers* l_i, $i = 0, ..., m$. □

Notice that problem (5.1) does not contain any equality constraint that depends on ζ. The usual way of dealing with such uncertain equalities in (ARC) is to eliminate adjustable variables [66, Section 7]. This means that we are implicitly forcing the adjustable variables that are eliminated to obey specific decision rules. This is not allowed in (SRC). We illustrate this in Example 5.4 of Section 5.5.

We will now outline the assumptions used in this chapter to express conditions under which $Opt(SRC) = Opt(ARC)$.

Assumptions. All the assumptions are with respect to problem (5.1). Throughout this chapter, we assume that

i. There is no equality constraint in problem (5.1) that depends on ζ.

ii. The uncertainty set $\mathcal{Z}$ is compact.

iii. The uncertainty set $\mathcal{Z} \subset \mathbb{R}^l$ is convex.

iv. $\mathcal{Y}(x)$ is a convex set for each $x \in \mathcal{X}$.

v. $\mathcal{Y}(x)$ is a compact set for each $x \in \mathcal{X}$.

vi. $f(., x, y)$ and $g_i(., x, y)$ are concave for each $x \in \mathcal{X}$, $y \in \mathcal{Y}(x)$, and $i = 1, ..., m$.

vii. $f(\zeta_0, x, .)$ and $g_i(\zeta_i, x, .)$ are convex for each $x \in \mathcal{X}$, $\zeta \in \mathcal{Z} = \mathcal{Z}_0 \times ... \times \mathcal{Z}_m$, and $i = 1, ..., m$.

Assumptions i, iii, iv, vi, and vii are essentially the framework of the static robust convex optimization considered in [26].

5.2.2 Constraint-wise uncertainty

In this subsection, we study problems with constraint-wise uncertainty and provide sets of conditions, under which $Opt(SRC)$ and $Opt(ARC)$ are equal.

Theorem 5.1 *If problem (5.1) has constraint-wise uncertainty and Assumptions i-vii hold, then $Opt(SRC) = Opt(ARC)$.*

Proof. The line of reasoning is the same as in [25, Theorem 2.1].

Case I: Suppose that (ARC) does not have a non-adjustable variable. First, we assume that (SRC) is feasible. So, it is sufficient to show that whenever $\bar{t} \geq Opt(ARC)$, then $\bar{t} \geq Opt(SRC)$ (feasibility of (SRC) implies $Opt(ARC) < \infty$). According to the definitions, we have:

$$Opt(ARC) = \tag{5.3}$$
$$\inf \left\{ t \;\middle|\; \forall \zeta \in \mathcal{Z} = \mathcal{Z}_0 \times ... \times \mathcal{Z}_m \;\; \exists y \in \mathcal{Y} : \begin{array}{ll} f(\zeta_0, y) \leq t, & \\ g_i(\zeta_i, y) \leq 0, & i = 1, ..., m \end{array} \right\}$$

and,

$$Opt(SRC) = \tag{5.4}$$
$$\inf \left\{ t \;\middle|\; \exists y \in \mathcal{Y} \;\; \forall \zeta \in \mathcal{Z} = \mathcal{Z}_0 \times ... \times \mathcal{Z}_m : \begin{array}{ll} f(\zeta_0, y) \leq t, & \\ g_i(\zeta_i, y) \leq 0, & i = 1, ..., m \end{array} \right\}.$$

If $\mathcal{Y} = \emptyset$, it is clear that $Opt(ARC) = Opt(SRC) = +\infty$. Now, assume that $\mathcal{Y} \neq \emptyset$. By contradiction, suppose that there is a scalar $\bar{t}$ such that $\bar{t} \geq Opt(ARC)$ and $\bar{t} < Opt(SRC)$. Because of the constraint-wise uncertainty, by setting $\beta = (1, 0, 0, ..., 0)^T$, $G_0(\zeta_0, y) = f(\zeta_0, y)$, and $G_i(\zeta_i, y) = g_i(\zeta_i, y)$, for $i = 1, ..., m$, and by (5.4), it follows that

$$\forall y \in \mathcal{Y} \;\; \exists \zeta^y \in \mathcal{Z} \; \exists i_y \in \{0, ..., m\} : G_{i_y}(\zeta^y_{i_y}, y) - \beta_{i_y} \bar{t} > 0.$$

Also, continuity implies

$$\forall y \in \mathcal{Y} \;\; \exists \epsilon^y > 0 \; \exists U_y \;\; \forall z \in U_y : G_{i_y}(\zeta^y_{i_y}, z) - \beta_{i_y} \bar{t} > \epsilon^y, \tag{5.5}$$

where U_y is the intersection of a 2-norm open ball with a strictly positive radius centered at y with $\mathcal{Y}$. Since $\mathcal{Y}$ is compact, there are $y^k \in \mathcal{Y}$, $k = 1, ..., N$, such that $\mathcal{Y} = \cup_{k=1}^N U_{y^k}$. So,

$$\forall z \in \mathcal{Y} \quad \max_k G_{i_{y^k}}(\zeta^{y^k}_{i_{y^k}}, z) - \beta_{i_{y^k}} \bar{t} > \epsilon, \tag{5.6}$$

where $\epsilon = \min_k \epsilon^{y^k}$. As a simplification, we set $\zeta^k = \zeta^{y^k}_{i_{y^k}}$, $i_k = i_{y^k}$ and

$$f_k(z) = G_{i_k}(\zeta^k, z) - \beta_{i_k}\bar{t} \qquad \forall z \in \mathcal{Y}.$$

Since $\mathcal{Y}$ is convex and all $f_k(z)$ are convex and continuous on $\mathcal{Y}$ due to Assumption vii, and because $\max_k f_k(z) \geq \epsilon$ for each $z \in \mathcal{Y}$, there are nonnegative weights λ_k with $\sum_k \lambda_k = 1$ such that

$$f(z) := \sum_k \lambda_k f_k(z) \geq \epsilon \qquad \forall z \in \mathcal{Y}. \tag{5.7}$$

We define

$$\begin{aligned} w_i &= \sum_{k:i_k=i} \lambda_k \qquad i = 0, ..., m \\ \bar{\zeta}_i &= \begin{cases} \sum_{k:i_k=i} \frac{\lambda_k}{w_i} \zeta^k, & w_i \neq 0 \\ \text{an arbitrary point in } \mathcal{Z}_i, & w_i = 0 \end{cases} \\ \bar{\zeta} &= \left[\bar{\zeta}_0, ..., \bar{\zeta}_m\right]. \end{aligned} \tag{5.8}$$

It is clear by convexity of $\mathcal{Z}$ that $\bar{\zeta} \in \mathcal{Z}$. Additionally, due to $\bar{t} \geq Opt(ARC)$, we have

$$\exists t \leq \bar{t} : \forall \zeta \in \mathcal{Z}\ \exists y \in \mathcal{Y}, \quad \begin{array}{l} f(\zeta_0, y) \leq t, \\ g_i(\zeta_i, y) \leq 0, \quad i = 1, ..., m, \end{array} \tag{5.9}$$

which means

$$\exists \bar{y} \in \mathcal{Y} : G_i(\bar{\zeta}_i, \bar{y}) - \beta_i \bar{t} \leq 0, \qquad i = 0, ..., m. \tag{5.10}$$

Also, we know that for each $i = 0, ..., m$, the functions $G_i(\zeta_i, \bar{y})$ are concave in ζ_i due to Assumption vi. Hence, for all $i = 0, ..., m$, and $w_i > 0$

$$\begin{aligned} G_i(\bar{\zeta}_i, \bar{y}) - \beta_i \bar{t} \quad &= G_i\left(\sum_{k:i_k=i} \frac{\lambda_k}{w_i}\zeta^k, \bar{y}\right) - \beta_i \bar{t} \\ &\geq \sum_{k:i_k=i} \frac{\lambda_k}{w_i} G_i(\zeta^k, \bar{y}) - \beta_i \bar{t} = \sum_{k:i_k=i} \frac{\lambda_k}{w_i} f_k(\bar{y}). \end{aligned}$$

Summing over the indices results in

$$\sum_{\substack{i=1 \\ w_i \neq 0}}^{m} w_i \left(G_i(\bar{\zeta}_i, \bar{y}) - \beta_i \bar{t}\right) \geq \sum_{k=1}^{N} \lambda_k f_k(\bar{y}). \tag{5.11}$$

By applying (5.7) and (5.10), the above inequality contradicts $\epsilon > 0$.

Now we consider the case where (SRC) is not feasible, which means $Opt(SRC) = +\infty$. To prove equality of (SRC) and (ARC) with respect to the worst-case objective value, it is sufficient to show that there is no $\bar{t} \in \mathbb{R}$ such that $\bar{t} \geq Opt(ARC)$. So, the same argument used in the previous part implies that $Opt(ARC) = +\infty$.

Case II: Now, we consider a general case, where (ARC) contains the non-adjustable variable x. As proved in Case I, for any $x \in \mathcal{X}$,

$$\begin{array}{lll} \sup_{\zeta\in\mathcal{Z}} & \inf_{y(\zeta)\in\mathcal{Y}(x)} & f(\zeta, x, y(\zeta)) \\ & \text{s.t.} & g_i(\zeta, x, y(\zeta)) \leq 0, \quad i = 1, ..., m, \end{array} \tag{5.12}$$

and

$$\begin{array}{lll} \inf_{y\in\mathcal{Y}(x)} & \sup_{\zeta\in\mathcal{Z}} & f(\zeta_0, x, y) \\ \text{s.t.} & g_i(\zeta_i, x, y(\zeta_0, ..., \zeta_m)) \leq 0, & \forall \zeta_i \in \mathcal{Z}_i, \quad i = 1, ..., m, \end{array} \tag{5.13}$$

have the same optimal value. Therefore, taking the infimum over all $x \in \mathcal{X}$ results in $Opt(SRC) = Opt(ARC)$. $\square$

Theorem 5.1 extends the results for linear problems, [27, Proposition 2.1], to nonlinear ones. In the following theorem, we replace Assumption v in Theorem 5.1 with two other assumptions in order to provide another set of conditions under which $Opt(SRC) = Opt(ARC)$. For this theorem, without loss of generality, we assume that (SRC) is

$$\begin{array}{lll} \inf\limits_{x\in\mathcal{X}} \inf\limits_{y\in\mathcal{Y}(x)} & c^T y & \\ \text{s.t.} & g_i(\zeta_i, x, y) \leq 0, \quad i = 0, ..., m, & \forall \zeta_i \in \mathcal{Z}_i, \end{array} \tag{5.14}$$

where $c \in \mathbb{R}^r$ is certain, and for $i = 0, ..., m$,

$$\mathcal{Z}_i = \{\zeta_i : \; h_{ik}(\zeta_i) \leq 0, \; k = 1, ..., K_i\}.$$

In what follows, the relative interior of a set S and domain of a function $f(.)$ are denoted by relint(S) and dom$(f(.))$, respectively.

Theorem 5.2 *Assume that for problem (5.14) the following assumptions hold:*

(a) $h_{ik}(.)$ *is convex,* $i = 0, ..., m$, $k = 1, ..., K_i$,

(b) There exists $(\zeta_0, ..., \zeta_m)$ *such that* $h_{ik}(\zeta_i) < 0$ *for all* $i = 0, ..., m$, $k = 1, ..., K_i$;

(c) For each $x \in \mathcal{X}$ *and* $\zeta \in \mathcal{Z}$;

$$\bigcap_{i=1}^{m} relint\,(dom\,(g_i(\zeta_i, x, .))) \bigcap relint(\mathcal{Y}(x)) \neq \emptyset.$$

Additionally, if Assumptions iv, vi, and vii hold, then $Opt(ARC) = Opt(SRC)$.

Proof. Consider the (ARC) corresponding to (5.14) to be

$$\begin{aligned} \inf_{x\in\mathcal{X}} \sup_{[\zeta_0,\ldots,\zeta_m]\in\mathcal{Z}} \inf_{y(\zeta)\in\mathcal{Y}(x)} \quad & c^T y(\zeta) \\ \text{s.t. } & g_i(\zeta_i, x, y(\zeta)) \le 0, \quad i = 0, \ldots, m. \end{aligned} \tag{5.15}$$

By [2, Lemma 9] (because of Assumptions iv and vii, and assumption (c)), the optimal value of (5.15) is equal to

$$\begin{aligned} \inf_{x\in\mathcal{X}} \sup_{\substack{u\,\in\,\mathbb{R}^{m+1}_{\le}\\ \{v^i\},\, v^{m+1}}} \sup_{\zeta=[\zeta_0,\ldots,\zeta_m]} \quad & \sum_{i=0}^{m} u_i g_i^*\left(\zeta_i, x, \frac{v^i}{u_i}\right) + u_{m+1}\delta^*_{\mathcal{Y}(x)}\left(\frac{v^{m+1}}{u_{m+1}}\right) \\ \text{s.t.} \quad & \sum_{i=0}^{m+1} v^i = c, \\ & h_{ik}(\zeta_i) \le 0, \quad i = 0, \ldots, m, \ k = 1, \ldots, K_i, \end{aligned} \tag{5.16}$$

where $\delta^*_{\mathcal{Y}(x)}\left(\frac{v^{m+1}}{u_{m+1}}\right) = \sup_{y\in\mathcal{Y}(x)} \frac{y^T v^{m+1}}{u_{m+1}}$ and

$$g_i^*\left(\zeta_i, x, \frac{v^i}{u_i}\right) = \sup_{y\in dom(g_i(\zeta_i, x, .))} \left\{ \frac{y^T v^i}{u_i} - g_i\left(\zeta_i, x, y\right) \right\}.$$

Problem (5.16) has the same optimal objective value as

$$\begin{aligned} \inf_{x\in\mathcal{X}} \sup_{\substack{u\,\in\,\mathbb{R}^{m+1}_{\le}\\ \{v^i\},\, v^{m+1}}} \sup_{w^i} \quad & \sum_{i=0}^{m} u_i g_i^*\left(\frac{w^i}{u_i}, x, \frac{v^i}{u_i}\right) + u_{m+1}\delta^*_{\mathcal{Y}(x)}\left(\frac{v^{m+1}}{u_{m+1}}\right) \\ \text{s.t.} \quad & \sum_{i=0}^{m+1} v^i = c, \\ & - u_i h_{ik}\left(\frac{w_i}{u_i}\right) \le 0, \quad i = 0, \ldots, m, \ k = 1, \ldots, K_i, \end{aligned}$$

which is the dual of (5.14), with the same optimal objective values according to [13, Theorem 1] (because the uncertainty is constraint-wise and assumptions (a) and (b), as well as Assumptions vi and vii, hold). So, $Opt(ARC) = Opt(SRC)$. □

For a problem with fixed recourse and constraint-wise uncertainty, Assumption ii (without any convexity assumption) implies equality of the objective values of (SRC) and (ARC). We prove this in the following theorem. Even though in this case the resulting (SRC) is intractable in general, there are cases for which (SRC) is tractable, for instance see [24, Section 1.4].

Theorem 5.3 *Assume that in problem (5.1), the uncertainty is constraint-wise and the uncertainty set is compact. If (ARC) has fixed recourse, then Opt(SRC) = Opt(ARC).*

Proof. First, we suppose problem (5.1) does not contain any non-adjustable variables. According to the definitions of (SRC) and (ARC), we have $Opt(ARC) \leq Opt(SRC)$. That means that if (SRC) is unbounded, then $Opt(SRC) = Opt(ARC) = -\infty$. Now, if ($SRC$) is not unbounded, we show that $Opt(ARC) \geq Opt(SRC)$.

Since (ARC) has fixed recourse, we can simplify (SRC) to the following problem:

$$\begin{aligned} \inf_{y\in\mathcal{Y},\ t} \quad & t \\ \text{s.t.} \quad & \tilde{f}(y) + \sup_{\zeta_0\in\mathcal{Z}_0} \bar{f}(\zeta_0) \leq t \\ & \tilde{g}_i(y) + \sup_{\zeta_i\in\mathcal{Z}_i} \bar{g}_i(\zeta_i) \leq 0, \quad i = 1, ..., m. \end{aligned} \tag{5.17}$$

Since $\bar{f}$ and $\bar{g}_i$, $i = 1, ..., m$, are continuous, and the uncertainty is constraint-wise, and $\mathcal{Z}$ is non-empty and compact, there is a point $\bar{\zeta} = [\bar{\zeta}_0, ..., \bar{\zeta}_m] \in \mathcal{Z}$ where $\bar{\zeta}_0$ is an optimal solution of $\sup_{\zeta_0\in\mathcal{Z}_0} \bar{f}(\zeta_0)$, and $\bar{\zeta}_i$ is an optimal solution of $\sup_{\zeta_i\in\mathcal{Z}_i} \bar{g}_i(\zeta_i)$, for all $i = 1, ..., m$. According to the definition of (ARC), $Opt(ARC) \geq q$, where

$$\begin{aligned} q := \inf_{\substack{y(\bar{\zeta}) \in \mathcal{Y} \\ t(\bar{\zeta})}} \quad & t(\bar{\zeta}) \\ & \tilde{f}(y(\bar{\zeta})) + \bar{f}(\bar{\zeta}_0) \leq t(\bar{\zeta}) \\ & \tilde{g}_i(y(\bar{\zeta})) + \bar{g}_i(\bar{\zeta}_i) \leq 0, \quad i = 1, ..., m, \end{aligned} \tag{5.18}$$

which is equivalent to (5.17). This implies that if (SRC) is infeasible, so is (5.18), and therefore $Opt(ARC) = Opt(SRC) = +\infty$. On the other hand, if (SRC) is feasible, then $Opt(SRC) = q \leq Opt(ARC)$. So, the equality of the optimal objective values of (ARC) and (SRC) has been proved.

Now, for the general case in which (ARC) contains a non-adjustable variable x, we have to solve:

$$\begin{aligned} \inf_{x\in\mathcal{X}} \sup_{\zeta\in\mathcal{Z}} \inf_{y(\zeta)\in\mathcal{Y}(x)} \quad & f(\zeta, x, y(\zeta)) \\ & g_i(\zeta, x, y(\zeta)) \leq 0, \quad i = 1, ..., m. \end{aligned} \tag{5.19}$$

According to the first part of the proof, we have that for each $x \in \mathcal{X}$, the objective value of

$$\begin{aligned} \sup_{\zeta\in\mathcal{Z}} \ \inf_{y(\zeta)\in\mathcal{Y}(x)} \quad & f(\zeta, x, y(\zeta)) \\ \text{s.t.} \quad & g_i(\zeta, x, y(\zeta)) \leq 0, \quad i = 1, ..., m \end{aligned} \tag{5.20}$$

is equal to the objective value of

$$\begin{array}{lll} \inf_{y\in\mathcal{Y}(x)} & \sup_{\zeta\in\mathcal{Z}} f(\zeta_0,x,y) & \\ \text{s.t.} & g_i(\zeta_i,x,y)\le 0, \quad \forall \zeta_i\in\mathcal{Z}_i, & i=1,...,m. \end{array} \tag{5.21}$$

It follows, then, that the optimal objective value of problem (5.19) equals that of

$$\begin{array}{lll} \inf_{x\in\mathcal{X},y\in\mathcal{Y}(x)} & \sup_{\zeta\in\mathcal{Z}} f(\zeta_0,x,y) & \\ \text{s.t.} & g_i(\zeta_i,x,y)\le 0, \quad \forall \zeta_i\in\mathcal{Z}_i, & i=1,...,m. \end{array} \tag{5.22}$$

Therefore, $Opt(ARC) = Opt(SRC)$. □

5.2.3 Non-constraint-wise uncertainty

Section 5.2.2 focuses on constraint-wise uncertainty. The question is what can be concluded for a problem in which some, but not all, of the uncertain parameters are constraint-wise. Consider the following problem:

$$\begin{array}{llll} (HRC) & \inf\limits_{x\in\mathcal{X}} \inf\limits_{y\in\mathcal{Y}(x),\, t} & t & \\ & \text{s.t.} & f(\zeta_0,\alpha,x,y)\le t & \forall \alpha\in\mathcal{A}, \zeta_0\in\mathcal{Z}_0, \\ & & g_i(\zeta_i,\alpha,x,y)\le 0, \quad i=1,...,m, & \forall \alpha\in\mathcal{A}, \zeta_i\in\mathcal{Z}_i, \end{array}$$

where $\zeta=(\zeta_0,...,\zeta_m)\in\mathcal{Z}=\mathcal{Z}_0\times...\times\mathcal{Z}_m$ and $\alpha\in\mathcal{A}\subseteq\mathbb{R}^d$ are uncertain parameters (ζ is constraint-wise and α is non-constraint-wise). This problem has a hybrid uncertainty, so we cannot use the results in Section 5.2.2 to deduce equality of the optimal values of the hybrid robust counterpart (HRC) and the corresponding hybrid adjustable robust counterpart (HARC). However, the following corollary states that if in such a case the same set of conditions as in Theorem 5.1 hold with respect to the constraint-wise uncertain parameters, then there exists an optimal decision rule that is a function of only the non-constraint-wise uncertain parameters. In other words, the two problems

$$\begin{array}{llll} (HARC) & \inf\limits_{x\in\mathcal{X}} \sup\limits_{\substack{\zeta\in\mathcal{Z}\\ \alpha\in\mathcal{A}}} \inf\limits_{\substack{y(\zeta,\alpha)\in\mathcal{Y}(x)\\ t(\zeta,\alpha)}} & t(\zeta,\alpha) & \\ & \text{s.t.} & f(\zeta_0,\alpha,x,y(\zeta,\alpha))\le t(\zeta,\alpha), & \\ & & g_i(\zeta_i,\alpha,x,y(\zeta,\alpha))\le 0, & i=1,...,m \end{array}$$

and

$$\begin{array}{llll} (HARC_\alpha) & \inf\limits_{x\in\mathcal{X}} \sup\limits_{\alpha\in\mathcal{A}} \inf\limits_{\substack{y(\alpha)\in\mathcal{Y}(x)\\ t(\alpha)}} & t(\alpha) & \\ & \text{s.t.} & f(\zeta_0,\alpha,x,y(\alpha))\le t(\alpha) & \forall\zeta_0\in\mathcal{Z}_0, \\ & & g_i(\zeta_i,\alpha,x,y(\alpha))\le 0, \quad i=1,...,m, & \forall\zeta_i\in\mathcal{Z}_i \end{array}$$

have the same optimal objective values. We denote the optimal objective values of $(HARC)$ and $(HARC_\alpha)$ by $Opt(HARC)$ and $Opt(HARC_\alpha)$, respectively.

Corollary 5.1 *Suppose that for all $\alpha \in \mathcal{A}$, the assumptions of Theorem 5.1 hold with respect to ζ, x, y. Then, $Opt(HARC) = Opt(HARC_\alpha)$.*

Proof. By fixing $\alpha \in \mathcal{A}$ and $x \in \mathcal{X}$ and applying Theorem 5.1, the optimal objective value of

$$\begin{aligned} \sup_{\zeta\in\mathcal{Z}} \inf_{\substack{y(\zeta,\alpha)\in\mathcal{Y}(x)\\ t(\zeta,\alpha)}} \quad & t(\zeta,\alpha) \\ \text{s.t. } & f(\zeta_0,\alpha,x,y(\zeta,\alpha)) \le t(\zeta,\alpha), \\ & g_i(\zeta_i,\alpha,x,y(\zeta,\alpha)) \le 0, \qquad i = 1,...,m \end{aligned}$$

and

$$\begin{aligned} \inf_{\substack{y(\alpha)\in\mathcal{Y}(x)\\ t(\alpha)}} \quad & t(\alpha) \\ \text{s.t. } & f(\zeta_0,\alpha,x,y(\alpha)) \le t(\alpha) \qquad \forall\zeta_0 \in \mathcal{Z}_0, \\ & g_i(\zeta_i,\alpha,x,y(\alpha)) \le 0, \qquad i = 1,...,m, \ \forall\zeta_i \in \mathcal{Z}_i \end{aligned}$$

are equal. The result follows from taking the supremum over $\alpha \in \mathcal{A}$ and infimum over $x \in \mathcal{X}$. $\square$

Corollary 5.1 can be used to reduce the complexity of solving adjustable robust optimization problems. This is because in order to solve $(HARC)$, one needs to find an optimal decision rule with respect to both α and ζ, but, we can ensure the existence of an optimal decision rule that only depends on α by applying this corollary.

It is important to note that if we restrict ourselves to one class of decision rules, e.g., affine decision rules, as is customary, then Corollary 5.1 does not necessarily guarantee the existence of an optimal affine decision rule that only depends on α. The following corollary, however, states that if the problem has fixed recourse with respect to the constraint-wise uncertain parameter ζ and we use a specific class of decision rules that are separable with respect to ζ and α, then there exists an optimal decision rule that depends only on α.

Let us denote by $\bar{y}_\omega(\alpha) : \mathbb{R}^d \to \mathbb{R}$ a function of α that belongs to a specific class parametrized by ω. One of the examples for $\bar{y}_\omega(\alpha)$ is a polynomial. In this case, ω could be the vector of coefficients for the monomials.

Corollary 5.2 *Assume that in (HARC),*

$$g_i(\zeta_i, \alpha, x, y) = \tilde{g}_i(\zeta_i, x) + \bar{g}_i(\alpha, x, y), \;\; i = 0, ..., m, \tag{5.23}$$

where $g_0(\zeta_0, \alpha, x, y) = f(\zeta_0, \alpha, x, y)$ *and* $\tilde{g}_i(\zeta_i, x)$ *and* $\bar{g}_i(\alpha, x, y)$ *are continuous for* $i = 0, ..., m$. *Also, assume that we restrict the decision rules to be in the form of* $y(\zeta) + \bar{y}_\omega(\alpha)$, *where* $y(.) : \mathbb{R}^l \longrightarrow \mathbb{R}^n$. *Then the optimal objective value of (HRC) when using this decision rule is equal to that of using decision rule* $y + \bar{y}_\omega(\alpha)$.

Proof. Consider the following problem:

$$\begin{aligned} \inf_{x\in\mathcal{X},\omega} \sup_{\zeta\in\mathcal{Z}} \quad & \inf_{y(\zeta),\ t(\zeta)} \; t(\zeta) \\ & \text{s.t. } \tilde{g}_0(\zeta_0, x) + \bar{g}_0(\alpha, x, y(\zeta) + \bar{y}_\omega(\alpha)) \le t(\zeta), \;\; \forall \alpha \in \mathcal{A}, \\ & \quad \tilde{g}_i(\zeta_i, x) + \bar{g}_i(\alpha, x, y(\zeta) + \bar{y}_\omega(\alpha)) \le 0, \;\; \forall \alpha \in \mathcal{A}, \;\; i = 1, ..., m, \\ & \quad y(\zeta) + \bar{y}_\omega(\alpha) \in \mathcal{Y}(x), \qquad \forall \alpha \in \mathcal{A}. \end{aligned} \tag{5.24}$$

Let $\mathcal{Y}(x) - \bar{y}_\omega(\alpha) := \{y - \bar{y}_\omega(\alpha) : \;\; y \in \mathcal{Y}(x)\}$, for any $\alpha \in \mathcal{A}$. By defining $\bar{\mathcal{Y}}(x, \omega) = \cap_{\alpha\in\mathcal{A}} [\mathcal{Y}(x) - \bar{y}_\omega(\alpha)]$ and

$$\hat{g}_i(x, \omega, y(\zeta)) = \sup_{\alpha\in\mathcal{A}} \bar{g}_i(\alpha, x, y(\zeta) + \bar{y}_\omega(\alpha)), \;\; i = 0, ..., m,$$

accordingly, we obtain an optimal objective value for (5.24) that is equal to the optimal objective value of

$$\begin{aligned} \inf_{x\in\mathcal{X},\omega} \sup_{\zeta\in\mathcal{Z}} \quad & \inf_{\substack{y(\zeta)\in\bar{\mathcal{Y}}(x,\omega),\\ t(\zeta)}} \; t(\zeta) \\ & \text{s.t. } \tilde{g}_0(\zeta_0, x) + \hat{g}_0(x, \omega, y(\zeta)) \le t(\zeta), \\ & \quad \tilde{g}_i(\zeta_i, x) + \hat{g}_i(x, \omega, y(\zeta)) \le 0, \quad i = 1, ..., m, \end{aligned} \tag{5.25}$$

which is the adjustable robust counterpart related to the following robust problem:

$$\begin{aligned} \inf_{x\in\mathcal{X},\omega} \quad & \inf_{y\in\bar{\mathcal{Y}}(x,\omega),\ t} \; t \\ & \text{s.t. } \tilde{g}_0(\zeta_0, x) + \hat{g}_0(x, \omega, y) \le t, \quad \forall \zeta_0 \in \mathcal{Z}_0, \\ & \quad \tilde{g}_i(\zeta_i, x) + \hat{g}_i(x, \omega, y) \le 0, \quad \forall \zeta_i \in \mathcal{Z}_i, \;\; i = 1, ..., m. \end{aligned} \tag{5.26}$$

In accordance with Theorem 5.3, (5.25) and (5.26) have the same optimal objective value. Using the definitions of $\bar{\mathcal{Y}}(x, \omega)$ and $\hat{g}_i(x, \omega, y)$, $i = 0, ..., m$, we can easily see that the optimal objective value of (5.26) is equal to the optimal objective value of

$$\begin{aligned} \inf_{x\in\mathcal{X},\omega} \quad & \inf_{y,\ t} \; t \\ & \text{s.t. } \tilde{g}_0(\zeta_0, x) + \bar{g}_0(\alpha, x, y + \bar{y}_\omega(\alpha)) \le t, \;\; \forall \alpha \in \mathcal{A}, \;\; \forall \zeta_0 \in \mathcal{Z}_0, \end{aligned} \tag{5.27}$$

$$\begin{aligned} &\tilde{g}_i\left(\zeta_i, x\right) + \bar{g}_i\left(\alpha, x, y + \bar{y}_\omega\left(\alpha\right)\right) \leq 0, \;\; \forall \alpha \in \mathcal{A}, \;\; \forall \zeta_i \in \mathcal{Z}_i, \;\; i = 1, ..., m, \\ &y + \bar{y}_\omega\left(\alpha\right) \in \mathcal{Y}\left(x\right), \qquad\qquad\qquad \forall \alpha \in \mathcal{A}. \end{aligned}$$

So, we have proved that the optimal objective value of (5.24) and (5.27) are the same. This means that the use of $y\left(\zeta\right) + \bar{y}_\omega\left(\alpha\right)$ and $y + \bar{y}_\omega\left(\alpha\right)$ as decision rules yields the same approximation of the optimal objective values. □

In Corollary 5.2, $y(\zeta)$ is a general function. For instance, if we assume that $\bar{y}_\omega\left(\alpha\right)$ lies in the class of affine functions, even for a general $y(\zeta)$, the optimal objective value is independent from ζ. The other example is when both $y(\zeta)$ and $\bar{y}_\omega\left(\alpha\right)$ are affine, which means that the decision rule is affine. We consider this case in the next corollary.

Corollary 5.3 *Suppose that in (HARC) the constraints and objective functions satisfy (5.23). Then, using an affine decision rule,* $y(\alpha) = u + W\alpha$ *or* $y(\zeta, \alpha) = u + V\zeta + W\alpha$*, where* $u \in \mathbb{R}^n$*,* $V \in \mathbb{R}^{n\times l}$*, and* $W \in \mathbb{R}^{n\times d}$*, yields the same approximate optimal value.* □

Corollary 5.3 mentions two different problems for approximating $(HARC)$: one considers $y(\alpha) = u+W\alpha$ as the form of decision rule and the other $y(\zeta, \alpha) = u+V\zeta+W\alpha$. We denote the optimal objective values of the former affinely adjustable robust counterparts by $Opt(AARC_\alpha)$ and $Opt(AARC_{\zeta,\alpha})$, respectively. Then, in general, for problem (HRC), we have

$$Opt(HARC) \leq Opt(AARC_{\zeta,\alpha}) \leq Opt(AARC_\alpha) \leq Opt(HRC). \tag{5.28}$$

In this section, we discussed conditions that turn inequalities in (5.28) into equalities. Theorems 5.1, 5.2, and 5.3 provide sets of conditions under which all of the inequalities can be replaced by equalities. In addition, under similar sets of conditions as in those theorems, Corollary 5.3 ensures us that the middle inequality in (5.28) turns into an equality. Other sets of conditions for which $Opt(HARC) = Opt(AARC_{\zeta,\alpha})$ are proposed in [33, 37, 74]. In Section 5.5, we provide some examples to show that these inequalities can be strict.

5.3 Applications

In this section, we present some applications of the results obtained in Section 5.2. In Section 5.3.1 we show by using Theorem 5.3 that between two deterministic equivalent formulations of facility location problems with binary adjustable variables, one is better than the other with respect to the robust optimal value. In Section 5.3.2 we use Corollary 5.1 to reduce the complexity and the size of the affinely adjustable robust counterpart. In Section 5.3.3 we show that for a class of two-stage LO problems, a part of the results in [35] is a direct consequence of Corollary 5.1. Finally, for problems with uncertain convex quadratic and conic quadratic constraints, we use Theorem 5.1 and Corollary 5.1 to show that if the uncertainty in the quadratic constraints is constraint-wise, then there exist optimal solutions for the adjustable variables that are independent of the constraint-wise uncertain parameters.

5.3.1 Facility location problem with uncertain demands

In this subsection, we extend the result in [13, Theorem 1] and show that this result also holds when parts of the adjustable variables are binary.

Assume that in a horizon with T periods, a facility should be assigned to some candidate locations in order to satisfy demands in different customers' locations. The goal is to find the best allocation that satisfies the demands and maximizes the profit. In [13] the authors study two equivalent formulations of this problem and show by using [25, Theorem 2.1] that if the demands are uncertain and the uncertainty set is a box, then the optimal value of the robust counterpart of one of the formulations is better than the other.

In order to describe the two formulations, we follow the description and notations from [18]. Let $T, L, N \in \mathbb{N}$, be the length of the horizon, the number of candidate locations to which a facility can be assigned, and the number of locations that have a demand for the facility, respectively. Let $\eta \in \mathbb{R}_+$ be the unit price of goods, and $c_i, C_i, K_i \in \mathbb{R}_+$ be the cost per unit of production, the cost per unit of capacity, and the cost of opening a facility at location i, respectively, for $i = 1, ..., L$. Moreover, let $d_{ij} \in \mathbb{R}_+$ be the cost of shipping units from location i to j, and $D_{j\tau} \in \mathbb{R}_+$ be the demand in period τ at location j, $i = 1, ..., L$, $j = 1, ..., N$, $\tau = 1, ..., T$. Decision variable $X_{ij\tau}$ represents the proportion of the demand at location j in period τ that is satisfied by facility i, and $P_{i\tau}$ represents the amount of good that is produced at facility i during the period τ. For each facility i, the decision variable $I_{i\tau}$ denotes whether the facility in location i is open or closed in period τ by taking 1 or 0 respectively, and $Z_{i\tau}$ denotes the capacity of the facility in this location and period in case it is open. Using these notations, a deterministic facility location problem is described by the following mixed integer LO [18]:

$$\begin{aligned}
\max_{\substack{X \in \mathbb{R}^{L\times N\times T} \\ I,\, Z,\, P \in \mathbb{R}^{L\times T}}} \quad & \sum_{\tau=1}^{T}\sum_{i=1}^{L}\sum_{j=1}^{N}(\eta - d_{ij})X_{ij\tau}D_{j\tau} - \sum_{\tau=1}^{T}\sum_{i=1}^{L}\left(c_i P_{i\tau} + C_i Z_{i\tau} + K_i I_{i\tau}\right) \\
\text{s.t.} \quad & \sum_{i=1}^{L} X_{ij\tau} \le 1, \quad j = 1, ..., N, \ \tau = 1, ..., T,
\end{aligned}$$

$$\sum_{j=1}^{N} X_{ij\tau} D_{j\tau} \le P_{i\tau}, \;\; i = 1, ..., L, \; \tau = 1, ..., T,$$
$$X_{ij\tau} \ge 0, \;\; i = 1, ..., L, \; j = 1, ..., N, \; \tau = 1, ..., T,$$
$$P_{i\tau} \le Z_{i\tau}, \; Z_{i\tau} \le M I_{i\tau}, \;\; i = 1, ..., L, \; \tau = 1, ..., T,$$
$$I \in \{0, 1\}^{L \times T},$$

where, M is a large enough constant. We call the above formulation, *proportion-shipping* formulation.

In [13] another equivalent formulation for the facility problem is studied, in which $X_{ij\tau} D_{j\tau}$ is replaced with $Y_{ij\tau}$ for all i, j and τ, where $Y_{ij\tau}$ represents how much good is shipped in the period τ from location i to j. We call the model with $Y_{ij\tau}$ *total-shipping* formulation.

Let the demands $D_{j\tau}$, $j = 1, ..., N$, $\tau = 1, ..., T$, be the uncertain parameters with interval uncertainty sets. In [13] the authors assume that I, Z are non-adjustable variables, whereas X, P in the *proportion-shipping* formulation, and Y, P in the *total-shipping* formulation are adjustable variables. However, we assume here that parts of I, Z are non-adjustable and the rest are adjustable variables.

Let us denote the optimal values of the robust counterpart of *proportion-shipping*, *total-shipping* formulations, and their adjustable robust counterparts by $Opt(RCproportion)$, $Opt(RCtotal)$, $Opt(ARCproportion)$, and $Opt(ARCtotal)$, respectively.

We show that the robust counterpart of the *total-shipping* formulation is better than the robust counterpart of the *proportion-shipping*, i.e.,

$$Opt(RCtotal) \ge Opt(RCproportion).$$

It is clear that the *proportion-shipping* and *total-shipping* formulations are equivalent for each realization of the demand. Therefore, the corresponding adjustable robust counterparts are equivalent as well, so $Opt(ARCtotal) = Opt(ARCproportion)$.

The uncertainty in the *total-shipping* formulation is constraint-wise, because the uncertain parameters appear only in $\sum_{i=1}^{L} Y_{ij\tau} \le D_{j\tau}$, which is constructed from the first constraint in the *proportion-shipping* formulation after the replacement $X_{ij\tau} D_{j\tau} = Y_{ij\tau}$, for each $i = 1, ..., L$, $j = 1, ..., N$, $\tau = 1, ..., T$. Therefore, $Opt(RCtotal) = Opt(ARCtotal)$, by Theorem 1. So, we have shown that

$$Opt(RCtotal) = Opt(ARCtotal) = Opt(ARCproportion) \ge Opt(RCproportion),$$

where the right inequality holds because the problem is a maximization one and the optimal objective value of the adjustable robust counterpart is not less than the optimal objective value of the robust counterpart. Hence, $Opt(RCtotal) \ge Opt(RCproportion)$.

In [13], the authors show that $Opt(RCtotal) \geq Opt(RCproportion)$ in [13, Theorem 1] for the case in which I, Z are non-adjustable variable. As observed in [13], the *proportion-shipping* formulation can be obtained from the *total-shipping* formulation by using a special decision rule $Y_{ij\tau} = X_{ij\tau}D_{j\tau}$, $i = 1, ..., L$, $j = 1, ..., N$ and $\tau = 1, ..., T$. One might think therefore that $Opt(RCtotal) \leq Opt(RCproportion)$. However, the contrary is true. This is caused by the fact that the decision rule does not have any constant term.

5.3.2 Inventory system problem with demand and cost uncertainty

We now apply the result of Section 5.2 to the inventory system problem in [25, Section 5], in which the authors only consider uncertainty in the demand and propose an affine decision rule to approximate the adjustable robust counterpart. If the cost is uncertain in addition to the demand, then the problem is not fixed recourse anymore and using an affine decision rule leads to a non-concave robust problem in the uncertain parameters. In this subsection, we show that by using Corollary 5.1, the affinely adjustable robust counterpart of the inventory system problem with demand and cost uncertainty can be reformulated to a LO problem.

To describe the inventory system problem in [25, Section 5], let $I, T \in \mathbb{N}$ be the number of producers and the length of the horizon, respectively. Assume that during the time period τ, the i-th producer produces $p_{i\tau}$ units with per-unit cost $c_{i\tau} \in \mathbb{R}_+$. Producer i has a production capacity $P_{i\tau} \in \mathbb{R}_+$ in period τ and overall capacity $Q_i \in \mathbb{R}_+$. Let v be the amount of the product in the warehouse at the beginning. Besides, assume that at period τ, the demand is $d_\tau \in \mathbb{R}_+$ and inventory has a minimal and maximal restriction of $v_{min} \in \mathbb{R}_+$ and $v_{max} \in \mathbb{R}_+$, respectively. The goal is to minimize the total cost. The problem described in [25] is as follows:

$$
\begin{aligned}
\min_{p \in \mathbb{R}^{I \times T}} \quad & \sum_{\tau=1}^{T} \sum_{i=1}^{I} c_{i\tau} p_{i\tau} \\
\text{s.t.} \quad & 0 \leq p_{i\tau} \leq P_{i\tau}, \ \sum_{\tau=1}^{T} p_{i\tau} \leq Q_i, \quad i = 1, ..., I, \ \tau = 1, ..., T, \\
& v_{min} \leq v + \sum_{s=1}^{\tau} \sum_{i=1}^{I} p_{is} - \sum_{s=1}^{\tau} d_s \leq v_{max}, \quad \tau = 1, ..., T.
\end{aligned}
\tag{5.29}
$$

Consider c and d as the uncertain parameters with box uncertainty for both parameters. The uncertainty set for d and $c_{i\tau}$ is denoted as D and $[\underline{c_{i\tau}}, \overline{c_{i\tau}}]$, $i = 1, ..., I$, $\tau = 1, ..., T$, respectively. Because (5.29) is not fixed recourse, using affine decision rules in c and d for the adjustable variable $p_{i\tau}$, leads to an optimization problem with a non-concave quadratic objective function in the uncertain parameter c.

Because c appears only in the objective function and for each realization of d in D, the problem is linear with the compact feasible and uncertainty set, so the assumptions of Corollary 5.1 hold. Therefore, the adjustable robust counterpart of (5.29) equals

$$\begin{aligned}
\max_{d\in D} \min_{p(d)\in\mathbb{R}^{I\times T}} \quad & \max_{\underline{c_{i\tau}} \leq c_{i\tau} \leq \overline{c_{i\tau}}} \sum_{\tau=1}^{T}\sum_{i=1}^{I} c_{i\tau}p_{i\tau}(d) \\
\text{s.t.} \quad & 0 \leq p_{i\tau}(d) \leq P_{i\tau}, \ \sum_{\tau=1}^{T} p_{i\tau}(d) \leq Q_i, \ \ i=1,...,I, \ \tau=1,...,T, \\
& v_{min} \leq v + \sum_{s=1}^{\tau}\sum_{i=1}^{I} p_{is}(d) - \sum_{s=1}^{\tau} d_s \leq v_{max}, \ \ \tau=1,...,T,
\end{aligned}$$

in which the objective function is equivalent to $\sum_{\tau=1}^{T}\sum_{i=1}^{I}\overline{c_{i\tau}}p_{i\tau}(d)$. So, we can approximate the above problem using affine decision rules only in d. This reduces the complexity and the number of variables compared to the problem acquired by using affine decision rules in c and d.

5.3.3 Two-stage linear optimization problems

Another application of Corollary 5.1 is for specific two-stage LO problems with uncertainty in both the constraints and the objective. In [35] a bound is derived for this class of problems and the authors show that if the uncertainty in the objective is independent of that in the constraints then this bound does not depend on the objective uncertainty. In this subsection, we show that this result is a direct consequence of Corollary 5.1. As in [35], we consider the adjustable robust counterpart corresponding to a LO problem

$$\begin{aligned}
(ARC_{LP}) \quad \min c^T x + \max_{(B,d)\in\mathcal{Z}} \min_{y(B,d)} \ & d^T y(B,d) \\
\text{s.t.} \ & Ax + By(B,d) \leq h, \\
& x \in \mathbb{R}^r_+, \\
& y(B,d) \in \mathcal{Y}(x),
\end{aligned}$$

where $A \in \mathbb{R}^{m\times r}$, $c \in \mathbb{R}^r_+$, $h \in \mathbb{R}^m$, $\mathcal{Y}(x) \subset \mathbb{R}^n_+$ is a polytope, B is an uncertain matrix, and d is an uncertain vector. Also, let $\mathcal{Z} = \mathcal{Z}^B \times \mathcal{Z}^d \subseteq \mathbb{R}^{m\times n}_+ \times \mathbb{R}^n_+$ be a convex compact uncertainty set. In addition, we suppose that $\mathcal{Z}^d$ is a polytope, as well. In [35], for problems with deterministic objective coefficient d, it is shown that

$$Opt(ARC_{LP}) \geq \rho(\mathcal{Z})Opt(SRC_{LP}), \tag{5.30}$$

where

$$\rho(\mathcal{Z}) = \max\left\{\kappa\left(T\left(\mathcal{Z},h\right)\right) \mid \ h > 0\right\},$$

$$T\left(\mathcal{Z},h\right)=\left\{B^T\mu\,\middle|\,h^T\mu=1,\ B\in\mathcal{Z},\ \mu\geq 0\right\},$$
$$\kappa\left(T\left(\mathcal{Z},h\right)\right)=\min\left\{\alpha\,|\,conv(T\left(\mathcal{Z},h\right))\subseteq\alpha T\left(\mathcal{Z},h\right)\right\},$$

and (SRC_{LP}) is the robust counterpart corresponding to (ARC_{LP}). Then, the authors in [35] show separately that for problem (ARC_{LP}), which has uncertainty on objective coefficient d and the matrix coefficient B, the lower bound is independent of the objective uncertainty, i.e., they show that $Opt(ARC_{LP})\geq\rho(\mathcal{Z}^B)Opt(SRC_{LP})$.

Here, we show that the latter result is a direct consequence of Corollary 5.1 and (5.30). To see that, consider the following problem

$$\begin{aligned}\min c^Tx+\max_{B\in\mathcal{Z}^B}\min_{y(B),\ t(B)}\ & t(B)\\ \text{s.t. }& \max_{d\in\mathcal{Z}^d}d^Ty(B)\leq t(B),\\ & Ax+By(B)\leq h,\\ & x\in\mathbb{R}^r_+,\\ & y(B)\in\mathcal{Y}(x).\end{aligned}\tag{5.31}$$

It is clear that all assumptions of Corollary 5.1 hold. So, applying this corollary to (ARC_{LP}) implies that (ARC_{LP}) and (5.31) have the same optimal objective values. Now, assume that $d^j\in\mathcal{Z}^d$, for $j=1,...,K$, are the extreme points of the polytope $\mathcal{Z}^d$. Then, the optimal objective value of (5.31) is equal to that of

$$\begin{aligned}\min_x c^Tx+\max_{B\in\mathcal{Z}^B}\min_{y(B),\ t(B)}\ & t(B)\\ \text{s.t. }& {d^j}^Ty(B)\leq t(B),\ \ j=1,...,K,\\ & Ax+By(B)\leq h,\\ & x\in\mathbb{R}^r_+,\\ & y(B)\in\mathcal{Y}(x).\end{aligned}\tag{5.32}$$

Defining

$$\bar{B}=\begin{pmatrix}{d^1}^T\\ \vdots\\ {d^K}^T\\ B\end{pmatrix},\ \ \bar{A}=\begin{pmatrix}0\\ \vdots\\ 0\\ A\end{pmatrix},\ \ \bar{h}=\begin{pmatrix}0\\ \vdots\\ 0\\ h\end{pmatrix},\ \ \beta=\begin{pmatrix}1\\ \vdots\\ 1\\ 0\end{pmatrix},$$

we rewrite (5.32) as

$$\begin{aligned}(\overline{ARC_{LP}})\quad \min_x c^Tx+\max_{\bar{B}\in\bar{\mathcal{Z}}}\min_{y(\bar{B}),\ t(\bar{B})}\ & t(\bar{B})\\ \text{s.t. }& \bar{A}x-\beta t(\bar{B})+\bar{B}y(\bar{B})\leq\bar{h},\\ & x\in\mathbb{R}^r_+,\\ & y(\bar{B})\in\mathcal{Y}(x),\end{aligned}$$

where $\bar{\mathcal{Z}} = [d^{1^T}, ..., d^{K^T}]^T \times \mathcal{Z}^B$. Consequently, we get

$$Opt(ARC_{LP}) = Opt(\overline{ARC_{LP}}) \geq \rho(\mathcal{Z}^B) Opt(SRC_{LP})$$

by applying (5.30) and the fact that $\rho(\bar{\mathcal{Z}}) = \rho(\mathcal{Z}^B)$.

It is worth mentioning that if the uncertainty set $\mathcal{Z}^B$ in (ARC_{LP}) is a Cartesian product of the uncertainty region of B^j with another set, where B^j is the j-th row of B, then it can be proved analogously that the bound is independent of the uncertainty in B^j. Even though this is not an extension of the results in [35], it gives an intuition behind why the bound is independent of constraint-wise uncertainty.

We emphasize that the proofs in [35] are for polytopal uncertainty sets. However, the authors provided us additional proofs for general uncertainty sets (private communications).

5.3.4 Uncertain problems containing convex quadratic and/or conic quadratic constraints

One application of the results derived in Section 5.2 is for the following problem:

$$\begin{aligned}
\inf_{x \in \mathcal{X}} \inf_{y \in \mathcal{Y}(x)} \quad & f(x, y) \\
\text{s.t.} \quad & g_j(\alpha, x, y) \leq 0, \ \forall \alpha \in \mathcal{A}, \ j = 1, ..., m, \\
& h_i(\zeta_i, x, y) \leq 0, \ \forall \zeta_i \in \mathcal{Z}_i, \ i = 1, ..., I, \\
& p_k(\theta_k, x, y) \leq 0, \ \theta_k \in \mathcal{T}_k, \ k = 1, ..., K,
\end{aligned}$$

where $g_j(\alpha, x, y)$, $j = 1, ..., m$, is a continuous function, the convex quadratic function h_i is defined as

$$h_i(\zeta_i, x, y) = \begin{pmatrix} x \\ y \end{pmatrix}^T A_i(\zeta_i) \begin{pmatrix} x \\ y \end{pmatrix} + b_i(\zeta_i)^T \begin{pmatrix} x \\ y \end{pmatrix} + c_i(\zeta_i),$$

and the conic quadratic function p_k is defined as

$$p_k(\theta_k, x, y) = \sqrt{\begin{pmatrix} x \\ y \end{pmatrix}^T B_k(\theta_k) \begin{pmatrix} x \\ y \end{pmatrix}} + d_k(\theta_k)^T \begin{pmatrix} x \\ y \end{pmatrix} + e_k(\theta_k),$$

where $\alpha \in \mathcal{R}^l$, $\zeta_i \in \mathcal{R}^{l_i}$, and $\theta_k \in \mathcal{R}^{l_{I+k}}$ are the uncertain parameters for some integers l, l_i, l_{I+k}, $i = 1, ..., I$, $k = 1, ..., K$, and x and y are non-adjustable and adjustable variables, respectively. We assume that the matrices $A_i(\zeta_i)$ and $B_k(\theta_k)$ are positive semi-definite for all $\zeta_i \in \mathcal{Z}_i$ and $\theta_k \in \mathcal{T}_k$, $i = 1, ..., I$, $k = 1, ..., K$. Also, we assume that $A_i(\zeta_i)$, $b_i(\zeta_i)$, $c_i(\zeta_i)$, $B_k(\theta_k)$, $d_k(\theta_k)$, and $e_k(\theta_k)$ are affine in ζ_i and θ_k, $i = 1, ..., I$, $k = 1, ..., K$, respectively.

This type of problem arises, for example, when a part of the problem is related to multi-stage mean-variance portfolio optimization [67], in which the asset return mean and covariance matrix are uncertain and these uncertainties only occur in the objective function (hence the problem has constraint-wise uncertainty).

If the uncertainty over α is constraint-wise and $g_j(\alpha, x, y)$ is concave in α and convex in y, $j = 1, ..., m$; $\mathcal{A}$, $\mathcal{Z}_i$ and $\mathcal{T}_k$ are convex, $i = 1, ..., I$, $k = 1, ..., K$; and $\mathcal{Y}(x)$ is compact and convex for all $x \in \mathcal{X}$, then by Theorem 5.1, the optimal values of the corresponding static and adjustable robust problems are equal, because h_i and p_k are convex in y and concave in ζ_i and θ_k, $i = 1, ..., I$, $k = 1, ..., K$, respectively. Moreover, if the uncertainty over α is not constraint-wise, then by Corollary 5.1, an optimal y exists for the corresponding adjustable robust counterpart that is independent of ζ_i and θ_k, $i = 1, ..., I$, $k = 1, ..., K$.

5.4 Illustrative examples

Example 5.1 ***(Illustrating Theorem 5.1)*** *Consider the following problem:*

$$\begin{aligned} \min\ & y_1 + y_2 \\ s.t.\ & \ln(\zeta) y_1^2 + y_2^2 \leq 3, \\ & y_1^2 + y_2^2 \leq 4, \end{aligned}$$

where $\zeta \in \mathcal{Z} = [1, 4]$ *is an uncertain parameter and* $y = (y_1, y_2)$ *is an adjustable variable.*

For this example, $Opt(SRC) = Opt(ARC)$ *because by defining*

$$\mathcal{Y} = \left\{ y \;\middle|\; y_1^2 + y_2^2 \leq 4 \right\},$$

we create the conditions under which the assumptions of Theorem 5.1 will hold for this problem. Since $\ln(\zeta)$ *is an increasing function,* (SRC) *is as follows:*

$$\begin{aligned} \min\ & y_1 + y_2 \\ s.t.\ & \ln(4) y_1^2 + y_2^2 \leq 3, \\ & y_1^2 + y_2^2 \leq 4, \end{aligned}$$

which has an optimal value of $-\sqrt{\frac{3\ln(4)}{1+\ln(4)}} \left(\frac{1}{\ln(4)} + 1 \right)$.

Even though $Opt(SRC) = Opt(ARC)$*, by using the symmetry bound introduced in [34], which is* $(1 + \rho)\, Opt(SRC) \leq Opt(ARC) \leq Opt(SRC)$*, where*

$$\rho = \min \left\{ \alpha \geq 0 \;\middle|\; \mathcal{Z} - (1 - \alpha)\frac{5}{2} \subset \mathbb{R}_+ \right\} = \frac{3}{5},$$

one gets $\left(\frac{8}{5}\right) Opt(SRC) \leq Opt(ARC) \leq Opt(SRC)$.

This example shows that the symmetry bound is not tight in the presence of constraint-wise uncertainty, even when the problem only has one uncertain parameter. $\square$

Example 5.2 ***(Illustrating Theorem 5.3)*** *Consider the uncertain problem*

$$\begin{aligned} \min\ & y^2 + x^3 \\ s.t.\ & y^3 + \zeta^3 x \leq 0, \\ & y^2 + x^2 \leq 8, \\ & |x| \leq 1, \end{aligned}$$

where $\zeta \in \mathcal{Z} = [-2, 2]$ *is an uncertain parameter,* y *is an adjustable variable, and* x *is a non-adjustable variable. For this problem,*

$$\mathcal{X} = [-1, 1], \ \mathcal{Y}(x) = \left\{ y \ \middle|\ y^2 + x^2 \leq 8 \right\}, \ \forall x \in \mathcal{X}.$$

First, we use Theorem 5.3 to calculate $Opt(ARC)$*, because the relevant assumptions hold for this problem. According to this theorem,* $Opt(ARC) = Opt(SRC)$*. Since* ζ^3 *is an increasing function,* (SRC) *is equivalent to*

$$\begin{aligned} \min\ & y^2 + x^3 \\ s.t.\ & y^3 + 8x \leq 0, \\ & y^3 - 8x \leq 0, \\ & y^2 + x^2 \leq 8, \\ & |x| \leq 1. \end{aligned}$$

It is easy to verify that $Opt(SRC) = 0$*. Now we solve the* (ARC) *problem*

$$\begin{aligned} \min_{x \in \mathcal{X}} \ \max_{\zeta \in \mathcal{Z}} \ \min_{y(\zeta)} \ & y(\zeta)^2 + x^3 \\ s.t.\ & y(\zeta)^3 + \zeta^3 x \leq 0, \\ & y(\zeta)^2 + x^2 \leq 8 \end{aligned}$$

directly. First, we solve

$$\begin{aligned} z^*(\zeta, x) := \ \min_{y(\zeta)} \ & y(\zeta)^2 \\ s.t.\ & y(\zeta)^3 + \zeta^3 x \leq 0, \\ & y(\zeta)^2 + x^2 \leq 8 \end{aligned}$$

for each $\zeta \in \mathcal{Z}$ *and* $x \in \mathcal{X}$*. It is clear that*

$$z^*(\zeta, x) = \begin{cases} \left(\sqrt[3]{-\zeta^3 x}\right)^2, & \zeta x \geq 0, \\ 0, & \textit{otherwise.} \end{cases}$$

Hence, Opt(ARC) $= \min_{x\in\mathcal{X}} x^3 + \max_{\zeta\in\mathcal{Z}} z^*(\zeta, x)$. *Therefore, we need the optimal objective value of* $\max_{\zeta\in[-2,2]}\ z^*(\zeta,x)$ *for each* $x \in \mathcal{X}$. *By checking two cases* $x \geq 0$ *and* $x < 0$, *we find* $4\sqrt[3]{x^2}$ *as its optimal objective value. Hence, Opt(ARC)* $=$ $min_{x\in[-1,1]} x^3 + 4\sqrt[3]{x^2} = 0$. □

Hitherto, we have studied examples regarding constraint-wise uncertainty. Now, we consider an example possessing hybrid uncertainty.

Example 5.3 ***(Hybrid uncertainty)*** *Consider the following uncertain problem:*

$$\begin{aligned} \min_{y,x}\ & -x \\ s.t.\ \ & (1-2\alpha)x + y \geq \zeta, \\ & \alpha x - y \geq 0, \\ & x \leq 1, \end{aligned} \tag{5.33}$$

where $\alpha \in [0,1]$ *is a non-constraint-wise and* $\zeta \in [-1,0]$ *a constraint-wise uncertain parameter,* y *is an adjustable variable, and* x *is a non-adjustable variable.*

Corollary 5.1 shows that there exists an optimal decision rule for $(HARC)$ *that is independent of* ζ. *In this example, we check the inequalities in (5.28). First, we find the optimal objective values of the static and adjustable robust counterparts corresponding to (5.33). After that, we discuss the dependency of the optimal decision rules on the uncertain parameters in the adjustable robust optimization problem.*

To calculate the optimal value of the robust counterpart corresponding to (5.33), it is sufficient to solve the following problem:

$$\begin{aligned} q^*_{SRC} = \min_{y,x}\ & -x \\ s.t.\ \ & x + y \geq 0, \\ & -x + y \geq 0, \\ & -y \geq 0, \\ & x - y \geq 0, \\ & x \leq 1, \end{aligned}$$

because the constraints in (5.33) are linear with respect to the uncertain parameters α *and* ζ. *This means that* $(0,0)$ *is the only robust feasible solution of (5.33). Hence,* $q^*_{SRC} = 0$.

The adjustable robust counterpart corresponding to (5.33) is as follows:

$$\begin{aligned} q^*_{ARC} = \min_{x}\ \max_{(\alpha,\zeta)\in\mathcal{Z}}\ \min_{y(\alpha,\zeta)}\ & -x \\ s.t.\ \ & (1-2\alpha)x + y(\alpha,\zeta) \geq \zeta, \\ & \alpha x - y(\alpha,\zeta) \geq 0, \\ & x \leq 1, \end{aligned} \tag{5.34}$$

where (α, ζ) *is the uncertain parameter and* $\mathcal{Z} = [0,1] \times [-1,0]$ *is the uncertainty set. According to the last constraint,* $q^*_{ARC} \geq -1$*. Fixing* $x = 1$*, we have*

$$\zeta + 2\alpha - 1 \leq y(\alpha, \zeta) \leq \alpha, \tag{5.35}$$

which means that $q^*_{ARC} = -1$ *by choosing* $y^*(\alpha,\zeta) = \zeta + 2\alpha - 1$ *as the optimal decision rule, which depends on both* α *and* ζ*. However,* $y^{**}(\alpha,\zeta) = \alpha$ *is another optimal decision rule for (5.34), which is independent of* ζ*. We thus show by this discussion that for (5.33) there is only one strict inequality in (19):*

$$-1 = Opt(HARC) = Opt(AARC_{\zeta,\alpha}) = Opt(AARC_{\alpha}) < Opt(HRC) = 0. \qquad \Box$$

5.5 Counterexamples when one of the conditions is not satisfied

In this section, we consider examples in which all of the assumptions of Theorem 5.1 are satisfied except one. Each example is associated with the assumption not satisfied.

Example 5.4 ***(Assumption i)*** *Consider the following problem, in which Assumption i is not satisfied because there is an equality constraint that is dependent on* ζ*:*

$$\begin{aligned} \min \ & -y_1 \\ s.t. \ & \zeta y_1 + y_2 = 1, \\ & 0 \leq y_1, y_2 \leq 10, \end{aligned} \tag{5.36}$$

where $\zeta \in [1,2]$*. It is clear that* $Opt(SRC) = 0$*, since* $(0,1)$ *is the only robust feasible solution.*

To calculate the optimal value of the corresponding (ARC)*, we eliminate the equality constraint in (5.36) and reach the following adjustable robust problem:*

$$\begin{aligned} \max_{\zeta \in \mathcal{Z}} \ \min_{y_1(\zeta)} \ & -y_1(\zeta) \\ s.t. \ & 0 \leq y_1(\zeta) \leq \frac{1}{\zeta}. \end{aligned} \tag{5.37}$$

It is clear that the optimal value of (5.37) is $\max_{\zeta \in [1,2]} -\frac{1}{\zeta} = -\frac{1}{2}$*. Hence,* $Opt(ARC) < Opt(SRC)$*. These optimal values are different because in the elimination, we use the decision rule* $y_2 = 1 - \zeta y_1$*, which is not allowed in the corresponding* (SRC)*.* $\Box$

Example 5.5 ***(Assumption ii)*** *Consider the following problem*

$$\begin{aligned} \min \ & -y^2 \\ s.t. \ & y \leq \zeta, \end{aligned} \tag{5.38}$$

where $\zeta \leq 0$. This problem does not satisfy Assumption ii, since the uncertainty set is not compact. It is clear that Opt$(SRC) = +\infty$, because (SRC) is infeasible. However, (ARC) is feasible and Opt$(ARC) = -\infty$. □

Example 5.6 ***(Constraint-wise uncertainty)*** *In [25], the authors consider the following uncertain problem:*

$$\begin{aligned} \min \;& -x \\ s.t. \;& (1-2\zeta)x + y \geq 0, \\ & \zeta x - y \geq 0, \\ & 0 \leq x \leq 1, \\ & |y| \leq 2, \end{aligned}$$

where $\zeta \in [0,1]$ is an uncertain parameter, y is an adjustable variable, and x is a non-adjustable variable. It is easy to check that all assumptions hold except "constraint-wise uncertainty". The corresponding (SRC) can be reformulated as

$$\begin{aligned} \min \;& -x \\ s.t. \;& x + y \geq 0, \\ & x - y \geq 0, \\ & -x + y \geq 0, \\ & -y \geq 0, \\ & 0 \leq x \leq 1, \\ & |y| \leq 2. \end{aligned}$$

It can easily be verified here that Opt$(SRC) = 0$. The corresponding (ARC) is as follows:

$$\begin{aligned} \min_{x} \max_{\zeta} \min_{y(\zeta)} \;& -x \\ s.t. \;& (1-2\zeta)x + y(\zeta) \geq 0, \\ & \zeta x - y(\zeta) \geq 0, \\ & 0 \leq x \leq 1, \\ & |y(\zeta)| \leq 2. \end{aligned} \tag{5.39}$$

Similar to the discussion in Example 5.3, we can verify that Opt$(ARC) = -1$, which means Opt$(ARC) <$ Opt(SRC). □

Example 5.7 ***(Assumption iii)*** *Consider the problem $\min_{y \in \mathcal{Y}} \zeta y$, where $\zeta \in \mathcal{Z}$ is the uncertain parameter, $\mathcal{Z} = \{-1, 2\}$ is the uncertainty set, and $\mathcal{Y} = [-1, 1]$. It is clear that all the assumptions of Theorem 5.1 hold except iii. A straightforward calculation leads us to*

$$Opt(SRC) \;=\; \min_{y \in \mathcal{Y}} \max_{\zeta \in \mathcal{Z}} \;\zeta y = \min\{\min_{y \in [0,1]} \max_{\zeta \in \mathcal{Z}} \;\zeta y,\; \min_{y \in [-1,0]} \max_{\zeta \in \mathcal{Z}} \;\zeta y\}$$

$$= \min\{\min_{y\in[0,1]} 2y, \ \min_{y\in[-1,0]} -y\} = 0,$$

and

$$Opt(ARC) = \max_{\zeta\in\mathcal{Z}} \min_{y(\zeta)\in X} \zeta y(\zeta) = \max\{\min_{y\in X} -y, \min_{y\in X} 2y\} = \max\{-1,-2\} = -1.$$

So, $Opt(ARC) < Opt(SRC)$. However, if we replace $\mathcal{Z}$ with $Conv(\mathcal{Z})$, then $Opt(SRC)$ remains the same but $Opt(ARC)$ becomes zero, which shows that convexity of $\mathcal{Z}$ is crucial for arriving at $Opt(ARC) = Opt(SRC)$. □

Example 5.8 ***(Assumption iv)*** *As a counterexample for cases in which Assumption iv is not satisfied, we can use the problem in Example 5.7 with $\mathcal{Z} = [-1,2]$ and $\mathcal{Y} = \{-1,1\}$. Then, $Opt(ARC) = 0 < Opt(SRC) = 1$.* □

Example 5.9 ***(Assumption vi)*** *Consider the problem*

$$\begin{aligned} \min \quad & -y_1 - y_2 \\ s.t. \quad & \zeta^2 + (1-\zeta)y_1 + (1+\zeta)y_2 \leq 3, \\ & |y_i| \leq 3, \quad i = 1,2, \end{aligned} \tag{5.40}$$

where $\zeta \in [-1,1]$ is an uncertain parameter and $y = (y_1, y_2)$ is an adjustable variable. It is clear that (5.40) is not concave in ζ, but convex in y_1 and y_2. Also, $\mathcal{Z} = [-1,1]$ and $\mathcal{Y} = \{(y_1,y_2) : \ |y_i| \leq 3, \ i = 1,2\}$ are compact and convex and the uncertainty is constraint-wise. The (SRC) corresponding to (5.40) is as follows:

$$\begin{aligned} \min \quad & -y_1 - y_2 \\ s.t. \quad & \max_{\zeta\in[-1,1]} \left[\zeta^2 + (1-\zeta)y_1 + (1+\zeta)y_2\right] \leq 3, \\ & |y_i| \leq 3 \quad i = 1,2. \end{aligned}$$

Due to the fact that the maximum value of a convex function over a convex set is attained at one of the extreme points [19, Theorem 3.4.7], (SRC) is equivalent to the following problem whose optimal objective value is -2:

$$\begin{aligned} \min \quad & -y_1 - y_2 \\ s.t. \quad & y_1 \leq 1, \\ & y_2 \leq 1, \\ & |y_i| \leq 3, \quad i = 1,2. \end{aligned}$$

To get an upper bound for $Opt(ARC)$, we choose $y_1(\zeta) = \frac{3}{2}(1+\zeta)$ and $y_2(\zeta) = \frac{3}{2}(1-\zeta)$ as a decision rule, and it is easy to check the feasibility of $(y_1(\zeta), y_2(\zeta))$. Hence, an upper bound for $Opt(ARC)$ is

$$\max_{\zeta\in[-1,1]} -y_1(\zeta) - y_2(\zeta) \ = \ \max_{\zeta\in[-1,1]} -3 \ = \ -3.$$

So, $Opt(ARC) \leq -3 < -2 = Opt(SRC)$. □

Example 5.10 ***(Assumption vii)*** *Consider the problem*

$$
\begin{aligned}
\min \quad & t \\
\text{s.t.} \quad & |y_1| \le t, \\
& |y_2| \le t, \\
& -(y_1-\zeta_1)^2 - (y_2-\zeta_1)^2 \le -4 - 2\zeta_1^2, \\
& -(y_1-\zeta_2)^2 - (y_2-\zeta_2)^2 \le -4 - 2\zeta_2^2, \\
& |y_i| \le 5, \quad i = 1,2,
\end{aligned}
\tag{5.41}
$$

where $\zeta_1 \in [-1,2]$ and $\zeta_2 \in [-2,1]$ are the uncertain parameters and $y = (y_1, y_2)$ is an adjustable variable. It is easy to check that (5.41) is concave (and, more precisely, it is linear) in the uncertain parameter ζ, and the uncertainty is constraint-wise. Also, $\mathcal{Z} = [-1,2] \times [-2,1]$ and $\mathcal{Y} = [-5,5] \times [-5,5]$ are convex and compact. However, the problem is not convex in the adjustable variable $y = (y_1, y_2)$. The (SRC) corresponding to (5.41) is equivalent to

$$
\begin{aligned}
\min \quad & \|y\|_\infty \\
\text{s.t.} \quad & (y_1+1)^2 + (y_2+1)^2 \ge 6, \\
& (y_1-2)^2 + (y_2-2)^2 \ge 12, \\
& (y_1-2)^2 + (y_2+2)^2 \ge 12, \\
& (y_1+1)^2 + (y_2-1)^2 \ge 6, \\
& |y_i| \le 5, \quad i = 1,2.
\end{aligned}
\tag{5.42}
$$

It is easy to verify that the optimal solution is $y_1 = -\frac{2+\sqrt{14}}{5} \approx -1.15$ and $y_2 = \frac{5+\sqrt{127+6\sqrt{14}}}{5} \approx 3.44$, with the approximated objective value 3.44 for the problem. We choose

$$
y_1(\zeta) = \begin{cases} -1.7, & \zeta_2 \le 0.3 \\ 1.6, & o.w. \end{cases} , \quad y_2(\zeta) = \begin{cases} 2.2, & \zeta_2 \le 0.3 \\ -1.6, & o.w. \end{cases}
$$

as a decision rule to find an upper bound for Opt(ARC). The feasibility of the decision rule can be easily checked, and it implies that $Opt(ARC) \le 2.2 < 3.44 \approx Opt(SRC)$. □

5.6 Conclusion

In this chapter, we show that for some classes of constraint-wise uncertain optimization problems, the static robust optimal solution is also optimal for adjustable robust problems. One class consists of problems that are convex with respect to the adjustable variables and concave with respect to the uncertain parameters, and that have a convex compact uncertainty set and adjustable variables that lie in a convex compact set. Moreover, we provide different examples to show that the assumptions are tight.

This result does not hold for problems where just some of the uncertain parameters are constraint-wise. We prove that under sets of assumptions similar to the pure constraint-wise case, there exists an optimal decision rule that does not depend on the constraint-wise uncertain parameters. Also, we show that for one class of problems, restricting decision rules to be affine and independent of the constraint-wise uncertain parameters yields the same optimal objective value as in cases where the decision rules are affine and dependent on both the constraint-wise and non-constraint-wise uncertain parameters.

Lastly, we apply our results to several classes of problems, such as facility location, inventory system, specific two-stage LO, convex quadratic, and conic quadratic problems.

List of notation

Notation	Description
$\binom{n}{m}$	$\frac{n!}{(n-m)!m!}$
$[n]$	$\begin{cases} \{1,...,n\} & n \neq 0 \\ \emptyset & n = 0 \end{cases}$
$\mathbb{R}^n$	n-dimensional Euclidean vector space
$\mathbb{R}^{n\times m}$	space of $n \times m$ real matrices
S_n	space of $n \times n$ real symmetric matrices
S_n^+	cone of $n \times n$ real positive semi-definite symmetric matrices
$\mathbb{N}^n$	n-tuples of nonnegative integers
$\mathbb{N}^n_d$	$\left\{\alpha \in \mathbb{N}^n : \sum_{i\in[n]} \alpha_i \le d\right\}$
$\hat{\mathbb{N}}^\ell_d$	$\left\{(\alpha,\beta) \in \mathbb{N}^{2m} : \alpha_j = \beta_j = 0 \text{ if } j \notin \mathcal{C}_\ell,\ \sum_{j\in[m]} \alpha_j + \beta_j \le d\right\}$, given $m \in \mathbb{N}$ and $\mathcal{C}_\ell \subseteq [m]$
$\bar{\mathbb{N}}^{\mathcal{D}}_d$	$\left\{\alpha \in \mathbb{R}^n : \alpha_i = 0 \text{ if } i \notin \mathcal{D},\ \sum_{i\in[n]} \alpha_i \le d\right\}$, given $n \in \mathbb{N}$
$\Delta(n,\tau)$	$\left\{x \in \mathbb{R}^n \,\middle|\, \tau x \in \mathbb{N}^n,\ \sum_{i\in[n]} x_i \le 1\right\}$
$\|x\|_2$	$\sqrt{x_1^2 + ... + x_n^2}$, where $x \in \mathbb{R}^n$
$\|x\|_\infty$	$\max_{i\in[n]} |x_i|$, where $x \in \mathbb{R}^n$
$\operatorname{trace}(A)$	summation of diagonal entries of a square matrix A
$\operatorname{rank}(A)$	number of linearly independent columns/rows of the matrix A
$\operatorname{rank}(M)$	number of elements in a basis of the linear space M
$\operatorname{vec}(A)$	$[A_{11},...,A_{1m},...,A_{n1},...,A_{nm}]^T$, where $A \in \mathbb{R}^{n\times m}$
$\operatorname{svec}(A)$	$[A_{11}, \sqrt{2}A_{12},...,\sqrt{2}A_{1n}, A_{22}, \sqrt{2}A_{23},...,\sqrt{2}A_{(n-1)n}, A_{nn}]^T$, where A is a symmetric matrix
$\operatorname{Range}(A)$	$\{Ax : x \in \mathbb{R}^m\}$, where $A \in \mathbb{R}^{n\times m}$

$A \succeq B$	$A - B \in S_n^+$, where $A, B \in \mathbb{R}^{n\times n}$
$A \succ B$	$A - B$ is a symmetric and positive definite matrix
$\lVert A\rVert_F$	$\sqrt{\sum_{i\in[n]}\sum_{j\in[m]} A_{ij}^2}$, where $A \in \mathbb{R}^{n\times m}$
$\lVert A\rVert_1$	$\sum_{i\in[n]}\sum_{j\in[m]} \lvert A_{ij}\rvert$, where $A \in \mathbb{R}^{n\times m}$
$\lVert A\rVert_\infty$	$\max_{\substack{i\in[n]\\ j\in[m]}} \lvert A_{ij}\rvert$, where $A \in \mathbb{R}^{n\times m}$
$\lVert A\rVert_{2,2}$	$\sup_{\lVert x\rVert_2=1} \lVert Ax\rVert_2$, where $A \in \mathbb{R}^{n\times m}$
$\lVert A\rVert_\Sigma$	summation of the singular values of A
$\mathbb{R}[x]$	space of polynomials in x
$\Sigma[x]_d$	cone of sum-of-squares polynomials of degree at most $2d$
$h_{\alpha\beta}(x)$	$\prod_{j\in[m]} g_j(x)^{\alpha_j}\left(1 - g_j(x)\right)^{\beta_j}$, given $m \in \mathbb{N}$, and $g_j(x)$, $j \in [m]$
$v_d(x)$	a vector with a basis for the ring of polynomials in x with degree at most d
$\mathbb{R}[x;\mathcal{D}]$	ring of polynomials in $\{x_i : i \in \mathcal{D}\}$
$\Sigma[x;\mathcal{D}]_d$	cone of sum-of-squares polynomials in $\{x_i : i \in \mathcal{D}\}$ of degree at most $2d$
$\deg(p)$	degree of the polynomial $p(x)$
$\delta^*_{\mathcal{Z}}(u)$	$\sup_{b\in\mathcal{Z}}\left\{u^T b\right\}$, where $\mathcal{Z} \subseteq \mathbb{R}^n$
$\delta^*_{\mathcal{Z}}(W)$	$\sup_{A\in\mathcal{Z}}\left\{\mathrm{trace}\left(AW^T\right)\right\}$, where $\mathcal{Z} \subseteq \mathbb{R}^{m\times n}$
$\delta^*_{\mathcal{Z}}(W,u)$	$\sup_{(A,b)\in\mathcal{Z}}\left\{\mathrm{trace}\left(AW^T\right) + u^T b\right\}$, where $\mathcal{Z} \subseteq \mathbb{R}^{m\times n} \times \mathbb{R}^n$
$\mathrm{adj}(v)$	set of vertices adjacent to v in a given graph
$\mathcal{K}_G$	set of maximal cliques of the graph G

Acronyms

ARC adjustable robust counterpart. 114–121, 128–137

ARO adjustable robust optimization. 8–12, 14, 16, 111–113

BSOS bounded degree sum of squares. 5–7, 13, 15, 20–24, 28, 30, 34, 36, 37, 39–42, 46, 48–53, 67, 68, 70, 72, 73

DTOC discrete-time optimal control. 50, 51, 71, 72

HARC hybrid adjustable robust counterpart. 121–124, 133, 134

HRC hybrid robust counterpart. 121, 123, 124

LMI linear matrix inequality. 21, 89

LO linear optimization. 12, 22, 41–43, 49, 113, 125, 127, 128, 138

PO polynomial optimization. 5, 13, 15, 19–21, 24, 49–54, 57, 61, 63, 66

PSD positive semi-definite. 1, 5, 21, 22, 24, 34, 51, 54, 58–60, 80

QCQO quadratically constrained quadratic optimization. 1–3, 5, 7, 11–15, 30, 49

RIP running intersection property. 49, 51, 71

SBSOS sparse bounded degree sum of squares. 68, 70, 72

SDO semi-definite optimization. 15, 20–24, 49, 50

SOCO second-order cone optimization. 1

SOS sum-of-squares. 5, 22, 52, 53

SRC static robust counterpart. 77–79, 81, 82, 84, 87, 92, 114–116, 118–121, 128–137

SRO static robust optimization. 8–12, 14, 16, 77, 111–113

Index

Bibliography

[1] N. Adhya, M. Tawarmalani, and N. V. Sahinidis. A Lagrangian approach to the pooling problem. *Industrial & Engineering Chemistry Research*, 38(5):1956–1972, 1999.

[2] I. Adler and F. Alizadeh. *Primal-dual interior point algorithms for convex quadratically constrained and semidefinite optimization problems.* Rutgers University. Rutgers Center for Operations Research [RUTCOR], 1995.

[3] A. A. Ahmadi and A. Majumdar. Some applications of polynomial optimization in operations research and real-time decision making. *Optimization Letters*, 10(4):709–729, 2016.

[4] F. A. Al-Khayyal, Ch. Larsen, and T. Van Voorhis. A relaxation method for nonconvex quadratically constrained quadratic programs. *Journal of Global Optimization*, 6(3):215–230, 1995.

[5] Kh. Y. Al-Qahtani and A. Elkamel. *Planning and integration of refinery and petrochemical operations.* John Wiley & Sons, 2011.

[6] M. Alfaki. Models and solution methods for the pooling problem. *Ph.D. thesis, University of Bergen*, 2012.

[7] M. Alfaki and D. Haugland. Strong formulations for the pooling problem. *Journal of Global Optimization*, 56(3):897–916, 2013.

[8] M. Alfaki and D. Haugland. A cost minimization heuristic for the pooling problem. *Annals of Operations Research*, 222(1):73–87, 2014.

[9] P. R. Amestoy, T. A. Davis, and I. S. Duff. An approximate minimum degree ordering algorithm. *SIAM Journal on Matrix Analysis and Applications*, 17(4):886–905, 1996.

[10] E. D. Andersen, C. Roos, and T. Terlaky. On implementing a primal-dual interior-point method for conic quadratic optimization. *Mathematical Programming*, 95(2):249–277, 2003.

[11] K. M. Anstreicher. On convex relaxations for quadratically constrained quadratic programming. *Mathematical Programming*, 136(2):233–251, 2012.

[12] MOSEK ApS. The mosek optimization tools version 3.2 (revision 8) user's manual and reference, 2002.

[13] A. Ardestani-Jaafari and E. Delage. The value of flexibility in robust location–transportation problems. *Transportation Science*, 2017.

[14] K. E. Atkinson. *An introduction to numerical analysis*. John Wiley, 1989.

[15] A. Aubry, A. De Maio, M. Piezzo, and A. Farina. Radar waveform design in a spectrally crowded environment via nonconvex quadratic optimization. *IEEE Transactions on Aerospace and Electronic Systems*, 50(2):1138–1152, 2014.

[16] T. E. Baker and L. S. Lasdon. Successive linear programming at Exxon. *Management Science*, 31(3):264–274, 1985.

[17] R. Baltean-Lugojan and R. Misener. Piecewise parametric structure in the pooling problem–from sparse strongly-polynomial solutions to NP-hardness. *Optimization Online*, 2017. http://www.optimization-online.org/DB_FILE/2016/05/5457.pdf.

[18] O. Baron, J. Milner, and H. Naseraldin. Facility location: A robust optimization approach. *Production and Operations Management*, 20(5):772–785, 2011.

[19] M. S. Bazaraa, H. D. Sherali, and Ch. M. Shetty. *Nonlinear programming: theory and algorithms*. John Wiley & Sons, 2013.

[20] A. Beck and Y. C. Eldar. Strong duality in nonconvex quadratic optimization with two quadratic constraints. *SIAM Journal on Optimization*, 17(3):844–860, 2006.

[21] A. Ben-Tal, D. den Hertog, A. De Waegenaere, B. Melenberg, and G. Rennen. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 59(2):341–357, 2013.

[22] A. Ben-Tal, D. den Hertog, and J.-Ph. Vial. Deriving robust counterparts of nonlinear uncertain inequalities. *Mathematical Programming*, 149(1-2):265–299, 2015.

[23] A. Ben-Tal, G. Eiger, and V. Gershovitz. Global minimization by reducing the duality gap. *Mathematical Programming*, 63(1-3):193–212, 1994.

[24] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust optimization*. Princeton University Press, 2009.

[25] A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski. Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99(2):351–376, 2004.

[26] A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805, 1998.

[27] A. Ben-Tal and A. Nemirovski. Robust solutions of uncertain linear programs. *Operations Research Letters*, 25(1):1–13, 1999.

[28] A. Ben-Tal and A. Nemirovski. *Lectures on modern convex optimization*. SIAM, 2001.

[29] A. Ben-Tal, A. Nemirovski, and C. Roos. Robust solutions of uncertain quadratic and conic-quadratic problems. *SIAM Journal on Optimization*, 13(2):535–560, 2002.

[30] A. Ben-Tal and M. Teboulle. Hidden convexity in some nonconvex quadratically constrained quadratic programming. *Mathematical Programming*, 72(1):51–63, 1996.

[31] A. Berman and N. Shaked-Monderer. *Completely positive matrices*. World Scientific, 2003.

[32] D. Bertsimas and H. Bidkhori. On the performance of affine policies for two-stage adaptive optimization: a geometric perspective. *Mathematical Programming*, 153(2):577–594, 2015.

[33] D. Bertsimas and V. Goyal. On the power and limitations of affine policies in two-stage adaptive optimization. *Mathematical Programming*, 134(2):491–531, 2012.

[34] D. Bertsimas and V. Goyal. On the approximability of adjustable robust convex optimization under uncertainty. *Mathematical Methods of Operations Research*, 77(3):323–343, 2013.

[35] D. Bertsimas, V. Goyal, and B. Y. Lu. A tight characterization of the performance of static solutions in two-stage adjustable robust linear optimization. *Mathematical Programming*, 150(2):281–319, 2015.

[36] D. Bertsimas, V. Gupta, and N. Kallus. Data-driven robust optimization. *Mathematical Programming*, pages 1–58, 2017.

[37] D. Bertsimas, D. A. Iancu, and P. A. Parrilo. Optimality of affine policies in multistage robust optimization. *Mathematics of Operations Research*, 35(2):363–394, 2010.

[38] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, 2017.

[39] D. Bienstock and A. Michalka. Polynomial solvability of variants of the trust-region subproblem. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 380–390. SIAM, 2014.

[40] J. Bisschop. *AIMMS optimization modeling.* Paragon Decision Technology, Bellevue, 2006.

[41] J. R. S. Blair and B. Peyton. An introduction to chordal graphs and clique trees. In *Graph theory and sparse matrix computation*, pages 1–29. Springer, 1993.

[42] Ch. E. Bodington and T. E. Baker. A history of mathematical programming in the petroleum industry. *Interfaces*, 20(4):117–127, 1990.

[43] O. Boni and A. Ben-Tal. Adjustable robust counterpart of conic quadratic problems. *Mathematical Methods of Operations Research*, 68(2):211–233, 2008.

[44] S. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge University Press, 2004.

[45] Th. F. Coleman and A. Liao. An efficient trust region method for unconstrained discrete-time optimal control problems. *Computational Optimization and Applications*, 4(1):47–66, 1995.

[46] A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust region methods.* SIAM, 2000.

[47] R. W. Cottle, J. Pang, and R. E. Stone. *The linear complementarity problem.* SIAM, 2009.

[48] J. Czyzyk, M. P. Mesnier, and J. J. Moré. The NEOS server. *IEEE Journal on Computational Science and Engineering*, 5(3):68 – 75, 1998.

[49] E. de Klerk. The complexity of optimizing over a simplex, hypercube or sphere: a short survey. *Central European Journal of Operations Research*, 16(2):111–125, 2008.

[50] E. Delage and Y. Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations Research*, 58(3):595–612, 2010.

[51] S. S. Dey and A. Gupte. Analysis of MILP techniques for the pooling problem. *Operations Research*, 63(2):412–427, 2015.

[52] G. Di Pillo, L. Grippo, and F. Lampariello. A class of structured quasi-newton algorithms for optimal control problems. *IFAC Proceedings Volumes*, 16(8):101–107, 1983.

[53] A. Drud. CONOPT: A system for large scale nonlinear optimization. *Reference manual for CONOPT subroutine library, ARKI Consulting and Development A/S, Bagsvaerd, Denmark*, 1996.

[54] L. El Ghaoui and H. Lebret. Robust solutions to least-squares problems with uncertain data. *SIAM Journal on Matrix Analysis and Applications*, 18(4):1035–1064, 1997.

[55] F. J. Fabozzi, P. N. Kolm, D. Pachamanova, and S. M. Focardi. *Robust portfolio optimization and management.* John Wiley & Sons, 2007.

[56] M. Fazel, H. Hindi, and S. P. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*, volume 6, pages 4734–4739. IEEE, 2001.

[57] U. Feige, K. Jain, M. Mahdian, and V. Mirrokni. Robust combinatorial optimization with exponential scenarios. In Matteo Fischetti and David P. Williamson, editors, *Integer Programming and Combinatorial Optimization*, volume 4513 of *Lecture Notes in Computer Science*, pages 439–453. Springer Berlin Heidelberg, 2007.

[58] L. R. Foulds, D. Haugland, and K. Jörnsten. A bilinear approach to the pooling problem. *Optimization*, 24(1-2):165–180, 1992.

[59] T. L. Friesz. *Nonlinear Programming and Discrete-Time Optimal Control*, pages 33–78. Springer US, 2010.

[60] L. Frimannslund, M. El Ghami, M. Alfaki, and D. Haugland. Solving the pooling problem with LMI relaxations. *S. Cafieri, B.G.-Tóth, E. Hendrix, L. Liberti and F. Messine (Eds), Proceedings of the Toulouse Global Optimization Workshop (TOGO)*, pages 51–54, 2010.

[61] M. Fukuda, M. Kojima, K. Murota, and K. Nakata. Exploiting sparsity in semidefinite programming via matrix completion I: General framework. *SIAM Journal on Optimization*, 11(3):647–674, 2001.

[62] A. Gharanjik, B. Shankar, M. Soltanalian, and B. Ottersten. An iterative approach to nonconvex QCQP with applications in signal processing. In *Sensor Array and Multichannel Signal Processing Workshop (SAM), 2016 IEEE*, pages 1–5. IEEE, 2016.

[63] D. Goldfarb and G. Iyengar. Robust convex quadratically constrained programs. *Mathematical Programming*, 97(3):495–515, 2003.

[64] D. Goldfarb and G. Iyengar. Robust portfolio selection problems. *Mathematics of Operations Research*, 28(1):1–38, 2003.

[65] B. L. Gorissen and D. den Hertog. Robust nonlinear optimization via the dual. *Optimization Online*, 2015. http://www.optimization-online.org/DB_FILE/2015/04/4886.pdf.

[66] B. L. Gorissen, İ. Yanıkoğlu, and D. den Hertog. A practical guide to robust optimization. *Omega*, 53:124–137, 2015.

[67] N. Gülpınar and B. Rustem. Worst-case robust decisions for multi-period mean–variance portfolio optimization. *European Journal of Operational Research*, 183(3):981–1000, 2007.

[68] A. Gupte, Sh. Ahmed, S. S. Dey, and M. S. Cheon. Relaxations and discretizations for the pooling problem. *Journal of Global Optimization*, pages 1–39, 2016.

[69] D. Haugland. The computational complexity of the pooling problem. *Journal of Global Optimization*, 64(2):199–215, 2016.

[70] C. A. Haverly. Studies of the behavior of recursion for the pooling problem. *ACM SIGMAP Bulletin*, pages 19–28, 1978.

[71] N. J. Higham. *Accuracy and stability of numerical algorithms*. SIAM, 2002.

[72] R. A. Horn and Ch. R. Johnson. *Matrix analysis*. Cambridge University Press, 2012.

[73] Y. Huang, A. De Maio, and Sh. Zhang. Semidefinite programming, matrix decomposition, and radar code design. *Convex Optimization in Signal Processing and Communications*, pages 192–228, 2010.

[74] D. A. Iancu, M. Sharma, and M. Sviridenko. Supermodularity and affine policies in dynamic robust optimization. *Operations Research*, 61(4):941–956, 2013.

[75] D. A. Iancu and N. Trichakis. Pareto efficiency in robust optimization. *Management Science*, 60(1):130–147, 2013.

[76] V. Jeyakumar, G. Li, and J. Vicente-Pérez. Robust SOS-convex polynomial optimization problems: exact SDP relaxations. *Optimization Letters*, 9(1):1–18, 2015.

[77] R. Jiang and D. Li. Convex relaxations with second order cone constraints for nonconvex quadratically constrained quadratic programming. *arXiv:1608.02096*, 2016. https://arxiv.org/pdf/1608.02096.pdf.

[78] N. L. Johnson, S. Kotz, and N. Balakrishnan. *Continuous Univariate Distributions*, volume 1. John Wiley, 2nd edition, 1994.

[79] R. M. Karp. *Reducibility among Combinatorial Problems*, pages 85–103. The IBM Research Symposia Series. Springer, Boston, MA, 1972.

[80] R. Karuppiah and I. E. Grossmann. Global optimization for the synthesis of integrated water systems in chemical processes. *Computers & Chemical Engineering*, 30(4):650–673, 2006.

[81] S. Kim, M. Kojima, and H. Waki. Generalized lagrangian duals and sums of squares relaxations of sparse polynomial optimization problems. *SIAM Journal on Optimization*, 15(3):697–719, 2005.

[82] A. Konar and N. D. Sidiropoulos. Hidden convexity in QCQP with toeplitz-hermitian quadratics. *IEEE Signal Processing Letters*, 22(10):1623–1627, 2015.

[83] S. Kotz, N. Balakrishnan, and N. L. Johnson. *Continuous multivariate distributions*, volume 1. John Wiley, 2nd edition, 2000.

[84] J. L. Krivine. Anneaux préordonnés. *Journal d'Analyse Mathématique*, 12(1):307–326, 1964.

[85] J. B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.

[86] J. B. Lasserre. Polynomial programming: LP-relaxations also converge. *SIAM Journal on Optimization*, 15(2):383–393, 2005.

[87] J. B. Lasserre. A lagrangian relaxation view of linear and semidefinite hierarchies. *SIAM Journal on Optimization*, 23(3):1742–1756, 2013.

[88] J. B. Lasserre, K. Toh, and Sh. Yang. A bounded degree SOS hierarchy for polynomial optimization. *EURO Journal on Computational Optimization*, 5(1):87–117, 2017.

[89] M. Laurent. Sums of squares, moment matrices and optimization over polynomials. In *Emerging applications of algebraic geometry*, pages 157–270. Springer, 2009.

[90] T. Lipp and S. Boyd. Variations and extension of the convex–concave procedure. *Optimization and Engineering*, 17(2):263–287, 2016.

[91] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra and Its Applications*, 284(1-3):193–228, 1998.

[92] J. Lofberg and P. A. Parrilo. From coefficients to samples: a new approach to SOS optimization. In *Proc. 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*, volume 3, pages 3154–3159. IEEE, 2004.

[93] A. Marandi, J. Dahl, and E. de Klerk. A numerical evaluation of the bounded degree sum-of-squares hierarchy of Lasserre, Toh, and Yang on the pooling problem. *Annals of Operations Research*, pages 1–26, 2017.

[94] H. Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952.

[95] M. Melamed, A. Ben-Tal, and B. Golany. On the average performance of the adjustable RO and its use as an offline tool for multi-period production planning under uncertainty. *Computational Management Science*, 13(2):293–315, 2016.

[96] R. Misener and Ch. A. Floudas. Advances for the pooling problem: Modeling, global optimization, and computational studies. *Applied and Computational Mathematics*, 8(1):3–22, 2009.

[97] R. Misener, J. P. Thompson, and Ch. A. Floudas. APOGEE: Global optimization of standard, generalized, and extended pooling problems via linear and logarithmic partitioning schemes. *Computers & Chemical Engineering*, 35(5):876–892, 2011.

[98] J. J. Moré and G. Toraldo. On the solution of large quadratic programming problems with bound constraints. *SIAM Journal on Optimization*, 1(1):93–113, 1991.

[99] K. G. Murty and F. Yu. *Linear complementarity, linear and nonlinear programming*. Berlin: Heldermann, 1988.

[100] Y. Nesterov and A. Nemirovskii. *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.

[101] J. Neter, M. H. Kutner, Ch. J. Nachtsheim, and W. Wasserman. *Applied linear statistical models*, volume 4. Irwin Chicago, 1996.

[102] R. A. Nicolaides. On a class of finite elements generated by lagrange interpolation. *SIAM Journal on Numerical Analysis*, 9(3):435–445, 1972.

[103] I. Nielsen. *Structure-Exploiting Numerical Algorithms for Optimal Control*. PhD thesis, Linköping University, 2017.

[104] D. P. O'Leary. A generalized conjugate gradient algorithm for solving a class of quadratic programming problems. *Linear Algebra and Its Applications*, 34:371–399, 1980.

[105] P. M. Pardalos and S. A. Vavasis. Quadratic programming with one negative eigenvalue is NP-hard. *Journal of Global Optimization*, 1(1):15–22, 1991.

[106] J. Park and S. Boyd. General heuristics for nonconvex quadratically constrained quadratic programming. *arXiv:1703.07870*, 2017. https://arxiv.org/pdf/1703.07870.pdf.

[107] J. Peng, C. Roos, and T. Terlaky. Primal-dual interior-point methods for second-order conic optimization based on self-regular proximities. *SIAM Journal on Optimization*, 13(1):179–203, 2002.

[108] I. Pólik and T. Terlaky. A survey of the S-lemma. *SIAM Review*, 49(3):371–418, 2007.

[109] B. Ran and D. Boyce. Discrete optimal control and nonlinear programming. In *Modeling Dynamic Transportation Networks*, pages 49–67. Springer, 1996.

[110] J. H. Rife. The effect of uncertain covariance on a chi-square integrity monitor. *Navigation*, 60(4):291–303, 2013.

[111] N. V. Sahinidis. *BARON 14.3.1: Global Optimization of Mixed-Integer Nonlinear Programs,* User's Manual, 2014.

[112] J. Shawe-Taylor and N. Cristianini. Estimating the moments of a random vector with applications. *Proceedings GRETSI 2003 Conference*, pages 1173–1178, 2003.

[113] J. F. Sturm and Sh. Zhang. On cones of nonnegative quadratic functions. *Mathematics of Operations Research*, 28(2):246–267, 2003.

[114] A. Takeda, Sh. Taguchi, and R. H. Tütüncü. Adjustable robust optimization models for a nonlinear two-period system. *Journal of Optimization Theory and Applications*, 136(2):275–295, 2008.

[115] M. Tawarmalani and N. V. Sahinidis. *Convexification and global optimization in continuous and mixed-integer nonlinear programming: theory, algorithms, software, and applications*, volume 65. Springer Science & Business Media, 2002.

[116] L. Trapani. Testing for (in)finite moments. *Journal of Econometrics*, 191(1):57–68, 2016.

[117] A. W. Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge University Press, 2000.

[118] L. Vandenberghe and M. S. Andersen. *Chordal Graphs and Semidefinite Optimization*, volume 1 of *Foundations and trends in optimization*. Now Publishers, 2015.

[119] H. Waki, S. Kim, M. Kojima, and M. Muramatsu. Sums of squares and semidefinite program relaxations for polynomial optimization problems with structured sparsity. *SIAM Journal on Optimization*, 17(1):218–242, 2006.

[120] T. Weisser, J. B. Lasserre, and K. Toh. Sparse-BSOS: a bounded degree SOS hierarchy for large scale polynomial optimization with sparsity. *Mathematical Programming Computation*, pages 1–32, 2017.

[121] Zh. Y. Wu, D. Li, L. Sh. Zhang, and X. Yang. Peeling off a nonconvex cover of an actual convex problem: hidden convexity. *SIAM Journal on Optimization*, 18(2):507–536, 2007.

[122] Y. Ye and Sh. Zhang. New results on quadratic minimization. *SIAM Journal on Optimization*, 14(1):245–267, 2003.

[123] X. J. Zheng, X. L. Sun, and D. Li. Nonconvex quadratically constrained quadratic programming: best DC decompositions and their SDP representations. *Journal of Global Optimization*, 50(4):695–712, 2011.