



**HAL**  
open science

# Nonlinear Adjustable Robust Optimization

Abbas Khademi

► **To cite this version:**

Abbas Khademi. Nonlinear Adjustable Robust Optimization. Mathematics [math]. PhD thesis, University of Tehran, 2024. English. NNT: . tel-04460774

**HAL Id: tel-04460774**

**<https://hal.science/tel-04460774v1>**

Submitted on 15 Feb 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright



UNIVERSITY OF TEHRAN

SCHOOL OF MATHEMATICS, STATISTICS AND COMPUTER SCIENCE

DOCTORAL THESIS

# Nonlinear Adjustable Robust Optimization

*Author:*

Abbas Khademi

*Supervisor:*

Prof. Majid Soleimani-damaneh  
(UNIVERSITY OF TEHRAN)

*Advisor:*

Dr. Ahmadreza Marandi  
(EINDHOVEN UNIVERSITY OF TECHNOLOGY)

A THESIS SUBMITTED IN FULFILLMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY  
IN APPLIED MATHEMATICS

FEBRUARY 6, 2024

UNIVERSITY OF TEHRAN  
COLLEGE OF SCIENCE  
SCHOOL OF MATHEMATICS, STATISTICS AND COMPUTER SCIENCE

The undersigned hereby certify that they have read and recommend to the School of Mathematics, Statistics and Computer Science for acceptance a thesis entitled “**Nonlinear Adjustable Robust Optimization**” by **Abbas Khademi** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**.

Dated: February 6, 2024

Research Supervisor: Prof. Majid Soleimani-damaneh (University of Tehran)

Research Advisor: Dr. Ahmadreza Marandi (Eindhoven University of Technology)

Examining Committee: Prof. Esmail Khorram (Amirkabir University of Technology)

Prof. Mohammad Bagher Asadi (University of Tehran)

Dr. Alireza Hosseini (University of Tehran)

Dr. Javad Koushki (K. N. Toosi University of Technology)

## Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

---

Abbas Khademi

February 6, 2024

## Abstract

Robust Optimization (RO) is a popular approach for dealing with uncertain data in optimization. In static robust optimization, decision variables represent here-and-now decisions made without exact knowledge of uncertain parameters but must be feasible when the actual data is within the uncertainty set. However, one mechanism to overcome the limitations of the static RO approach is Adjustable Robust Optimization (ARO), which leverages adaptability. The main difference between static RO and ARO approaches is the decision-making manner. In ARO problems, some variables are here-and-now decisions, while others are wait-and-see decisions made later based on the observed parameters in the uncertainty set.

In this PhD thesis, we address two main topics in mathematical optimization. The first topic concerns a class of nonlinear ARO problems with uncertainty in the objective function and constraints. By utilizing Fenchel's duality, we derive an equivalent dual reformulation that is a nonlinear static robust optimization problem. We then apply perspective relaxation and an alternating method to handle non-concavity and design a new dual-based cutting plane algorithm that can find a reasonable lower bound for the optimal objective value. Through numerical experiments, we show the effectiveness of the cutting plane algorithm in producing locally robust solutions with an acceptable optimality gap.

The second topic focuses on the reformulation of quadratic optimization problems using ARO. Quadratic Optimization (QO) has been extensively studied in the literature due to its practical applicability in numerous problems. Despite its practicality, QO problems are generally NP-hard. Consequently, researchers have developed various numerical methods for finding approximate optimal solutions. In this thesis, we analyze QO problems through the lens of robust optimization techniques. We first demonstrate that any QO problem can be reformulated as a disjoint bi-convex QO problem. Subsequently, we present an equivalent ARO reformulation and utilize some methods from the relevant literature to approximate this reformulation. Specifically, we show that employing a so-called decision rule technique to approximate the ARO reformulation can be interpreted as applying a linearization-relaxation technique to its bi-convex reformulation problem. Additionally, we have designed an algorithm capable of finding a solution that is close to optimal based on our new reformulations. Our numerical results highlight the efficiency of our algorithm, particularly for large-sized instances, in comparison with standard off-the-shelf solvers. This work offers a novel perspective on quadratic problems and paves the way for further research in this domain of mathematical optimization.

**Keywords:** Adjustable Robust Optimization, Fenchel Duality, Biconvex Programming, Perspective Function, Cutting Plane Methods, Quadratic Optimization, Decision Policy, Linearization-Relaxation Technique.

**TO THOSE WHO DID NOT AND DO NOT HAVE  
EDUCATIONAL FACILITIES AND COULD NOT STUDY**

---

Abbas Khademi  
February 6, 2024

## Acknowledgments

Writing a thesis is certainly not a solitary endeavor, but rather a collaborative effort involving numerous individuals who have contributed to its triumphant culmination. In this part, I am fortunate to have the opportunity to express my gratitude to these people.

First and foremost, I would like to express my deepest appreciation to my supervisor, Professor Majid Soleimani-damaneh. His guidance and support throughout the entire research process were invaluable. His intellectual insights, technical expertise, and unwavering commitment to excellence significantly enhanced my research skills and analytical thinking. I have learned a great deal from him, not only in the field of Optimization but also in research ethics and values that are essential in daily life.

I would also like to extend my profound gratitude to Dr. Ahmadreza Marandi for his astute discussions and unwavering support in exploring various research ideas. His guidance as my academic advisor was instrumental in fostering my intellectual curiosity. This thesis celebrates the amicable relationship I share with Ahmadreza, both as a mentor and a friend. I have derived significant intellectual benefits from his perspicacity and discerning approach to research. His constant encouragement and motivational zeal have been an illuminating force in my research endeavors. Additionally, we did regular weekly meetings to discuss ongoing research projects and exchange new ideas, which helped me stay on track and make progress in my research.

I would also thank all my teachers. Each one has taught me valuable lessons and skills that helped me become who I am today. Specially, I would like to express my gratitude to Dr. Morteza Mirdehghan, who introduced me to the fascinating world of Optimization during my undergraduate studies and inspired me to pursue further research in this area. I am also thankful to Dr. Alireza Hosseini and Dr. Mohammad B. Asadi for deepening my knowledge and understanding of the subject matter by teaching some courses during my time as a graduate student.

In addition, I owe a debt of gratitude to my friends, Aliakbar Hosseini and Yaghoob Rahimi, for their insightful scientific discussions and feedback. I would also like to thank M. Reza Modares, M. Ali Asadi-vasfi, and Ava Azarshab for their encouragement. Special mention goes to Moslem Zamani, Hadi Abbaszadeh, and Sahand Asgharieh whose assistance during my stay in the Netherlands was invaluable. My sincere appreciation also goes to the family of the late Prof. Ebrahim Ghasemi-Nejad for their 9-month financial contribution as a scholarship.



I am grateful to the Department of Industrial Engineering and Innovation Sciences for their warm hospitality during my visiting research period at Eindhoven University of Technology.

Teaching at the University of Tehran was an enriching experience that allowed me to feel more connected to the academic community. I am also grateful to all of my students in these years.

Last but not least, I want to express my profound appreciation for my parents' unwavering love, trust, and understanding. Their constant support and encouragement have been a source of strength and motivation throughout my academic journey. In particular, my mother has played an instrumental role in shaping my personality, character, and academic pursuits. To show my heartfelt gratitude, I dedicate this thesis to my mother, who is the strongest woman in the world to me.

In conclusion, I am immensely grateful to all those who have played a part in my academic journey.

ABBAS KHADEMI,  
FEBRUARY 6, 2024,  
UNIVERSITY OF TEHRAN, TEHRAN

## Preface

In light of human advancements in numerous fields and the increasing demand for resources on the one hand, and their limitations on the other hand, the need for optimal use of resources becomes more crucial. Mathematical optimization has become an important part of many decision-making problems in management, economics, medicine, engineering, etc. In classical optimization models, all parameters are considered to be exactly known, resulting in deterministic problems. In real life, however, many of the parameters are not known at the moment of decision-making and have uncertainty in their essence. There are various approaches for dealing with uncertainty in the optimization and mathematical modeling literature. The most commonly used ones are stochastic optimization and robust optimization.

In stochastic optimization, probabilistic information (distribution) on the uncertain parameters is required and the decision-maker aims to optimize expected objective values; for a more detailed description of stochastic optimization, we refer to the textbook [107]. The main disadvantage of stochastic optimization is that the exact distributions of uncertain parameters are often unknown. In contrast, Robust Optimization (RO) does not require any probabilistic information. In RO, the best solution is chosen among those that are safe-guarded against all scenarios in a pre-specified set, called uncertainty set.

The concept of static RO was first proposed by Soyster in the 1970s, who studied a linear optimization problem with a box uncertainty set [111]. Later in the 1990s, static RO was formally introduced [15, 16, 57] and its computational advantage has resulted in its wide usage in applications, including in portfolio selection [79, 121], scheduling [40], operations management [85], etc.

In static RO, all decision variables represent here-and-now decisions, meaning the decisions are made before realization of the uncertain parameters [13]. However, in many practical applications, the value of some decisions can be adjusted after realization of (part of) uncertain parameters. These kinds of decisions are called wait-and-see decisions. Adjustable Robust Optimization (ARO) is an extension of the static RO wherein decision variables are divided into two types: here-and-now and wait-and-see [14]. In recent years, the application of ARO has been widespread in many areas such as network design [127], location-transportation problems [88], facility location problems [56], chemical engineering [73, 82], logistics [74], radiotherapy [103], to name a few.

Although ARO improves solution quality (in the sense of being less conservative), its computational complexity is higher than static RO [123]. Under some circumstances, ARO and static RO are equivalent [87]; therefore, using the static RO

model for some cases can be advantageous. Moreover, as wait-and-see decisions are chosen after realizing uncertain parameters, they can be considered as decision rules or functions of the uncertain parameters. A way to approximate an ARO problem is by restricting wait-and-see variables to be affine in uncertain parameters, and use the so-called affine decision rules (ADR). Under some conditions, ADR can provide a “good” approximation of an ARO problem [14, 25]. Next to ADR, there are also other decision rules, like piece-wise constant [100], quadratic [119], and polynomial [29], each has their own advantages and disadvantages. In addition to decision rules, there are several other approximation techniques for solving linear ARO problems in the literature, including Benders decomposition [126], finite adaptability [67], partitioning the uncertainty set [23, 97], and copositive approach [122].

This thesis aims to contribute to the growing literature on robust optimization by proposing novel approaches to solving nonlinear ARO problems and indefinite Quadratic Optimization (QO) problems. Chapter 1 provides the necessary definitions and notations. Chapter 2 briefly goes over the initial idea and birth of the adjustable robustness concepts and reviews some approximations of linear ARO problems. Chapter 3 focuses on a class of nonlinear ARO problems with uncertainty in the objective function and constraints, proposing a cutting plane algorithm to obtain locally robust solutions. Chapter 4 presents a new reformulation technique for indefinite QO problems as ARO, offering new possibilities for solving large-scale problems in the field of mathematical optimization. Finally, Chapter 5 presents conclusions and future research.

## Thesis Outline

### Chapter 1

This chapter covers various definitions and notations, such as extended real-valued functions, proper and closed functions, and convex conjugate functions. It also provides illustrations to explain these concepts and discusses Fenchel duality, which relates a primal convex optimization problem to its dual. Finally, the chapter concludes by discussing two types of robust optimization, static and adjustable, and highlights their differences.

### Chapter 2

This chapter aims to provide a short review of the literature relating to linear ARO problems. We address the limitations of static robust optimization, which assumes that all decisions must be made before uncertainty is realized, and does not apply to real-world problems where some decisions can be delayed until

uncertainty is revealed. This chapter delves into non-decision policy approaches for solving linear ARO. These approaches are presented and examined here to provide a comprehensive understanding of how to handle uncertainty in linear optimization problems effectively.

### Chapter 3

This chapter is focused on a class of nonlinear ARO problems with uncertainty in the objective function and constraints. We apply Fenchel's duality on the wait-and-see variables to obtain an equivalent dual reformulation as a nonlinear static RO problem. We provide conditions for the ARO problem to be convex on the here-and-now decision, and propose perspective relaxation and an alternating method to handle non-concavity. By employing these techniques, we obtain an upper bound and a lower bound for the optimal objective value of the ARO model. We also present convergence analysis and numerical experiments to illustrate the effectiveness of our cutting plane algorithm in producing locally robust solutions with an acceptable optimality gap. Overall, this chapter presents a novel approach to solving nonlinear ARO problems, based on the following paper:

- Khademi, A., Marandi, A., & Soleimani-damaneh, M. (2023). A New Dual-Based Cutting Plane Algorithm for Nonlinear Adjustable Robust Optimization. *Journal of Global Optimization*.

### Chapter 4

The goal of this chapter is to demonstrate that any QO problem can be reformulated initially as a disjoint bi-convex problem and subsequently as an ARO problem with convex quadratic objective functions and constraints, thereby rendering it suitable for ARO techniques. We further reveal that approximating the ARO reformulation using a structured affine decision rule is tantamount to applying relaxation techniques to the bi-convex reformulation. Additionally, we introduce an algorithm to calculate bounds on the optimal value of QO problems, employing a decision-rule approximation to establish a lower bound and a mountain-climbing procedure to enhance the quality of the solutions. We investigate several upper and lower bounds that can be applied to obtain approximate solutions and provide computational results to demonstrate the effectiveness of this method in solving large-scale problems. This chapter is based on the following paper:

- Khademi, A., & Marandi, A. (2024). Quadratic Optimization Through the Lens of Adjustable Robust Optimization. Submitted.

## **Chapter 5**

In this chapter, we summarize our contributions and explore several research directions for the future.

# Contents

<b>List of Figures</b>	xiv
<b>List of Tables</b>	xv
<b>1 Preliminaries</b>	<b>1</b>
1.1 Conjugacy and Perspective Function	1
1.2 Fenchel Duality	6
1.3 Robust Optimization	9
1.4 Adjustable Robust Optimization Approximations	15
1.5 Uncertainty Sets	20
1.6 Convex Quadratic Problems	22
<b>2 Linear Adjustable Robust Optimization</b>	<b>24</b>
2.1 Introduction	24
2.2 Approximation of Linear ARO	27
2.2.1 Dual Formulations	27
2.2.2 Column-and-Constraint Generation	28
2.2.3 Copositive Approach	30
2.3 Conclusion	31
<b>3 A Dual-Based Cutting Plane Algorithm for Nonlinear Adjustable Robust Optimization</b>	<b>32</b>
3.1 Introduction	32
3.2 Dual Reformulation	34
3.3 On Upper Bound Calculation	40
3.4 Lower Bound Calculation	46
3.4.1 Alternating Iterative Algorithm	49
3.4.2 Dual-Based Cutting Plane Algorithm	54
3.5 Numerical Experiments	56
3.5.1 Problem Setting	56
3.5.2 Numerical results	59
3.6 Conclusions	63
<b>4 Quadratic Optimization Through the Lens of Adjustable Robust Optimization</b>	<b>67</b>
4.1 Introduction	67
4.1.1 Notation	71

4.2	New Reformulations for Quadratic Optimization Problems	71
4.3	ARO Based Approximations	77
4.3.1	Decision Rules	77
4.3.2	Fourier–Motzkin Elimination	82
4.3.3	Finite Scenario Approach	83
4.4	Solution Method	85
4.5	Numerical Experiments	86
4.5.1	Concave Quadratic Minimization	87
4.5.2	Standard Quadratic Optimization	91
4.6	Conclusions	97
<b>5</b>	<b>Conclusions and Future Research</b>	<b>103</b>
5.1	Future Directions	104
	<b>References</b>	<b>105</b>

## List of Figures

1.1	Domain and epigraph of an extended real-valued function.	2
1.2	Convex conjugate function.	3
1.3	A proper, closed, and convex function and its convex perspective function.	4
1.4	The robust feasible region.	12
1.5	Adjustable robust optimization. The blue square is the uncertainty set $\mathcal{U}$ .	14
1.6	Affine and piece-wise constant decision policies.	17
1.7	Classical uncertainty sets.	22
3.1	The comparison of the optimality gaps of the solutions obtained by Algorithm 1, Algorithm 2, and finite-scenario approach for the instances in Class One.	61
3.2	The comparison of the solution time of Algorithm 1, Algorithm 2, and finite-scenario approach for the instances in Class One.	62
3.3	The comparison of the optimality gaps of the solutions obtained by Algorithm 1, Algorithm 2, and finite-scenario approach for the instances in Class Two.	63
3.4	The comparison of solution times between Algorithm 1, Algorithm 2, and the finite-scenario approach for instances in Class Two.	64



## List of Tables

1.1	Conjugate functions of some popular forms.	6
1.2	Support functions of some popular sets.	6
1.3	Convex quadratic problems and their corresponding Dorn duals.	23
3.1	Statistic of optimality gaps of instances in Class One.	60
3.2	Statistic of optimality gaps of instances in Class Two.	61
3.3	Detailed numerical results from Class One.	65
3.4	Detailed numerical results from Class Two.	66
4.1	Optimality gaps and solver times of concave minimization instances from [105].	89
4.2	Statistic of optimality gaps and solver times for randomly generated concave minimization instances.	91
4.3	Statistic of solution gaps and solution times of (StQO) instances in Class One.	95
4.4	Statistic of solution gaps and solution times of (StQO) instances in Class Two.	95
4.5	Statistic of solution gaps and solution times of generated large-sized (StQO) instances.	96
4.6	Comparison result on concave quadratic minimization test instances from [105].	98
4.7	Time Results for concave quadratic minimization test instances from [105].	98
4.8	Other classes of ARO reformulations of indefinite QO problem.	99
4.9	Other classes of ARO reformulations of concave QO minimization.	100
4.10	Detailed result on concave quadratic minimization.	101
4.11	Detailed numerical results on standard quadratic optimization.	102

# CHAPTER 1

## Preliminaries

John von Neumann:

*“If people do not believe that mathematics is simple, it is only because they do not realize how complicated life is.”*

In this thesis, we use the following definitions and notations. We first recall some standard terminology from convex analysis.

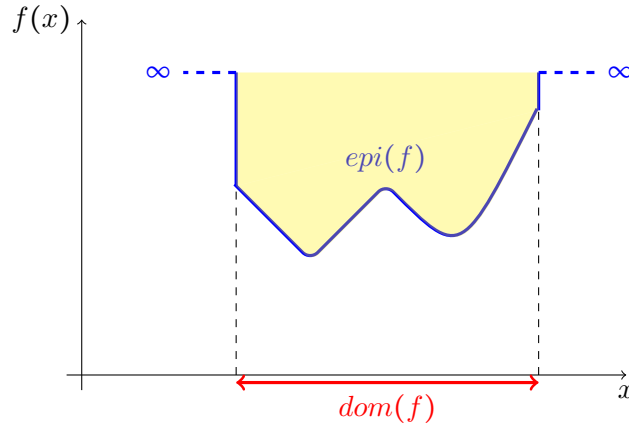
### Notation

We use  $[m]$  to denote the set  $\{1, 2, \dots, m\}$ , and  $[m_0]$  to denote the set  $\{0, 1, \dots, m\}$ . The column vector of all zeros will be denoted by  $\mathbf{0}$ . The sets of all non-negative and all extended-real numbers are denoted by  $\mathbb{R}_+ := [0, \infty)$  and  $\bar{\mathbb{R}} := [-\infty, \infty]$ , respectively. We denote by  $ri(\mathcal{S})$  the relative interior of a set  $\mathcal{S} \subseteq \mathbb{R}^n$ .

### 1.1 Conjugacy and Perspective Function

An extended real-valued function is a function that can adopt the value plus infinity. Suppose that  $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\infty\}$ , the set of points above the graph of an extended real-valued function  $f$  is called the *epigraph* of  $f$ , i.e.,  $epi(f) = \{(x, t) : f(x) \leq t\} \subseteq \mathbb{R}^{n+1}$ . The projection of the epigraph onto the  $x$  space is referred to as the *domain* of  $f$  (see Figure 1.1).

An extended real-valued function  $f$  is *proper* if its domain is not empty, i.e., if  $f(x) > -\infty$  for all  $x \in \mathbb{R}^n$  and  $f(x) < +\infty$  for at least one  $x \in \mathbb{R}^n$ . A function  $f$  is *closed* if its epigraph is a closed set. The epigraph of a proper convex and closed function is convex, closed, non-empty, and contains no vertical line.



**Figure 1.1:** Domain and epigraph of an extended real-valued function.

**Definition 1.1.1** (Conjugate Functions, [99]).<sup>1</sup> The conjugate of a function  $g : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$  is the function  $g^* : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$  defined as  $g^*(y) := \sup_{x \in \mathbb{R}^n} \{y^\top x - g(x)\}$ , where  $y \in \mathbb{R}^n$ .  $\square$

This definition with  $n = 1$  is visually represented in Figure 1.2. For given  $y$ , the highest value of  $yx - g(x)$  is achieved when the derivative of  $g(x)$  equals  $y$ . At this point, the tangent line of  $g$  intersects the  $y$ -axis at the coordinates  $(0, -g^*(y))$  [101].

The *indicator function* of a set  $S \subseteq \mathbb{R}^n$ , denoted by  $\delta_S$ , is defined as

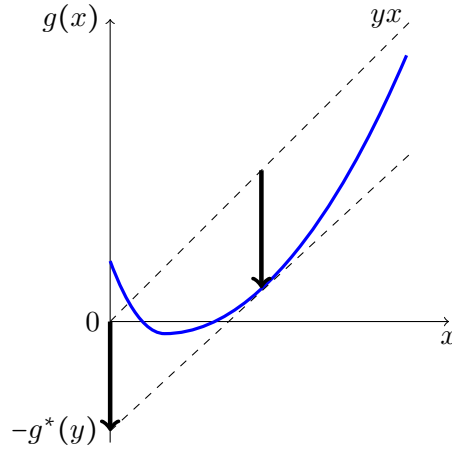
$$\delta_S(x) = \begin{cases} 0, & x \in S, \\ \infty, & x \notin S. \end{cases}$$

The *support function*  $\delta_S^* : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$  of a set  $S \subseteq \mathbb{R}^n$  is  $\delta_S^*(y) := \sup_{x \in S} \{y^\top x\}$ , where  $y \in \mathbb{R}^n$ . It is worth mentioning that the support function corresponding to  $S$  is the conjugate of  $\delta_S$ .

**Remark 1.1.1** (Support Function and Conjugacy). The conjugate function of  $g$  could be expressed by using support function of  $\text{epi}(g)$  as  $g^*(y) = \delta_{\text{epi}(g)}^*(y, -1)$ .  $\square$

**Definition 1.1.2** (Perspective Functions, [99, 129]). The convex perspective of a proper,

<sup>1</sup>This definition also known as *convex conjugate function* in the literature, since there exist *concave conjugate function* of a function  $g : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$  which is the function  $g_* : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$  defined as  $g_*(y) := \inf_{x \in \mathbb{R}^n} \{y^\top x - g(x)\}$ , where  $y \in \mathbb{R}^n$ . From now, if we say conjugate function it means that convex conjugate function.



**Figure 1.2:** Convex conjugate function.

closed, and convex function  $g : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$  is the function  $g^{per} : \mathbb{R}^n \times \mathbb{R}_+ \rightarrow \bar{\mathbb{R}}$  defined by

$$g^{per}(x, t) = \begin{cases} tg\left(\frac{x}{t}\right), & t > 0, \\ \delta_{dom(g^*)}^*(x), & t = 0. \end{cases}$$

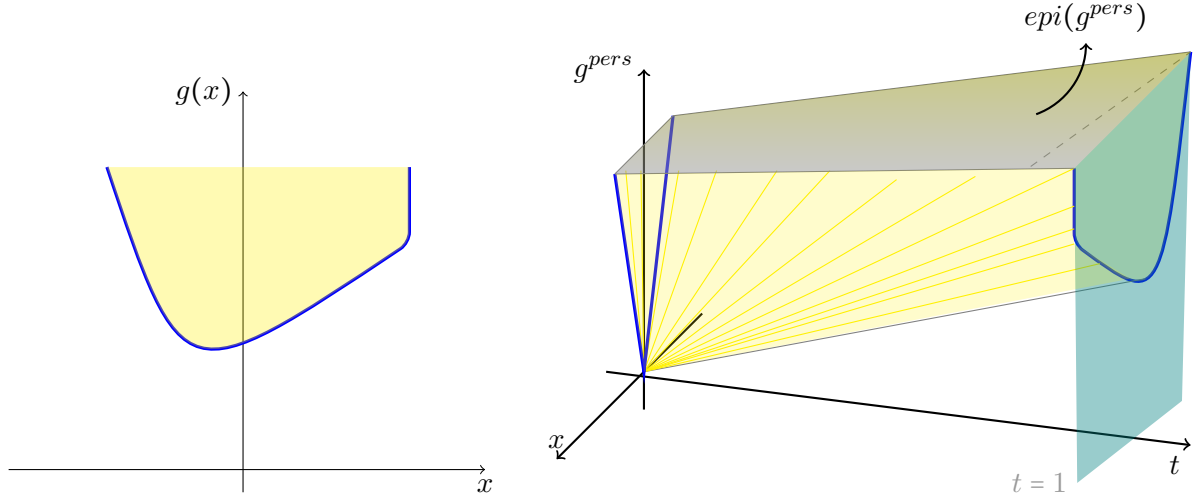
□

The convex perspective of a proper, closed, and convex function is also proper, closed, and convex (more precisely, jointly in  $(x, t)$ ); see [99, page 35] for convexity and properness, and [99, page 67 and Theorem 13.3] for closedness. Figure 1.3 displays a proper, closed, and convex function along with its convex perspective function.

**Remark 1.1.2** ([27]). A proper, closed, and convex function  $g$  conforms to the following relation with its convex conjugate and perspective functions

$$g^{per}(x, t) = \sup_y \{y^\top x - tg^*(y) \mid y \in dom(g^*)\}.$$

In the convex analysis literature,  $g^{per}(x, 0)$  is called the *asymptotic function* or *recession function* of  $g$ . Moreover,  $g^{per}(x, 0) = \liminf_{\substack{x' \rightarrow x \\ t' \downarrow 0}} t'g\left(\frac{x'}{t'}\right)$  [7, 69]. □



**Figure 1.3:** A proper, closed, and convex function and its convex perspective function.

**Proposition 1.1.1.** *If  $g$  is a proper, closed, and convex function, then*

$$\sup_{t>0, x \in \mathbb{R}^n} g^{per}(x, t) = \sup_{t \geq 0, x \in \mathbb{R}^n} g^{per}(x, t),$$

and

$$\inf_{t>0, x \in \mathbb{R}^n} g^{per}(x, t) = \inf_{t \geq 0, x \in \mathbb{R}^n} g^{per}(x, t).$$

*Proof.* Let  $x^0 \in \mathbb{R}^n$ . We have

$$\begin{aligned} g^{per}(x^0, t_0 = 0) &= \liminf_{(x^i, t_i) \rightarrow (x^0, 0)} g^{per}(x^i, t_i > 0) \\ &\leq \sup_{(x^i, t_i) \rightarrow (x^0, 0)} g^{per}(x^i, t_i > 0) \\ &\leq \sup_{t>0, x \in \mathbb{R}^n} g^{per}(x, t). \end{aligned}$$

So,  $\sup_{t>0, x \in \mathbb{R}^n} g^{per}(x, t) = \sup_{t \geq 0, x \in \mathbb{R}^n} g^{per}(x, t)$ .

As  $\inf_{t>0, x} g^{per}(x, t) \geq \inf_{t \geq 0, x} g^{per}(x, t)$ , let  $\ell \in \{g^{per}(x, t) | t \geq 0, x \in \mathbb{R}^n\}$ . We want to show  $\ell \geq \inf_{t>0, x} g^{per}(x, t)$ .

1. If  $\ell = g^{per}(x^0, t_0)$  for some  $x^0 \in \mathbb{R}^n$  and  $t_0 > 0$ , then  $\ell \geq \inf_{t>0, x} g^{per}(x, t)$ .

2. If  $\ell = g^{per}(x^0, 0)$  for some  $x^0 \in \mathbb{R}^n$ , then

$$\begin{aligned} \ell = g^{per}(x^0, 0) &= \liminf_{(x^i, t_i) \rightarrow (x^0, 0)} g^{per}(x^i, t_i > 0) \\ &\geq \inf_{(x^i, t_i) \rightarrow (x^0, 0)} g^{per}(x^i, t_i) \\ &\geq \inf_{t > 0, x \in \mathbb{R}^n} g^{per}(x, t). \end{aligned}$$

The proof is complete. □

As a consequence of the above proposition, we have

$$\sup_{t > 0, x} -g^{per}(x, t) = \sup_{t \geq 0, x} -g^{per}(x, t).$$

The definitions are extended to partial conjugate and perspective. The partial conjugate of a function  $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \bar{\mathbb{R}}$  with respect to its second argument (likewise for first argument) is the function  $g^{*2} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \bar{\mathbb{R}}$  defined as  $g^{*2}(x, w) = \sup_{y \in \mathbb{R}^{n_y}} \{w^\top y - g(x, y)\}$ , and its domain is denoted by  $dom(g^{*2})(x, \cdot)$ . If  $h : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \bar{\mathbb{R}}$  is a proper, closed, and concave function in its second argument, then its concave partial perspective  $h^{per} : \mathbb{R}^{n_x} \times \mathbb{R}_+ \times \mathbb{R}^{n_u} \rightarrow \bar{\mathbb{R}}$  is defined as

$$h^{per}(x, t, u) = \begin{cases} th\left(x, \frac{u}{t}\right), & t > 0, \\ -\delta_{dom((-h)^{2*}(x, \cdot))}^*(u), & t = 0. \end{cases}$$

Henceforth, for ease of notation, we use  $0h(x, u/0)$  instead of  $h^{per}(x, 0, u)$ .

**Theorem 1** (Properties of conjugate functions, [99]). *The conjugate of a function  $g$  is closed and convex. Moreover, if  $g$  is closed and convex, then  $g^{**} = g$ . Finally, a convex function  $g$  is proper if and only if  $g^*$  is proper.*

In Table 1.1, we present some transformation rules that are useful for the computation of conjugate functions.

Furthermore, support functions of several important sets are listed in Table 1.2.

**Table 1.1:** Conjugate functions of some popular forms.

Number	Function and assumptions	Conjugate function	Reference
1	$f(x) + a$	$f^*(y) - a$	[39, page 95]
2	$f(x + a)$	$f^*(y) - a^\top y$	[99, page 107]
3	$f(x) + a^\top x$	$f^*(y - a)$	[99, page 107]
4	$af(x), a > 0$	$af^*\left(\frac{y}{a}\right)$	[99, page 140]
5	$f(ax), a > 0$	$f^*\left(\frac{y}{a}\right)$	[99, page 107]
6	$\sum_{i=1}^m f_i(x)$	$\min_{y^i} \left\{ \sum_{i=1}^m f_i^*(y^i) \mid \sum_{i=1}^m y^i = y \right\}$	[99, page 145]
7	$\sum_{i=1}^m f_i(x^i)$	$\sum_{i=1}^m f_i^*(y^i)$	[39, page 95]
8	$f(Ax + b)$	$f^*(A^{-\top}y) - b^\top A^{-\top}y$	[99, page 107]
a	A nonsingular		
b	otherwise	$\inf \{f^*(z) - b^\top z \mid A^\top z = y\}$	[101, Section B.4]

*Note:* The functions  $f(x)$  and  $f_i(x)$  in this table are convex. In row 6, it is assumed that  $\cap_{i=1}^m \text{ri}(\text{dom}(f_i)) \neq \emptyset$ ; if  $f_i(x)$  is linear for some  $i$ , then the corresponding  $\text{ri}(\text{dom}(f_i))$  can be replaced by  $\text{dom}(f_i)$  in this condition. This is called *the sum rule for conjugate functions*. In row 7, the partition of  $x$  is denoted as  $x^1 \dots x^m$ . Row 8 is called the *linear substitution rule*.

**Table 1.2:** Support functions of some popular sets.

Number	$S, S \neq \emptyset$	$\delta_S^*(y)$	Reference
1	$\{x \mid Ax = b\}$	$\min_z \{b^\top z \mid A^\top z = y\}$	[39, page 380]
2	$\{x \mid Ax \leq b\}$	$\min_z \{b^\top z \mid A^\top z = y, z \geq 0\}$	[39, page 380]
3	$\{x \mid \ x\ _p \leq \rho\}$	$\rho \ y\ _q, \frac{1}{p} + \frac{1}{q} = 1$	[12, page 272]
4	$S = \bigcap_{i=1}^m S_i$	$\min_{y^i} \left\{ \sum_{i=1}^m \delta_{S_i}^*(y^i) \mid \sum_{i=1}^m y^i = y \right\}$	[99, page 146]
7	$S = S_1 \times \dots \times S_m$	$\sum_{i=1}^m \delta_{S_i}^*(y^i), y^1 \dots y^m$ is a partition of $y$	[12, page 294]

## 1.2 Fenchel Duality

In mathematical programming, the duality concept provides a powerful tool for understanding and solving optimization problems. Identifying relationships between the primal and dual problems provides valuable insights and alternative formulations. Fenchel Duality is a specific form of duality based on the conjugate function. In this section, we provide a brief overview of Fenchel duality.

**Definition 1.2.1** (Fenchel's Dual of a Convex Programming, [101, 129]). Consider the following primal convex optimization problem:

$$\begin{aligned} \inf_y \quad & g_0(y) \\ \text{s.t.} \quad & g_i(y) \leq 0, \quad i \in [m], \end{aligned} \tag{P}$$

where the functions  $g_i : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$ ,  $i \in [m_0]$ , are proper, closed, and convex. The Fenchel dual of (P) is defined as

$$\begin{aligned} \sup_{\lambda, \{w^i\}_{i=0}^m} \quad & -g_0^*(w^0) - \sum_{i=1}^m (g_i^*)^{per}(w^i, \lambda_i) \\ \text{s.t.} \quad & \lambda \geq \mathbf{0}, \quad \sum_{i=0}^m w^i = \mathbf{0}, \quad w^0 \in \text{dom}(g_0^*), \\ & (w^i, \lambda_i) \in \text{dom}((g_i^*)^{per}), \quad i \in [m]. \end{aligned} \tag{D}$$

□

For the ease of notation in (D), we use  $\lambda_i g_i^* \left( \frac{w^i}{\lambda_i} \right)$  to denote  $(g_i^*)^{per}(w^i, \lambda_i)$ , even for  $\lambda_i = 0$ . So, we write (D) as follows:

$$\begin{aligned} \sup_{\lambda, \{w^i\}_{i=0}^m} \quad & -g_0^*(w^0) - \sum_{i=1}^m \lambda_i g_i^* \left( \frac{w^i}{\lambda_i} \right) \\ \text{s.t.} \quad & \lambda \geq \mathbf{0}, \quad \sum_{i=0}^m w^i = \mathbf{0}, \quad w^0 \in \text{dom}(g_0^*), \\ & \frac{w^i}{\lambda_i} \in \text{dom}(g_i^*), \quad i \in [m], \end{aligned}$$

where  $\frac{w^i}{\lambda_i} \in \text{dom}(g_i^*)$  for  $\lambda_i = 0$  means  $\delta_{\text{dom}(g_i)}^*(w^i) < \infty$ .

**Remark 1.2.1.** In problem (D), the constraints corresponding to the domain are essential and in many cases, they also lead to convex constraints for (D). Moreover, since (D) is a maximization problem, these constraints hold explicitly. These constraints, in many cases, enable us to eliminate the variables  $w^i$ . For brevity, in [101, 129] the dual problem has been written as follows:

$$\begin{aligned} \sup_{\lambda, \{w^i\}_{i=0}^m} \quad & -g_0^*(w^0) - \sum_{i=1}^m \lambda_i g_i^* \left( \frac{w^i}{\lambda_i} \right) \\ \text{s.t.} \quad & \lambda \geq \mathbf{0}, \quad \sum_{i=0}^m w^i = \mathbf{0}. \end{aligned}$$



□

**Definition 1.2.2** (Slater regularity). Problem (P) is Slater regular if there exists some  $y^s \in \cap_{i \in [m_0]} \text{ri}(\text{dom}(g_i))$ , such that  $g_i(y^s) < 0$  for all  $i \in [m]$ . □

It is important to note that if (P) is Slater regular, then the optimal values of (P) and (D) are equal [129].

The following example presents the Fenchel dual of a general linear programming problem.

**Example 1.2.1** (Linear Programming). We consider the standard linear programming problem

$$\begin{aligned} \inf_x \quad & c^\top x \\ \text{s.t.} \quad & Ax = b, \\ & x \geq 0. \end{aligned} \tag{LP}$$

The objective function is  $f(x) := c^\top x$ . Let  $\delta_{S_1}(x)$  and  $\delta_{S_2}(x)$  denote as the indicator functions of the sets  $S_1 := \{x \mid Ax = b\}$ , and  $S_2 := \{x \mid x \geq 0\}$ , respectively. In this case, we consider the function  $h(x)$  defined by

$$h(x) := f(x) + \delta_{S_1}(x) + \delta_{S_2}(x),$$

and get the unconstrained problem

$$\inf_x h(x) \tag{LP_1}$$

The Fenchel's dual is given by

$$\begin{aligned} \sup_{w^0} \quad & -h^*(w^0) \\ \text{s.t.} \quad & w^0 = 0, \\ & w^0 \in \text{dom}(h^*). \end{aligned}$$

Now according to the sum rule for conjugate functions (see Table 1.1), we get

$$\begin{aligned} \sup_{y^0, y^1, y^2} \quad & -f^*(y^0) - \delta_{S_1}^*(y^1) - \delta_{S_2}^*(y^2) \\ \text{s.t.} \quad & y^0 + y^1 + y^2 = 0. \end{aligned}$$

Note that

$$\begin{aligned}f^*(y) &= 0, \quad \text{dom}(f^*) = \{c\}, \\ \delta_{S_1}^*(y) &= \min_z \{b^\top z \mid A^\top z = y\}, \\ \delta_{S_2}^*(y) &= 0, \quad \text{dom}(\delta_{S_2}^*) = \{y \mid y \leq 0\}.\end{aligned}$$

Substitution of the expressions for  $f^*$  and  $\delta_{S_i}^*$  ( $i = 1, 2$ ) yields the following formulation of the dual problem:

$$\begin{aligned}\sup_{z, y^0, y^1, y^2} \quad & -b^\top z \\ \text{s.t.} \quad & y^0 + y^1 + y^2 = 0, \\ & y^0 = c, \\ & A^\top z = y^1, \\ & y^2 \leq 0.\end{aligned}$$

We can eliminate the vectors  $y^i$ , which gives

$$\begin{aligned}\sup_z \quad & -b^\top z \\ \text{s.t.} \quad & -A^\top z \leq c.\end{aligned}$$

Changing the sign of  $z$  leads to the following well-known duality problem for linear optimization

$$\begin{aligned}\sup_z \quad & b^\top z \\ \text{s.t.} \quad & A^\top z \leq c.\end{aligned}$$

□

### 1.3 Robust Optimization

The concept of robust optimization was first introduced by Soyster in 1973, and later developed by El Ghaoui, Ben-Tal, Nemirovski, Bertsimas, and den Hertog [111, 13, 22]. Mathematically, robust optimization involves defining an uncertainty set that encompasses

all potential parameter values that a decision-maker wants to protect against. The goal is to generate solutions that are feasible for all of these values or, in other words, for the worst-case scenario within the uncertainty set.

Static robust optimization assumes that decisions must be made before the value of uncertain parameters is known. Its objective is to find solutions that are robust to changes in the value of these uncertain parameters. In contrast, Adjustable Robust Optimization (ARO) relaxes this assumption by allowing some decisions to be made before the values of uncertain parameters are known, referred to as “here-and-now” decisions, while other decisions, called “wait-and-see” decisions, can be made after (part of) the uncertain parameter values are revealed [14]. Before delving into the mathematical definitions of these two optimization issues, it is important to briefly mention some sources of uncertainty.

There are various factors that contribute to uncertainty in parameters, with some of them being difficult to determine. For instance, measurement errors arise from inaccuracies in physical devices, such as thermometers or incomplete inventory data. On the other hand, forecast errors result from imprecise knowledge of future demand for products, discount rates, or resource prices. Implementation errors occur when there are inaccuracies in the application of devices, such as electrical power or metal component lengths. The reliability of past data, also known as veracity in the context of Big Data, is affected by various factors such as wrongly entered data, missing observations, and other forms of data unreliability. Even slight deviations from assumed parameter values can have significant effects, leading to infeasible constraints or considerable losses in the objective function value. Therefore, it is essential to employ optimization methods that can effectively account for and accommodate uncertainty, as highlighted in [13].

In the rest of this chapter, we recall some definitions for robust optimization. The general form of an uncertain nonlinear optimization problem is as follows

$$\inf_{x \in \mathcal{X}} \inf_{y \in \mathcal{Y}(x,u)} f_0(x, y, u) \tag{1.1}$$

where  $x \in \mathbb{R}^{n_x}$  is a vector containing *non-adjustable* (also known as *here-and-now* or *static*) decisions,  $y \in \mathcal{Y}(x, u) \subseteq \mathbb{R}^{n_y}$  is a vector containing *adjustable* (also known as *wait-and-*

see) decisions,  $\mathcal{Y}(x, u) = \{y \in \mathbb{R}^{n_y} : f_j(x, y, u) \leq 0, j \in [m]\}$ ,  $u \in \mathcal{U} \subseteq \mathbb{R}^{n_u}$  is an uncertain parameter,  $\mathcal{U}$  is the uncertainty set, and  $\mathcal{X} \subseteq \mathbb{R}^{n_x}$  is a set with additional constraints on  $x$ .

The static and adjustable robust counterparts corresponding to uncertain problem (1.1) can be defined as follows.

**Definition 1.3.1** (Static Robust Optimization, [87]). The static Robust Counterpart (RC) for uncertain problem (1.1) is defined by

$$\begin{aligned} \inf_{x \in \mathcal{X}, y} \quad & \sup_{u \in \mathcal{U}} f_0(x, y, u) \\ \text{s.t.} \quad & \sup_{u \in \mathcal{U}} f_j(x, y, u) \leq 0, \quad j \in [m]. \end{aligned} \tag{RC}$$

□

Notably, mandating the constraint to hold in the worst-case scenario is tantamount to enforcing a constraint that necessitates the proposition to hold for all feasible values of  $u$  within the uncertainty set  $\mathcal{U}$ , i.e.,

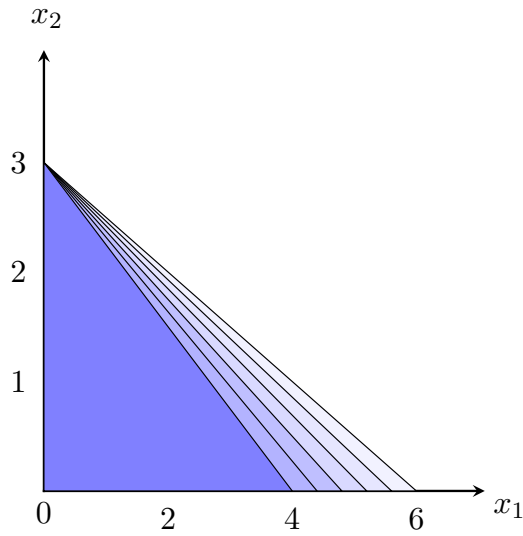
$$\sup_{u \in \mathcal{U}} f_j(x, y, u) \leq 0 \iff f_j(x, y, u) \leq 0, \quad \forall u \in \mathcal{U}.$$

The first form of the constraint implies that it can be reformulated if  $f_j$  exhibits concavity in its third argument, as optimizing a concave function is considered to be an easy task. When  $f_j$  is concave in  $u$ , computationally tractable reformulations of the robust counterpart can be found [12].

**Example 1.3.1.** *We illustrate how parameter uncertainty affects the following problem*

$$\begin{aligned} \min_{x_1, x_2} \quad & -2x_1 - x_2 \\ \text{s.t.} \quad & (2 + u)x_1 + 4x_2 \leq 12, \\ & x_1, x_2 \geq 0, \end{aligned}$$

with  $u$  as an uncertain parameter. If  $u = 0$ , as this is a simple instance, we can easily obtain the optimal solution, which is  $x_1 = 6$ ,  $x_2 = 0$ , resulting in an objective value of  $-12$ . We consider  $u = 0$  as nominal value. The corresponding optimal value (solution) is called the



**Figure 1.4:** The robust feasible region.

nominal optimal value (solution). It is evident that any value of  $u$  greater than 0 renders the nominal solution infeasible. In order to acquire a robust solution, we define an uncertainty set for the parameter  $u$ , and for now, we will adhere to set  $\mathcal{U} = \{u \mid 0 \leq u \leq 1\}$ .

The feasible regions for different values of  $u$  within  $\mathcal{U}$  exhibit overlapping sections. The darkest area in Figure 1.4 represents the robust feasible region, encompassing the solutions that are feasible for all  $u \in \mathcal{U}$ .

In order to obtain a solution that is robust, meaning feasible regardless of the variation in  $u$  within the uncertainty set, we must solve the following model:

$$\begin{aligned}
 \min_{x_1, x_2} \quad & -2x_1 - x_2 \\
 \text{s.t.} \quad & (2 + u)x_1 + 4x_2 \leq 12, \quad \forall u \in \mathcal{U} \\
 & x_1, x_2 \geq 0,
 \end{aligned} \tag{1.2}$$

The model has an infinite number of constraints. Fortunately, we can reformulate this

constraint as below to overcome this issue

$$\begin{aligned}
& (2 + u)x_1 + 4x_2 \leq 12, \quad \forall u \in \mathcal{U} \\
& \iff 2x_1 + \max_{u \in \mathcal{U}}\{ux_1\} + 4x_2 \leq 12, \\
& \iff 2x_1 + \min_z\{z \mid z \geq x_1, z \geq 0\} + 4x_2 \leq 12, \\
& \iff \begin{cases} 2x_1 + z + 4x_2 \leq 12, \\ z \geq x_1, \\ z \geq 0. \end{cases}
\end{aligned}$$

The equivalence in the second line arises from a worst-case perspective. In the third line, we employ strong duality in linear programming. Finally, in the last line, we utilize the fact that the constraint is satisfied for the minimizer  $z \geq 0$  if and only if there exists an  $x \geq 0$  that satisfies both  $2x_1 + z + 4x_2 \leq 12$  and  $z \geq x_1$ . It should be noted that in the final statement,  $z$  is not necessarily required to be the minimizer. So, by employing duality, the whole optimization problem can be reformulated into an equivalent system of deterministic constraints

$$\begin{aligned}
& \min_{x_1, x_2, z} \quad -2x_1 - x_2 \\
& \text{s.t.} \quad 2x_1 + z + 4x_2 \leq 12, \\
& \quad \quad x_1 - z \leq 0, \\
& \quad \quad x_1, x_2, z \geq 0.
\end{aligned}$$

Therefore, the optimal robust solution is  $x_1 = 4$ ,  $x_2 = 0$ , and  $z = 4$  with optimal objective value is  $-8$ . Note that, we know (1.2) is equivalent with

$$\begin{aligned}
& \min_{x_1, x_2} \quad -2x_1 - x_2 \\
& \text{s.t.} \quad 2x_1 + 4x_2 + \max_{u \in \mathcal{U}}\{ux_1\} \leq 12, \\
& \quad \quad x_1, x_2 \geq 0.
\end{aligned}$$

Another trick to reach a deterministic version is the fact that the worst-case problem is a linear programming problem, and we know that its optimal value lies among the vertices of

the uncertainty region. By using vertex enumeration we get

$$\begin{aligned} \min_{x_1, x_2} \quad & -2x_1 - x_2 \\ \text{s.t.} \quad & 2x_1 + 4x_2 \leq 12, \\ & 2x_1 + 4x_2 + x_1 \leq 12, \\ & x_1, x_2 \geq 0. \end{aligned}$$

The optimal robust solution of the last problem is  $x_1 = 4$  and  $x_2 = 0$  with the same optimal value as before.  $\square$

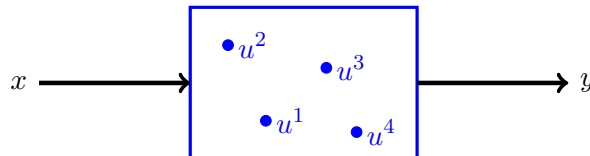
**Assumption 1.** In (1.1), we assume that  $f_j : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_u} \rightarrow \bar{\mathbb{R}}$ ,  $j \in [m_0]$  are convex in  $x$ , proper, closed, and convex in  $y$ , and concave in  $u$ .  $\square$

**Definition 1.3.2** (Adjustable Robust Optimization, [87]). The Adjustable Robust Counterpart (ARC) for uncertain problem (1.1) is defined by

$$\begin{aligned} \inf_{x \in \mathcal{X}} \sup_{u \in \mathcal{U}} \inf_y f_0(x, y, u) \\ \text{s.t. } f_j(x, y, u) \leq 0, \quad j \in [m], \end{aligned} \tag{ARC}$$

where  $x \in \mathbb{R}^{n_x}$  is a vector containing static (or here-and-now) decisions, and  $y \in \mathbb{R}^{n_y}$  is a vector containing adjustable (or wait-and-see) decisions.  $\square$

Figure 1.5 provides a visual representation of (ARC). Here-and-now decision  $x$  should be chosen without knowledge of the uncertainty and here-and-now decision  $y$  can be adjusted according to realizing scenarios  $\{u^1, u^2, u^3, u^4\}$ .



**Figure 1.5:** Adjustable robust optimization. The blue square is the uncertainty set  $\mathcal{U}$ .

**Remark 1.3.1.** The following representation, expressed as

$$\begin{aligned} \inf_{x \in \mathcal{X}, y(u)} \quad & \sup_{u \in \mathcal{U}} \quad f_0(x, y(u), u) \\ \text{s.t.} \quad & \forall u \in \mathcal{U}: f_j(x, y(u), u) \leq 0, \quad j \in [m] \end{aligned} \tag{1.3}$$

serves as an equivalent formulation of (ARC). While both the formulations (ARC) and (1.3) are commonly used in various papers without proof of their equivalence, [113] provides a formal proof establishing their equivalence. The formulation (1.3) is more intuitive when solving the models via decision rule policy.  $\square$

**Definition 1.3.3** (Fixed Recourse Problem, [87]). Problem (1.1) has fixed-recourse when

$$f_j(x, y, u) = \hat{f}_j(x, u) + \hat{g}_j(x, y), \quad j \in [m_0].$$

$\square$

In this thesis, we consider a special case of the above notion defined as follows.

**Definition 1.3.4** (Separable Fixed Recourse Problem). We say (1.1) has separable fixed-recourse when

$$f_j(x, y, u) = \hat{f}_j(x, u) + g_j(y), \quad j \in [m_0].$$

$\square$

## 1.4 Adjustable Robust Optimization Approximations

There is a wide range of methods available in the literature to tackle ARO problems. In this context, we will emphasize some of the most notable approaches.

- **Finite Scenario Approach:** This approach involves consideration of a limited subset, drawn from the uncertainty set, comprising a finite number of scenarios. Instead of making decision rules, one can introduce a new single optimization variable to each scenario. It should be noted that this approach does not serve as a complete solution method, as it only ensures feasibility for a portion of scenarios within the uncertainty set. The decisions derived from this approach may be infeasible in practice, and the



objective value merely acts as a lower bound for the optimal objective value of the adjustable robust optimization model [20]. Nevertheless, it remains a valuable approach for evaluating the effectiveness of other solution methods. With careful selection, a small set of scenarios within the subset can provide robust lower bounds. One possible strategy for scenario selection is to consider the set of scenarios that activate the affine adjustable robust model, as introduced in [66].

- **Affine Decision Rules:** For linear adjustable robust optimization problems this is an approximate solution technique. The wait-and-see decision can be uniquely adapted to individual scenarios via the use of affine decision rules for the variable  $y$ , which is a function of the scenario  $u \in \mathcal{U}$ . The approximation for the wait-and-see decisions can be expressed as follows:

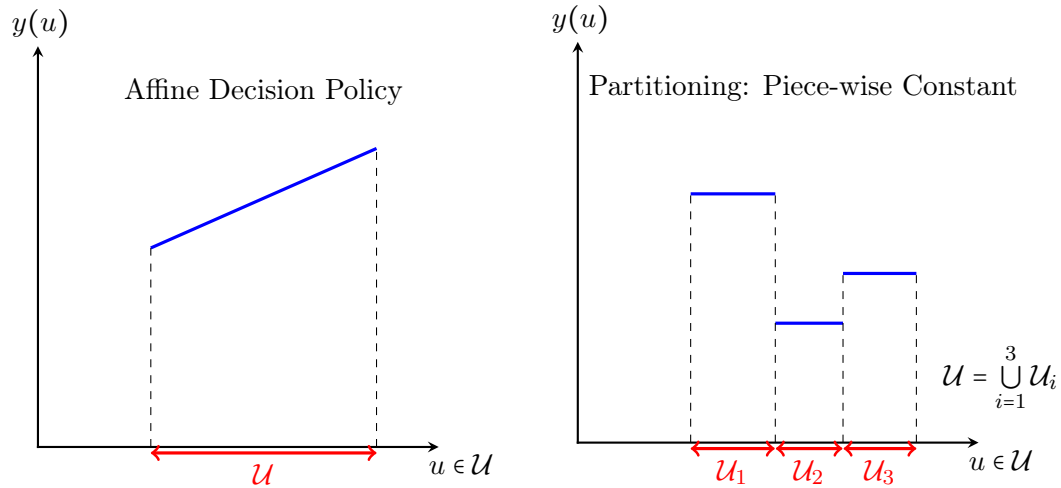
$$y(u) := z + Zu,$$

where  $z \in \mathbb{R}^{n_y}$  and  $Z \in \mathbb{R}^{n_y \times n_u}$  are the auxiliary variables and jointly optimized alongside the here-and-now decision variable  $x$ . While an explicit theorem establishing the optimality gap between solving (ARC) problem with this affine decision rule is yet to be found, empirical evidence indicates its excellent performance, particularly in inventory problems [13, 14, 28].

- **Piece-wise Design Policies:** Another strategy that can be employed to approach the upper bound for the optimal value of the (ARC) problem is the utilization of piece-wise decision policies. These methods involve partitioning the uncertainty set into smaller subsets and applying decision policies to adjustable (wait-and-see) variables. In piece-wise constant policies, a distinct copy of the adjustable variable is assigned to each partition. This approach is commonly known as the finite adaptability or K-adaptability approach [67, 97]. In piece-wise linear policies, a piece-wise affine decision rule is applied to the adjustable variables associated with each partition. As a result of using this decision strategy, an upper bound approximation to the (ARC) problem is obtained [23]. These piece-wise design policies allow the decision-making process to be tailored to the different subsets of the uncertainty sets.

The decision rule showcased in Figure 1.6 illustrates how a decision rule can be

constructed using piece-wise constant and affine policies. In piece-wise constant policies, the wait-and-see decisions do not follow a continuous pattern but remain constant for each subset of uncertainty.



**Figure 1.6:** Affine and piece-wise constant decision policies.

- Partitioning Uncertainty Set:** Closely tied to piece-wise design policy, this scheme involves partitioning the uncertainty set into  $K$  distinct subsets and assigning decision rules to each partition. This concept was first introduced in [11, 114], where a predetermined partitioning approach was initially emphasized. Subsequently, an algorithmic methodology for iterative partition refinement was introduced in [23, 97]. These papers provide compelling examples that highlight the strong performance of this method. However, it is important to note that not every refinement guarantees improvement, and convergence may not be achieved for all problem instances. Furthermore, as the partition is further refined, the size of the model increases accordingly.
- Fourier-Motzkin Elimination:** This approach provides a solution for addressing linear ARO problems with fixed recourse by employing Fourier-Motzkin elimination (FME) to eliminate adjustable variables. It is important to note that FME can introduce a significant number of additional constraints, potentially growing exponentially. However, in the context of ARO in the linear case with a small number of adjustable variables, FME has demonstrated its effectiveness in determining the optimal solution

by eliminating all adjustable variables [128]. When dealing with larger models, a possible strategy involves eliminating a specific subset of adjustable variables and applying decision rules to the remaining variables.

To conclude the discussion on solution approaches, we will illustrate the concepts we have discussed using the following example.

**Example 1.4.1.** *Let us consider the following linear ARO problem*

$$\begin{aligned}
 \min_{x, y(\cdot)} \quad & 3x \\
 \text{s.t.} \quad & x - y(u) \leq -u, \quad \forall u \in [0, 1] \\
 & -x + y(u) \leq u + 2, \quad \forall u \in [0, 1] \\
 & y(u) \geq 1. \quad \forall u \in [0, 1]
 \end{aligned}$$

▷ Finite scenario approach: *We restrict ourselves to only finite scenarios exclusively, as below*

$$\mathcal{S} = \left\{ \frac{1}{2}, \frac{2}{5} \right\} \subseteq [0, 1].$$

*Now, we can assign two individual variables  $y_1$  and  $y_2$  to each considered scenarios. The optimal objective value of the deterministic optimization model below represents a lower bound because we only guarantee feasibility for a strict subset of the uncertainty region.*

$$\begin{aligned}
 \min_{x, y_1, y_2} \quad & 3x \\
 \text{s.t.} \quad & x - y_1 \leq -\frac{1}{2}, \quad -x + y_1 \leq \frac{1}{2} + 2, \quad y_1 \geq 1, \\
 & x - y_2 \leq -\frac{2}{5}, \quad -x + y_2 \leq \frac{2}{5} + 2, \quad y_2 \geq 1.
 \end{aligned}$$

*The optimal value is  $z_1^* = -\frac{21}{5}$ .*

▷ Affine decision rule: *This approximate approach provides an upper bound on the optimal value of the problem. By considering affine policy  $y(u) := wu + z$ , we get the following static*

*robust optimization model*

$$\begin{aligned}
& \min_{x,w,z} && 3x \\
& \text{s.t.} && x - (wu + z) \leq -u, \quad \forall u \in [0, 1] \\
& && -x + (wu + z) \leq u + 1, \quad \forall u \in [0, 1] \\
& && (wu + z) \geq 1. \quad \forall u \in [0, 1]
\end{aligned}$$

We emphasize that  $w$  and  $z$  are the static (or here-and-now) decision variables. So, we can use static RO techniques to solve it. By using vertex enumeration, the optimal solutions are  $w^* = 1$ ,  $z^* = 1$ , and  $x^* = -1$  with the optimal value  $z_2^* = -3$ . Overall, we get the upper bound and a decision rule:

$$y(u) = u + 1.$$

So, the moment we have the value of the uncertain parameter  $u$ , we have a value for the adjustable variable  $y$ .

▷ Finite adaptability approach: By partitioning the uncertainty set into smaller subsets  $[0, \frac{1}{2}]$  and  $[\frac{1}{2}, 1]$ . Instead of making decision policies, we apply constant decision rules to adjustable (wait-and-see) variables on each partition, i.e., we can attach two individual optimization variables  $y_1$  and  $y_2$  to each of the partitions. Therefor, we get the following static RO problem

$$\begin{aligned}
& \min_{x,y_1,y_2} && 3x \\
& \text{s.t.} && x - y_1 \leq -u, \quad \forall u \in [0, \frac{1}{2}] \\
& && -x + y_1 \leq u + 2, \quad \forall u \in [0, \frac{1}{2}] \\
& && x - y_2 \leq -u, \quad \forall u \in [\frac{1}{2}, 1] \\
& && -x + y_2 \leq u + 2, \quad \forall u \in [\frac{1}{2}, 1] \\
& && y_1, y_2 \geq 1.
\end{aligned}$$

By using the vertex enumeration technique, the optimal value of the last static RO problem is  $z_3^* = -3$ . Hitherto, the optimal value of the original problem is in  $[-\frac{21}{5}, -3]$ .

▷ Fourier-Motzkin elimination: Note that for given  $u \in [0, 1]$ , in the main problem, we can eliminate adjustable variable  $y(u)$ , since

$$\begin{aligned} x - u &\leq y(u), \\ y(u) &\leq u + x + 2, \\ 1 &\leq y(u). \end{aligned}$$

So, we have

$$\begin{aligned} 1 &\leq u + x + 2, \\ \iff 0 &\leq x + u + 1. \end{aligned}$$

Note that other constraint in elimination process is  $u \geq -1$ , due to  $u \in [0, 1]$ , which is redundant. Finally, the next static RO problem is an equivalent reformulation of the main ARO problem

$$\begin{aligned} \min_x \quad & 3x \\ \text{s.t.} \quad & 0 \leq x + u + 1, \quad \forall u \in [0, 1]. \end{aligned}$$

Obviously, the optimal value is  $z^* = -3$ . □

## 1.5 Uncertainty Sets

In this section, we delve into the classical choices for uncertainty sets in robust optimization. These uncertainty sets play a crucial role in handling uncertain parameters and ensuring robustness in the optimization process. By carefully defining and selecting the appropriate uncertainty sets, we can proficiently manage uncertainties and make sound decisions that are resilient to adverse conditions.

One of the commonly used uncertainty sets is the *box uncertainty set*. Within this set, uncertain parameters are presumed to reside within predetermined bounds, creating a

hyperrectangle or a box. It is mathematically described as follows

$$\mathcal{U}_{\text{box}} = \{u \in \mathbb{R}^{n_u} : L \leq u \leq U\},$$

where  $L, U \in \mathbb{R}^{n_u}$  are given parameters. The box uncertainty set provides simplicity and tractability, but it may lead to overly conservative solutions if the actual uncertainty distribution significantly diverges from the assumed bounds.

Another classical option is the *ellipsoidal uncertainty set*. In contrast to the rigid box set, the ellipsoidal uncertainty set provides greater flexibility in representing uncertainty, allowing for a wider range of potential parameter values. Formally, it is expressed as follows

$$\mathcal{U}_{\text{ellipsoid}} = \{u \in \mathbb{R}^{n_u} : \|Qu\|_2 \leq 1\},$$

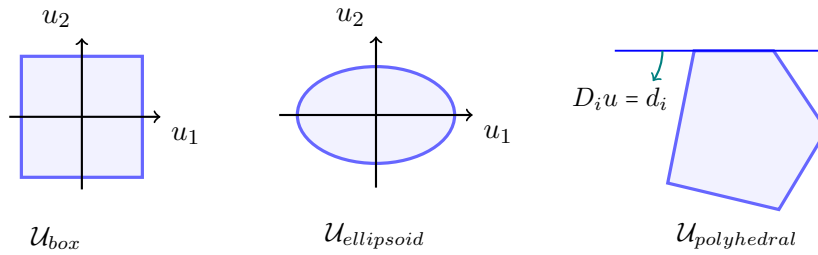
where  $Q \in \mathbb{R}^{n_u \times n_u}$  is given shape matrix. One notable advantage is its capacity to manage conservatism. However, a limitation arises when incorporating nonlinear functions into the model, which can pose challenges.

*Polyhedral uncertainty set* is another widely favored option in robust optimization. The uncertain parameters are restricted to a polyhedron characterized by a collection of linear inequalities. It is expressed by given parameters  $D \in \mathbb{R}^{r \times n_u}$  and  $d \in \mathbb{R}^r$ , as follows

$$\mathcal{U}_{\text{polyhedral}} = \{u \in \mathbb{R}^{n_u} : Du \leq d\}.$$

The polyhedral uncertainty set has the ability to capture intricate uncertainties and offers a versatile framework for modeling uncertainty, rendering it suitable for diverse applications. Nevertheless, solving optimization problems involving polyhedral uncertainty sets can present computational difficulties, particularly in the case of high-dimensional problems.

In Figure 1.7, classical uncertainty sets are depicted in the two-dimensional space  $\mathbb{R}^2$ .



**Figure 1.7:** Classical uncertainty sets.

## 1.6 Convex Quadratic Problems

Convex quadratic problems and their corresponding duals are important issues in optimization theory. These types of problems involve minimizing a convex quadratic function subject to linear constraints, and their dual problems involve maximizing a concave quadratic function subject to linear constraints. In Table 1.3, we can see a list of different convex quadratic problems and their corresponding Dorn dual problems. Proof of Dorn duality theorems for quadratic programs has been shown using the duality theorem for linear programs [55].

**Table 1.3:** Convex quadratic problems and their corresponding Dorn duals.

Primal Problem	Type	Dual Problem
	I	
$\min_x \frac{1}{2}x^\top Cx + p^\top x$ $\text{s.t. } Ax \geq b,$ $x \geq 0.$		$\max_{u,v} -\frac{1}{2}u^\top Cu + b^\top v$ $\text{s.t. } A^\top v - Cu \leq p,$ $v \geq 0.$
	II	
$\min_x \frac{1}{2}x^\top Cx + p^\top x$ $\text{s.t. } Ax \geq b.$		$\max_{u,v} -\frac{1}{2}u^\top Cu + b^\top v$ $\text{s.t. } A^\top v - Cu = p,$ $v \geq 0.$
	III	
$\min_x \frac{1}{2}x^\top Cx + p^\top x$ $\text{s.t. } Ax = b,$ $x \geq 0.$		$\max_{u,v} -\frac{1}{2}u^\top Cu + b^\top v$ $\text{s.t. } A^\top v - Cu \leq p.$
	IV	
$\min_x \frac{1}{2}x^\top Cx + p^\top x$ $\text{s.t. } Ax = b.$		$\max_{u,v} -\frac{1}{2}u^\top Cu + b^\top v$ $\text{s.t. } A^\top v - Cu = p.$

*Note:* Notice that at the optimal solution, in all the types listed above,  $u^* = x^*$ .



# CHAPTER 2

## Linear Adjustable Robust Optimization

George Dantzig:

*“... it is interesting to note that the original problem that started my research is still outstanding - namely the problem of planning or scheduling dynamically over time, particularly planning dynamically under uncertainty. If such a problem could be successfully solved it could eventually through better planning contribute to the well-being and stability of the world.”*

According to static Robust Optimization (RO), all decisions must be made before uncertainty is realized. However, this assumption does not hold true in many real-world problems. For example, a location-transportation problem can be complicated by the fact that while the locations of the production/storage facilities need to be decided as quickly as possible, it is possible to delay deciding how many goods to deliver to each customer until their respective uncertain demands are revealed. The Adjustable Robust Optimization (ARO) problem was introduced to address uncertainty in such problems [14]. The purpose of this chapter is to provide a short review of the literature relating to linear AROs.

### 2.1 Introduction

Consider the following family of uncertain Linear Programming problems:

$$\left\{ \min_z \{c^\top z : Az \leq b\} \right\}_{\zeta \in [A,b,c] \in \mathcal{Z}} \quad (2.1)$$

the parameters  $\zeta$ , which consist of  $A$ ,  $b$ , and  $c$ , vary within a specified uncertainty set  $\mathcal{Z} \subset \mathbb{R}^{m \times n} \times \mathbb{R}^m \times \mathbb{R}^n$ . Mathematically, the robust counterpart of uncertain problem (2.1) is

$$\min_z \left\{ \sup_{\zeta \in [A, b, c] \in \mathcal{Z}} c^\top z : Az \leq b, \forall \zeta \equiv [A, b, c] \in \mathcal{Z} \right\}. \quad (2.2)$$

In the static robust optimization approach, all variables represent decisions that need to be made prior to obtaining knowledge about the uncertain actual parameter realization. The decision-making process in the real world often consists of multiple stages, allowing for certain decisions to be made either after obtaining knowledge of the uncertain data or accurately predicting it. The next example clarifies this.

**Example 2.1.1.** *Consider a factory that produces  $P(t)$  units to satisfy demand  $d_t$  on day  $t = 1, 2$ . The actual value of  $d_t$  becomes known only at the end of day  $t$ . The decision on how much to produce on day  $t$  must be made at the beginning of that day. When choosing  $P(1)$ , we have no information about the actual demand, so  $P(1)$  represents a static decision. For  $P(2)$ , we already know the actual demand  $d_1$ , so we can say  $P(2)$  is a function of  $d_1$ . In other words, the adjustable variables depend on part of the uncertain data.  $\square$*

All variables that can be influenced by the realizations of the data are referred to as adjustable, while the remaining variables are categorized as static. This leads us to partition the vector  $z$  of variables in (2.1) as  $z = (x^\top, y^\top)^\top$ , where the sub-vector  $x$  represents the static variables and  $y$  represents the adjustable variables.

By making a clear distinction between the static and adjustable variables, we can express problem (2.1) in an equivalent form as follows:

$$\min_{(\tau, x), y} \left\{ \tau : c^\top \begin{pmatrix} x \\ y \end{pmatrix} \leq \tau, Ux + Vy \leq b \right\}_{[U, V, b, c] \in \mathcal{Z}} \quad (2.3)$$

and treat  $(\tau, x)$  as the static part of the solution. In the above reformulation (2.1), the objective function remains unaffected by both the uncertain data and the adjustable variables  $y$ . From this point onwards, without loss of generality, we can assume that an uncertain LP

problem can be rewritten as follows:

$$LP_{\mathcal{Z}} = \left\{ \min_{x,y} c^\top x : Ux + Vy \leq b \right\}_{\zeta=[U,V,b] \in \mathcal{Z}}. \quad (2.4)$$

**Definition 2.1.1** ([14]). The adjustable robust counterpart of the uncertain problem  $LP_{\mathcal{Z}}$  is defined as

$$\min_x \{c^\top x : \forall (\zeta = [U, V, b] \in \mathcal{Z}) \exists y : Ux + Vy \leq b\}. \quad (\text{ARC-LP})$$

In contrast, the robust counterpart of  $LP_{\mathcal{Z}}$  is:

$$\min_x \{c^\top x : \exists y \forall (\zeta = [U, V, b] \in \mathcal{Z}) : Ux + Vy \leq b\}. \quad (\text{RC-LP})$$

The greater flexibility of the (ARC-LP) compared to the (RC-LP) is evident, as former has a larger robust feasible set that allows for a better optimal value while still adhering to all potential constraint realizations. The distinction between (ARC-LP) and (RC-LP) can be remarkably substantial, as illustrated in the two subsequent examples.

**Example 2.1.2** (from [14]). *Let us consider an uncertain LP problem with a single equality constraint:*

$$\alpha x + \beta y = 1,$$

where the uncertain data  $(\alpha, \beta)$  can take values in the uncertainty set

$$\mathcal{Z} = \{(\alpha, \beta) \mid \alpha \in [\frac{1}{2}, 1], \beta \in [\frac{1}{2}, 1]\}.$$

Then the feasible set of (RC-LP) is  $\{x \mid \exists y \forall (\alpha, \beta) \in \mathcal{Z} : \alpha x + \beta y = 1\} = \emptyset$ . This happens because in particular for  $\alpha = 1$ , for each  $\beta \in [\frac{1}{2}, 1]$  the constraint  $x + \beta y = 1$  implies  $x = 1, y = 0$  as the unique solution. And then for each  $\alpha \in [\frac{1}{2}, 1]$ ,  $\alpha(1) + \beta(0) = 1$  does not hold. At the same time, the feasible set of the (ARC) is  $\{x \mid \forall (\alpha, \beta) \in \mathcal{Z} \exists y : \alpha x + \beta y = 1\} = \mathbb{R}$ , since for any fixed  $\bar{x}$ , the constraint can be satisfied by taking  $y = \frac{1-\alpha\bar{x}}{\beta}$ .  $\square$

**Example 2.1.3** (from [14]). Consider the following uncertain LP:

$$\min_{x,y} \{-x : (1 - 2\xi)x + y \geq 0, \xi x - y \geq 0, x \leq 1\}_{0 \leq \xi \leq 1}$$

Note that here the uncertainty  $\xi$  influences both the first and second constraints. It can be easily seen that the optimal value of the robust counterpart of this problem is

$$\min_x \{-x \mid \exists y \forall (\xi \in [0, 1]) : (1 - 2\xi)x + y \geq 0, \xi x - y \geq 0, x \leq 1\} = 0,$$

achieved at the unique solution  $x = 0$  and  $y = 0$ . The optimal value of (ARC) is

$$\min_x \{-x \mid \forall (\xi \in [0, 1]) \exists y : (1 - 2\xi)x + y \geq 0, \xi x - y \geq 0, x \leq 1\} = -1,$$

where for any  $\bar{x} \leq 1$  we can take  $y = \xi \bar{x}$  to obtain feasibility. □

## 2.2 Approximation of Linear ARO

In the rest of this short chapter will cover some of the methods and approaches that have been suggested in the literature to handle linear ARO problems. Several of these approaches have been reviewed in the previous chapter. Now, we will be reviewing a wide variety of non-decision policy approaches to solving linear ARO, which are presented here.

### 2.2.1 Dual Formulations

Along the lines of [21], consider the following ARO problem:

$$\begin{aligned} \min_{x \in \mathcal{X}} \quad & c^\top x \\ \text{s.t.} \quad & \forall \zeta \in \mathcal{U}, \exists y \geq 0 : Ax + By \geq R\zeta + r, \end{aligned} \tag{2.5}$$

where  $\mathcal{X} \subset \mathbb{R}^n$  is a set with additional constraints on the here-and-now decisions. The wait-and-see variable  $y$  has dimension  $k$  and we denote the number of constraints in the model by  $m$ . So,  $B \in \mathbb{R}^{m \times k}$ . Furthermore, we have  $c \in \mathbb{R}^n$ ,  $A \in \mathbb{R}^{m \times n}$ ,  $R \in \mathbb{R}^{m \times L}$  and  $r \in \mathbb{R}^m$ . The matrix  $R$  is chosen constant in this model, so the model only has uncertainty on the

right-hand side. Furthermore,  $\mathcal{U} := \{\zeta \in \mathbb{R}^L \mid D\zeta \leq d, \zeta \geq 0\}$  is the polyhedral uncertainty set. The main contributions of this part come from the following dual formulation of (2.5).

$$\begin{aligned} \min_{x \in \mathcal{X}} \quad & c^\top x \\ \text{s.t.} \quad & \forall w \in \mathcal{V}, \exists \lambda \geq 0 : \begin{cases} w^\top (Ax - r) - d^\top \lambda \geq 0, \\ D^\top \lambda \geq R^\top w, \end{cases} \end{aligned} \quad (2.6)$$

where  $\mathcal{V} = \{w \geq 0 : B^\top w \leq 0, e^\top w = 1\}$ . The above formulation is obtained by dual of (2.5) over  $y$ , and then over  $\zeta$ , and exploits the scalar multiplicity of the adjustable dual variable  $\lambda$ . The dual formulation (2.6) is again an adjustable linear optimization model. Therefore, all existing solution approaches for adjustable models can be used to solve or approximate the dual formulation. The new dual model differs from the primal formulation in its dimension and uses a different description of the uncertainty set. The relation between the optimal primal affine policy can be directly obtained from the optimal affine policy in the dual formulation. The dual formulation can reduce computational time to solve the original problem.

### 2.2.2 Column-and-Constraint Generation

The approach of column-and-constraint generation was initially presented in [126] to address the challenges of solving adjustable robust linear optimization problems.

Consider the following linear adjustable robust problem

$$\begin{aligned} \min_{x \in \mathcal{X}} \quad & \max_{u \in \mathcal{U}} \quad \min_y \quad a^\top x + b^\top y \\ \text{s.t.} \quad & Ax + By + Cu \geq c, \\ & y \geq 0. \end{aligned} \quad (2.7)$$

where  $a \in \mathbb{R}^n, b \in \mathbb{R}^m, c \in \mathbb{R}^r, A \in \mathbb{R}^{r \times n}, B \in \mathbb{R}^{r \times m}, C \in \mathbb{R}^{r \times l}$  are the problem's parameters and  $\mathcal{U}$  is a polytope of the form  $\{u \in \mathbb{R}^l : Du \leq d, u \geq 0\}$  such that  $D \in \mathbb{R}^{q \times l}$  and  $d \in \mathbb{R}^q$ . In general, the column-and-constraint generation algorithm is used to find an optimal solution to linearly adjustable problems. Based on a finite number of uncertainty realizations, it finds both a

lower bound on the objective value and the corresponding here-and-now decision  $x$ , as well as an upper bound on the wait-and-see objective if  $x$  is known.

Before discussing the full algorithm, here we shortly present the optimization problem that will be used to obtain the lower bound in the first part of the algorithm. Given  $x$ , for any  $V = \{u^1, \dots, u^k\} \subseteq U$ , a lower bound for the second-stage cost of (2.7) can be obtained using the following optimization problem

$$\begin{aligned} \underline{Z}(x, V) = & \min_{\theta, \{y^s\}_{1 \leq s \leq k}} a^\top x + \theta \\ \text{s.t.} & \quad b^\top y^s \leq \theta, \quad s = 1, \dots, k \\ & \quad Ax + By^s + Cu^s \geq c, \quad s = 1, \dots, k \\ & \quad y^s \geq 0, \quad s = 1, \dots, k. \end{aligned}$$

Notice that given a here-and-now decision  $x$  and an uncertainty parameter  $u$ , the second-stage cost is given by

$$\begin{aligned} \underline{Z}(x, \{u\}) = & \min_y b^\top y \\ \text{s.t.} & \quad Ax + By + Cu \geq c, \\ & \quad y \geq 0. \end{aligned}$$

Given a here-and-now decision  $x$ , an upper bound on the second-stage cost is given by

$$\begin{aligned} \bar{Z}(x) = \max_{u \in \mathcal{U}} \underline{Z}(x, \{u\}) \equiv & a^\top x + \max_{u \in \mathcal{U}} \min_y b^\top y \\ \text{s.t.} & \quad Ax + By + Cu \geq c, \\ & \quad y \geq 0, \end{aligned}$$

and the uncertainty realization that results in this cost is given by

$$\bar{u}(x) \in \arg \max_{u \in \mathcal{U}} \underline{Z}(x, \{u\}).$$

Given these notations, the column-and-constraint generation algorithm is presented below.

### Column-and-Constraint Generation Method

**Input:**  $\mathcal{X}, A, B, C, a, b, c, D, d, \epsilon > 0, u^0 \in \mathcal{U}$

**Initialization:**

$$UB^0 = \infty, LB^0 = -\infty, V^0 = \{u^0\}, k = 0$$

**Repeat**

- Update  $k \leftarrow k + 1$
- Compute  $x^k = \operatorname{argmin}_{x \in X} \underline{Z}(x, V^{k-1})$  and the corresponding lower bound value  
 $LB^k = \underline{Z}(x^k, V^{k-1})$
- Compute  $u^k = \bar{u}(x^k)$  and the corresponding upper bound  $UB^k = \bar{Z}(x^k) \equiv \underline{Z}(x^k, \{u^k\})$ .
- Update  $V^k \leftarrow V^{k-1} \cup \{u^k\}$

**Until:**  $\frac{UB^k - LB^k}{\max(\min(|LB^k|, |UB^k|), 1)} < \epsilon$ .

**Return:**  $x^k, UB^k$ .

This approach sometimes, and in some applications often, produces infeasible here-and-now solutions, even though the problem is feasible. However, this concern was successfully addressed and resolved in a later work, documented in [30].

### 2.2.3 Copositive Approach

It is possible to reformulate the linear adjustable problem as a copositive optimization problem, which in turn leads to tractable, semi-definite-based approximations at least as good as the affine policy under some assumptions [122].

In this section, we consider a robust linear optimization problem with uncertainty on the right-hand side to review this approximation as follows:

$$\begin{aligned} \min_{x \in \mathcal{X}} \max_{u \in \mathcal{U}} \min_y c^\top x + d^\top y \\ \text{s.t. } Ax + By \geq Fu, \end{aligned} \tag{2.8}$$

where  $A \in \mathbb{R}^{m \times n_1}, B \in \mathbb{R}^{m \times n_2}, c \in \mathbb{R}^{n_1}, d \in \mathbb{R}^{n_2}, F \in \mathbb{R}^{m \times k}$  and  $\mathcal{X} \subseteq \mathbb{R}^{n_1}$  is a convex and closed set containing the here-and-know decision  $x$ . The nonempty full-dimensional uncertainty set

$\mathcal{U} \subseteq \mathbb{R}^k$  is convex and compact.

Within (2.8), define

$$\pi(x) := \max_{u \in \mathcal{U}} \min_{y \in \mathbb{R}^{n_2}} \{d^\top y \mid By \geq Fu - Ax\}.$$

The dual of inner minimization is

$$\max_{w \in \mathcal{W}} (Fu - Ax)^\top w,$$

where  $\mathcal{W} := \{w \geq 0 \mid B^\top w = d\}$ . Hence, strong duality for linear programming implies

$$\pi(x) = \max_{(u,w) \in \mathcal{U} \times \mathcal{W}} (Fu - Ax)^\top w.$$

In [122] it has been proved that the copositive representation of the above is

$$\begin{aligned} \pi(x) &= \max \operatorname{tr} \left( (F - Axe_1^\top)^\top Z_{21} \right) \\ &\text{s.t. } \operatorname{diag} (EZE^\top) = 0, \\ &Z \in \text{CPP} (\widehat{\mathcal{U}} \times \mathbb{R}_+^m), \\ &\operatorname{tr} ((g_1 g_1^\top)^\top Z) = 1, \end{aligned} \tag{2.9}$$

where  $\widehat{\mathcal{U}}$  is the full-dimensional homogenization cone of  $\mathcal{U}$ ,  $\text{CPP}(\cdot)$  denote completely positive cone,  $E \in \mathbb{R}^{n_2 \times (k+m)}$  is a matrix with structure  $E = (-de_1^\top \ B^\top)$ ,  $e_1 \in \mathbb{R}^k$  and  $g_1 \in \mathbb{R}^{k+m}$  are the first unit vector, and  $Z \in \mathbb{R}^{(k+m) \times (k+m)}$  is a symmetric matrix with the block structure  $Z = \begin{pmatrix} Z_{11} & Z_{21}^\top \\ Z_{21} & Z_{22} \end{pmatrix}$ . Then applying standard conic duality theory on (2.9) implies copositive relation of original ARO. See [122] for more details.

## 2.3 Conclusion

We conclude this chapter with the last sentence of the survey paper [123]:

*“Finally, adjustable robust nonlinear optimization appears to be very difficult, and since there is almost no literature about this topic, much more research is needed.”*



# CHAPTER 3

## A Dual-Based Cutting Plane Algorithm for Nonlinear Adjustable Robust Optimization

Werner Heisenberg:

*“Uncertainty is NOT I don’t know. It is I can’t know. I am uncertain does not mean I could be certain.”*

### 3.1 Introduction

Most studies in Adjustable Robust Optimization (ARO) problems are focusing on linear and integer-linear [5, 26, 68]; for more additional details, see the survey paper [123]. There are only a few papers devoted to the nonlinear case due to its theoretical and computational challenges. In [113], the authors considered a nonlinear ARO problem with a polytope uncertainty set and proposed a method to solve such problems under some quasi-convexity conditions. ARO models with second-order cone constraints and ellipsoidal uncertainty sets are considered in [38], where the authors show that applying the affine decision rule would result in a semi-definite optimization problem. In [52], the authors considered a nonlinear ARO model with linear uncertainty (the functions are linear in the uncertain parameters and the uncertainty set is a polyhedron), and derive an equivalent ARO problem, which is linear in the wait-and-see decisions.

In this chapter, we show how to use duality to reformulation a general nonlinear ARO problem and solve it. More specifically, the main contribution of our work can be summarized as follows:

- First, we consider a general nonlinear adjustable robust optimization problem. Applying Fenchel’s duality and dualizing over the continuous wait-and-see decisions, we obtain an equivalent static robust optimization reformulation (dual reformulation). Then, we provide conditions under which the dual reformulation is convex on decision variables.
- Second, we show under some conditions that a convex relaxation of the dual reformulation is equivalent to approximating the ARO problem using constant decision rule.
- Finally, we design an algorithm based on the dual reformulation. The algorithm consists of two main phases: In the first phase, we use an alternating method exploiting the structure of the dual reformulation. We show under which conditions, the alternating method converges to a local worst-case scenario within the uncertainty set. In the second phase, we use finite-scenario approach, given the obtained scenarios in the first phase, to find a solution. Given this solution, we find new local worst-case scenarios and repeat this two-phase procedure until satisfying a stopping criterion. Using this algorithm, we have a lower bound on the original problem and obtain a locally robust solution (The term “local” is used as we obtain a robust solution taking a part of the uncertainty set into account). We further improve the lower bound by introducing new cuts. Our computational results show that our algorithm can provide a locally robust solution with an acceptable optimality gap.

The main results of this chapter have been appeared in

- [76] Khademi, A., Marandi, A., & Soleimani-damaneh, M. (2023). A New Dual-Based Cutting Plane Algorithm for Nonlinear Adjustable Robust Optimization. *Journal of Global Optimization*.

The rest of the chapter is organized as follows. We reformulate a general nonlinear adjustable robust problem as a nonlinear static robust counterpart using Fenchel’s duality in Section 3.2. In Section 3.3, we apply a convex relaxation technique on the dual reformulation to obtain an upper bound and show the relationship between the corresponding static robust counterpart of the ARO problem and this relaxation. Finally, we propose a new algorithm in Section 3.4 to construct a lower bound and obtain a locally robust solution. Our numerical

results are presented in Section 3.5.

## 3.2 Dual Reformulation

In this section, we derive the dual formulation of (ARC). In the next theorem, we show how Fenchel duality is used for this goal.

**Theorem 2.** *Let Assumption 1 hold. Also, in (ARC) let us assume that*

$$\forall (x \in \mathcal{X}, u \in \mathcal{U}), \exists y : f_j(x, y, u) < 0, \quad j \in [m]. \quad (3.1)$$

Then, (ARC) is equivalent to the nonlinear static robust counterpart

$$\begin{aligned} & \inf_{x \in \mathcal{X}, \tau} \tau \\ \text{s.t.} \quad & -f_0^{*2}(x, w^0, u) - \sum_{j=1}^m \lambda_j f_j^{*2}\left(x, \frac{w^j}{\lambda_j}, u\right) \leq \tau, \quad \forall \begin{pmatrix} \lambda \\ \{w^j\}_{j=0}^m \\ u \end{pmatrix} \in \mathcal{Z}, \end{aligned} \quad (3.2)$$

where

$$\mathcal{Z} = \left\{ \begin{pmatrix} \lambda \\ \{w^j\}_{j=0}^m \\ u \end{pmatrix} \in \mathbb{R}^{m+n_y(m+1)+n_u} \left| \begin{array}{l} \lambda \geq \mathbf{0}, \quad u \in \mathcal{U}, \quad \sum_{j=0}^m w^j = \mathbf{0}, \\ w^0 \in \text{dom}(f_0^{*2}(x, \cdot, u)), \\ \frac{w^j}{\lambda_j} \in \text{dom}(f_j^{*2}(x, \cdot, u)), \quad j \in [m] \end{array} \right. \right\}.$$

*Proof.* In (ARC), we consider the inner minimization problem over  $y$  for a given  $x \in \mathcal{X}$  and  $u \in \mathcal{U}$ . Because of (3.1) and Assumption 1, we know for the given  $x \in \mathcal{X}$  and  $u \in \mathcal{U}$ , the inner minimization is Slater regular. Therefore, we can apply Fenchel's duality (Definition 1.2.1),

and rewrite (ARC) as follows:

$$\begin{aligned} \inf_{x \in \mathcal{X}} \sup_{u \in \mathcal{U}} \sup_{\lambda, w^j} & -f_0^{*2}(x, w^0, u) - \sum_{j=1}^m \lambda_j f_j^{*2}\left(x, \frac{w^j}{\lambda_j}, u\right) \\ \text{s.t.} \quad & \sum_{j=0}^m w^j = \mathbf{0}, \quad \lambda \geq \mathbf{0}, \\ & w^0 \in \text{dom}(f_0^{*2}(x, \cdot, u)), \quad \frac{w^j}{\lambda_j} \in \text{dom}(f_j^{*2}(x, \cdot, u)), \quad j \in [m]. \end{aligned}$$

Therefore, (ARC) can be reformulated as

$$\begin{aligned} \inf_{x \in \mathcal{X}} \sup_{u \in \mathcal{U}, \lambda, w^j} & -f_0^{*2}(x, w^0, u) - \sum_{j=1}^m \lambda_j f_j^{*2}\left(x, \frac{w^j}{\lambda_j}, u\right) \\ \text{s.t.} \quad & \sum_{j=0}^m w^j = \mathbf{0}, \quad \lambda \geq \mathbf{0}, \\ & w^0 \in \text{dom}(f_0^{*2}(x, \cdot, u)), \quad \frac{w^j}{\lambda_j} \in \text{dom}(f_j^{*2}(x, \cdot, u)), \quad j \in [m]. \end{aligned}$$

Using the definition of  $\mathcal{Z}$  and epigraph reformulation, we may rewrite problem (ARC) as (3.2), which completes the proof.  $\square$

The above theorem shows that a nonlinear adjustable robust optimization can be reformulated as a nonlinear static robust optimization under a Slater condition. In the equivalent dual reformulation (3.2), the uncertain parameters include the dual multipliers (i.e.,  $\lambda, \{w^j\}_{j=0}^m$ ), in addition to the original uncertain parameter  $u$ .

The conjugate functions and their domains can be easily computed for a wide range of convex functions; see, e.g., [101, Table E.1].

In the proof of the above theorem, we did not use the convexity of  $f_j$  functions on  $x$  and their concavity on  $u$ . However, we usually take convex functions on decision variables and concave functions on uncertain parameters to get tractable models. The benefit of the dual reformulation obtained in Theorem 2 is that we can get upper and lower bounds for the optimal objective value of the original model (ARC). Later, in Sections 3.3 and 3.4, we explain how to achieve these goals.

In the following corollary, we derive the formulation of the dual problem for cases where  $f_j(x, y, u)$ ,  $j \in [m_0]$  are separable.

**Corollary 3.2.1.** *Consider the following ARC:*

$$\begin{aligned} \inf_{x \in \mathcal{X}} \sup_{u \in \mathcal{U}} \inf_y \hat{f}_0(x) + h_0(u) + g_0(y) \\ \text{s.t. } \hat{f}_j(x) + h_j(u) + g_j(y) \leq 0, \quad j \in [m]. \end{aligned} \quad (3.3)$$

Let  $g_j$  be proper, convex, and closed in  $y$ . Assume that there exists some  $y$  such that

$$g_j(y) < -\sup_{x \in \mathcal{X}} \hat{f}_j(x) - \sup_{u \in \mathcal{U}} h_j(u), \quad j \in [m].$$

Then, the nonlinear ARC (3.3) is equivalent to the following static RO problem

$$\begin{aligned} \inf_{x \in \mathcal{X}, \tau} \tau \\ \text{s.t. } \sum_{j=0}^m \lambda_j \hat{f}_j(x) + \sum_{j=0}^m \lambda_j h_j(u) - \sum_{j=0}^m \lambda_j g_j^*\left(\frac{w^j}{\lambda_j}\right) \leq \tau, \quad \begin{pmatrix} \lambda \\ \{w^j\}_{j=0}^m \\ u \end{pmatrix} \in \mathcal{P}, \end{aligned}$$

where

$$\mathcal{P} = \left\{ \begin{pmatrix} \lambda \\ \{w^j\}_{j=0}^m \\ u \end{pmatrix} \left| \begin{array}{l} u \in \mathcal{U}, \sum_{j=0}^m w^j = \mathbf{0}, \\ \lambda_0 = 1, \lambda_j \geq 0, \\ \frac{w^j}{\lambda_j} \in \text{dom}(g_j^*), j \in [m_0] \end{array} \right. \right\}.$$

*Proof.* By setting  $f_j(x, y, u) := \hat{f}_j(x) + h_j(u) + g_j(y)$ , for  $\lambda_j > 0$  we have

$$\begin{aligned} \lambda_j f_j^{*2}\left(x, \frac{w^j}{\lambda_j}, u\right) &= \lambda_j \sup_y \left\{ \left(\frac{w^j}{\lambda_j}\right)^\top y - f_j(x, y, u) \right\} \\ &= \sup_y \left\{ (w^j)^\top y - \lambda_j (\hat{f}_j(x) + h_j(u) + g_j(y)) \right\} \\ &= \sup_y \left\{ (w^j)^\top y - \lambda_j \hat{f}_j(x) - \lambda_j h_j(u) - \lambda_j g_j(y) \right\} \\ &= -\lambda_j \hat{f}_j(x) - \lambda_j h_j(u) + \sup_y \left\{ (w^j)^\top y - \lambda_j g_j(y) \right\} \\ &= -\lambda_j \hat{f}_j(x) - \lambda_j h_j(u) + \lambda_j \sup_y \left\{ \left(\frac{w^j}{\lambda_j}\right)^\top y - g_j(y) \right\} \\ &= -\lambda_j \hat{f}_j(x) - \lambda_j h_j(u) + \lambda_j g_j^*\left(\frac{w^j}{\lambda_j}\right). \end{aligned}$$

Furthermore, for  $\lambda_j = 0$  we have

$$\begin{aligned}
 \lambda_j f_j^{*2}\left(x, \frac{w^j}{\lambda_j}, u\right) &= \delta_{\text{dom}((f_j^{*2})^{*2}(x, \cdot, u))}^*(w^j) \\
 &= \delta_{\text{dom}(f_j(x, \cdot, u))}^*(w^j) \\
 &= \delta_{\text{dom}(g_j)}^*(w^j) \\
 &= \delta_{\text{dom}(g_j^*)}^*(w^j),
 \end{aligned}$$

where the first equality follows from the definition of the partial convex perspective, the second equality holds because of the closedness and convexity of  $f_j(x, \cdot, u)$ . Therefore, Theorem 2 and the above equivalences concludes the corollary.  $\square$

A natural question is whether (ARC) (or its equivalent form (3.2)) is convex with respect to  $x$ . In other words, for a given optimal decision rule and a worst-case scenario, whether optimization on  $x$  is convex. The following example shows that the answer to this question is negative in general.

**Example 3.2.1.** Consider an instance of (ARC) with  $m = 1$ ,  $\mathcal{X} = [1, +\infty)$ ,  $\mathcal{U} = [1, 2]$ ,  $f_0(x, y, u) = x^2uy$ , and  $f_1(x, y, u) = -x + \frac{1}{2}y^2 - u$ . The partial conjugate of  $f_0$  and  $f_1$  with respect to their second argument are given by

$$f_0^{*2}(x, w^0, u) = \begin{cases} 0, & w^0 = x^2u, \\ \infty, & w^0 \neq x^2u, \end{cases} \quad f_1^{*2}(x, w^1, u) = x + \frac{(w^1)^2}{2} + u.$$

Thus, the dual reformulation of (ARC) for this example is

$$\begin{aligned}
 \inf_{x \in \mathcal{X}} \quad & \sup_{u \in \mathcal{U}, \lambda, w^0, w^1} -f_0^{*2}(x, w^0, u) - \lambda f_1^{*2}\left(x, \frac{w^1}{\lambda}, u\right) \\
 \text{s.t.} \quad & w^0 + w^1 = 0, \quad \lambda \geq 0, \\
 & w^0 \in \text{dom}(f_0^{*2}(x, \cdot, u)), \quad \frac{w^1}{\lambda} \in \text{dom}(f_1^{*2}(x, \cdot, u)).
 \end{aligned} \tag{3.4}$$

For fixed  $x$  and  $u$ , if  $\lambda = 0$ , then

$$\begin{aligned} -\lambda f_1^{*2}(x, \frac{w^1}{\lambda}, u) &= -\delta_{\text{dom}((f_j^{*2})^{*2}(x, \cdot, u))}(w^1) \\ &= -\delta_{\text{dom}(f_j(x, \cdot, u))}(w^1) \\ &= -\delta_{\mathbb{R}}(w^1) = -\delta_{\{0\}}(w^1). \end{aligned}$$

Thus,  $-\lambda f_1^{*2}(x, \frac{w^1}{\lambda}, u) = -\infty$  when  $w^1 \neq 0$ . Furthermore,  $w^1 = 0$  is infeasible, due to  $w^1 = -w^0 = -x^2u \neq 0$ . So we can ignore  $\lambda = 0$ .

So, (3.4) is equivalent to

$$\begin{aligned} \inf_{x \in \mathcal{X}} \sup_{u \in \mathcal{U}} \sup_{\lambda, w^0, w^1} &-\frac{(w^1)^2}{2\lambda} - \lambda(x + u) \\ \text{s.t.} \quad &w^0 + w^1 = 0, \lambda > 0, w^0 = x^2u. \end{aligned} \quad (3.5)$$

Given  $x$  and  $u$ , the inner supremum can be written as

$$\sup_{\lambda > 0} -\frac{x^4u^2}{2\lambda} - \lambda(x + u), \quad (3.6)$$

as we know  $x \geq 1$  and  $u \geq 1$ . Moreover, the objective function is concave in  $\lambda$ . So, the supremum happens at  $\lambda = \sqrt{\frac{x^4u^2}{2(x+u)}}$ . Hence, (3.5) is equivalent to

$$\inf_{x \in \mathcal{X}} \sup_{u \in \mathcal{U}} -\sqrt{2x^2u}\sqrt{x+u}.$$

Given  $x \geq 1$ , the inner supremum is  $\sup \left\{ -\sqrt{2x^2u}\sqrt{x+u} \mid u \in [1, 2] \right\}$ , whose objective function is decreasing on the given interval  $[1, 2]$ . Therefore,  $u = 1$  is the worst-case scenario, and so

$$\sup_{u \in \mathcal{U}} -\sqrt{2x^2u}\sqrt{x+u} = -\sqrt{2x^2}\sqrt{x+1}.$$

Finally, we get  $\inf_x \left\{ -\sqrt{2x^2}\sqrt{x+1} \mid x \in \mathcal{X} \right\}$ , which is a non-convex problem.  $\square$

**Remark 3.2.1.** Note that

$$f_j^{*2}(x, w^j, u) = \sup_y \left\{ (w^j)^\top y - f_j(x, y, u) \right\}$$

implies

$$-f_j^{*2}(x, w^j, u) = \inf_y \left\{ -(w^j)^\top y + f_j(x, y, u) \right\}.$$

Indeed, the function  $\mathcal{K}_{w^j, y, u}(x) := -(w^j)^\top y + f_j(x, y, u)$  is convex on  $x$  for any  $w^j, y, u$ , but the inf operator breaks down the convexity. In other words, the conjugate function  $-f_j^{*2}(x, w^j, u)$  is not convex in  $x$ . Therefore, based on the dual formulation, we see if the conjugate function  $-f_j^{*2}(x, w^j, u)$  is convex in  $x$ , then the problem is a convex optimization problem in  $x$ . In the next theorem, we show this for separable fixed-recourse problems.  $\square$

**Theorem 3.** *Under the assumption of Theorem 2, if (ARC) is separable fixed-recourse, then the dual reformulation of (ARC) is convex in  $x$ .*

*Proof.* Since  $f_j(x, y, u) = \hat{f}_j(x, u) + g_j(y)$  for all  $j \in [m_0]$ , according to Theorem 2, (ARC) is equivalent to

$$\begin{aligned} \inf_{x \in \mathcal{X}} \quad & \sup_{u \in \mathcal{U}, \lambda_j, w^j} \quad \sum_{j=0}^m \lambda_j \hat{f}_j(x, u) - \sum_{j=0}^m \lambda_j g_j^*\left(\frac{w^j}{\lambda_j}\right) \\ \text{s.t.} \quad & \sum_{j=0}^m w^j = \mathbf{0}, \quad \lambda_0 = 1, \quad \lambda_j \geq 0, \quad j \in [m], \\ & w^0 \in \text{dom}(g_0^*), \quad \frac{w^j}{\lambda_j} \in \text{dom}(g_j^*), \quad j \in [m]. \end{aligned} \quad (3.7)$$

Note that  $\hat{f}_j(x, u)$  is convex on  $x$  for each  $j \in [m_0]$ . By denoting

$$F_{u, \lambda, w^j}(x) := \sum_{j=0}^m \lambda_j \hat{f}_j(x, u) - \sum_{j=0}^m \lambda_j g_j^*\left(\frac{w^j}{\lambda_j}\right), \quad \text{and} \quad \mathcal{F}(x) := \sup_{u, \lambda, w^j} F_{u, \lambda, w^j}(x),$$

which are convex on  $x$ , problem (3.7) is equivalent to  $\inf_{x \in \mathcal{X}} \mathcal{F}(x)$  which is a convex optimization problem.  $\square$

Considering Theorem 3, we focus on separable fixed-recourse case in the rest of the chapter.



### 3.3 On Upper Bound Calculation

In this section, we assume that the non-empty uncertainty set  $\mathcal{U}$  has the following structure:

$$\mathcal{U} := \{u \in \mathbb{R}^{n_u} \mid c_i(u) \leq 0, \ i \in [t]\},$$

where the function  $c_i: \mathbb{R}^{n_u} \rightarrow \bar{\mathbb{R}}$  is proper, closed, and convex for each  $i \in [t]$ .

In the next theorem, we show how using perspective functions result in an upper bound for (3.7).

**Theorem 4.** *For any fixed  $x \in \mathcal{X}$ , let  $\hat{f}_j(x, u)$  be proper and concave in  $u$  for each  $j \in [m_0]$ .*

*Then,*

$$\begin{aligned} & \sup_{u, \lambda_j, w^j, \theta^j} \sum_{j=0}^m \lambda_j \hat{f}_j(x, \frac{\theta^j}{\lambda_j}) - \sum_{j=0}^m \lambda_j g_j^*(\frac{w^j}{\lambda_j}) \\ \text{s.t.} \quad & \sum_{j=0}^m w^j = \mathbf{0}, \ \lambda_0 = 1, \ \theta^0 = u, \ \lambda_j \geq 0, \ j \in [m], \\ & w^0 \in \text{dom}(g_0^*), \ \frac{w^j}{\lambda_j} \in \text{dom}(g_j^*), \ j \in [m], \\ & \lambda_j c_i(\frac{\theta^j}{\lambda_j}) \leq 0, \ j \in [m_0], i \in [t], \end{aligned} \tag{3.8}$$

*provides an upper bound on the optimal value of*

$$\begin{aligned} & \sup_{u, \lambda_j, w^j} \sum_{j=0}^m \lambda_j \hat{f}_j(x, u) - \sum_{j=0}^m \lambda_j g_j^*(\frac{w^j}{\lambda_j}) \\ \text{s.t.} \quad & \sum_{j=0}^m w^j = \mathbf{0}, \ \lambda_0 = 1, \ \lambda_j \geq 0, \ j \in [m], \\ & w^0 \in \text{dom}(g_0^*), \ \frac{w^j}{\lambda_j} \in \text{dom}(g_j^*), \ j \in [m], \\ & c_i(u) \leq 0, \ i \in [t]. \end{aligned} \tag{3.9}$$

*Proof.* We show that any feasible solution to (3.9) corresponds to a feasible solution to (3.8) with the same objective value. For this goal, for any solution  $(u, \lambda, \{w^j\}_{j=0}^m)$ , define  $\theta^j := \lambda_j u$  for each  $j \in [m]$ . We prove that  $(u, \lambda, \{w^j\}_{j=0}^m, \{\theta^j\}_{j=1}^m)$  is a feasible solution to (3.8). For each  $j \in [m]$  and  $i \in [t]$  with  $\lambda_j > 0$  obviously we have  $\lambda_j c_i(\frac{\theta^j}{\lambda_j}) \leq 0$ . If  $\lambda_j = 0$  for some  $j \in [m]$ , we get  $\theta^j = \mathbf{0}$  and

$$0c_i(\mathbf{0}/0) = \delta_{\text{dom}(c_i^*)}^*(\mathbf{0}) = 0, \quad \forall i \in [t],$$

where the first equality follows from the definition of the convex perspective, and the second equality holds because the conjugate function of a proper convex function is also proper (from [9, Theorem 4.5]), and so  $\text{dom}(c_i^*) \neq \emptyset$ . All other constraints of (3.8) are clearly satisfied. Now, we show that the objective value at  $(u, \lambda, \{w^j\}_{j=0}^m)$  in (3.9) equals to that value at  $(u, \lambda, \{w^j\}_{j=0}^m, \{\theta^j\}_{j=1}^m)$  in (3.8). To this end, it is sufficient to show that  $\lambda_j \hat{f}_j(x, \frac{\theta^j}{\lambda_j}) = \lambda_j \hat{f}_j(x, u)$  for all  $j \in [m]$ . It is trivial for the case  $\lambda_j > 0$ . If  $\lambda_j = 0$  for some  $j \in [m]$ , then  $\theta^j = \mathbf{0}$ , and so

$$0 \hat{f}_j(x, \mathbf{0}/0) = -\delta_{\text{dom}((- \hat{f}_j)^{*2}(x, \cdot))}^*(\mathbf{0}) = 0 = 0 \hat{f}_j(x, u),$$

where the first equality follows from the definition of the partial concave perspective, and the second equality holds because (for any  $x$ ) the partial conjugate  $(- \hat{f}_j)^{*2}(x, \cdot)$  of proper convex function  $- \hat{f}_j(x, \cdot)$  is also proper, leading to  $\text{dom}((- \hat{f}_j)^{*2}(x, \cdot)) \neq \emptyset$ .  $\square$

Problem (3.9) is not a convex programming in general, while problem (3.8) is. More specifically, by lifting the problem to a higher dimension and using the perspective functions, we obtain a concave relaxation on  $\lambda_j, w^j, \theta^j$ . This approach has been recently used in the literature of nonlinear optimization for other purposes [50, 81, 117]. In the next example, we show that this relaxation may not be tight.

**Example 3.3.1.** Let  $x \in \mathcal{X}$ . Consider an instance of problem (3.9) with  $t = m = 1$ ,  $n_u = 1$ ,  $n_y = 2$ ,  $\hat{f}_0(x, u) = -u^2$ ,  $\hat{f}_1(x, u) = \frac{1}{u}$ ,  $c_1(u) = u + 1$ ,  $g_0(y) = y_1$ , and  $g_1(y) = \frac{1}{2}y^\top y + y_2$ . Set  $p^0 := (1, 0)^\top$ ,  $p^1 := (0, 1)^\top$ . The conjugates of  $g_0$  and  $g_1$  are given by

$$g_0^*(w^0) = \begin{cases} 0, & w^0 = p^0, \\ \infty, & w^0 \neq p^0, \end{cases}$$

$$g_1^*(w^1) = \frac{1}{2} (w^1 - p^1)^\top (w^1 - p^1).$$

Hence, problem (3.9) in this example reads as

$$\begin{aligned} & \sup_{u, \lambda_1, w^0, w^1} \hat{f}_0(x, u) + \lambda_1 \hat{f}_1(x, u) - g_0^*(w^0) - \lambda_1 g_1^*\left(\frac{w^1}{\lambda_1}\right) \\ \text{s.t.} \quad & \lambda_1 \geq 0, \quad w^0 + w^1 = \mathbf{0}, \quad u \leq -1, \\ & w^0 = p^0. \end{aligned}$$

If  $\lambda_1 = 0$  in some feasible solution of the above problem, then

$$-\lambda_1 g_1^*\left(\frac{w^1}{\lambda_1}\right) = -\delta_{\text{dom}(g_1)}^*(w^1) = -\delta_{\{0\}}(w^1).$$

So,  $-\lambda_1 g_1^*\left(\frac{w^1}{\lambda_1}\right) = -\infty$  when  $w^1 \neq \mathbf{0}$ . Furthermore,  $w^1 = \mathbf{0}$  is infeasible, due to  $w^1 = -w^0 = -p^0 = (-1, 0)^\top$ . Hence, we can ignore  $\lambda_1 = 0$ . Now, due to  $w^0 = p^0$  and  $w^0 + w^1 = \mathbf{0}$ , the last problem can be rewritten as

$$\begin{aligned} z_1 & := \sup_{u, \lambda_1} -u^2 + \frac{\lambda_1}{u} - \frac{1}{2} \left( \lambda_1 + \frac{1}{\lambda_1} \right) \\ \text{s.t.} \quad & \lambda_1 > 0, \quad u \leq -1. \end{aligned}$$

Let us denote the objective function of the last problem by

$$J(u, \lambda_1) = -u^2 + \frac{\lambda_1}{u} - \frac{1}{2} \left( \lambda_1 + \frac{1}{\lambda_1} \right).$$

This function is bounded above over the feasible set  $\mathcal{K} = \{(u, \lambda_1) \mid u \leq -1, \lambda_1 > 0\}$ . To obtain  $z_1$ , first we examine the points for which the gradient of  $J(\cdot, \cdot)$  vanishes. We have

$$\nabla J(u, \lambda_1) = \left( -2u - \frac{\lambda_1}{u^2}, \frac{1}{u} - \frac{1}{2} + \frac{1}{2\lambda_1^2} \right)^\top.$$

Thus,  $\nabla J(u, \lambda_1) = 0$  implies  $\lambda_1 = -2u^3$ , and  $-4u^6 + 8u^5 + 1 = 0$ . On the other hand,  $u \leq -1$  (feasibility) leads  $-4u^6 + 8u^5 + 1 < 0$ . So, the maximizers of  $J(u, \lambda_1)$  are not in the interior of the feasible set  $\mathcal{K}$ . They are on the boundary of  $\mathcal{K}$ , i.e.,  $\lambda_1 = 0$  or  $u = -1$ . As  $\lambda_1 > 0$ , we

continue with  $u = -1$ , and we have

$$\sup_{\lambda_1 > 0} -1 - \lambda_1 - \frac{1}{2} \left( \lambda_1 + \frac{1}{\lambda_1} \right).$$

The optimal solution of the above problem occurs at  $\lambda_1 = \frac{1}{\sqrt{3}}$ , and hence  $z_1 = -1 - \sqrt{3}$ .

Analogously, problem (3.8) in this example is

$$\begin{aligned} z_2 := \sup_{u, \lambda_1, \theta_1} & -u^2 + \frac{(\lambda_1)^2}{\theta_1} - \frac{1}{2} \left( \lambda_1 + \frac{1}{\lambda_1} \right) \\ \text{s.t.} & \quad \lambda_1 > 0, \quad u \leq -1, \quad \theta_1 \leq -\lambda_1. \end{aligned}$$

It is not difficult to see that  $-2$  is an upper bound for the objective function of the above problem on its feasible region. Furthermore, the objective value at the feasible sequence  $\{u_n = -1, (\lambda_1)_n = 1, (\theta_1)_n = -n\}_{n \geq 1}$  equals to  $-2 - \frac{1}{n}$  which goes to  $-2$  as  $n \rightarrow +\infty$ . This implies  $z_2 = -2$ . Therefore,  $z_1 = -1 - \sqrt{3} < -2 = z_2$ .  $\square$

In Theorem 4,  $x \in \mathcal{X}$  is fixed and arbitrary. Now, by taking minimum over all  $x \in \mathcal{X}$  in (3.9) and (3.8), we obtain an upper bound for the separable fixed-recourse version of the dual reformulation of (ARC) as follow:

$$\begin{aligned} \inf_{x \in \mathcal{X}} \quad & \sup_{u, \lambda_j, w^j, \theta^j} \sum_{j=0}^m \lambda_j \hat{f}_j(x, \frac{\theta^j}{\lambda_j}) - \sum_{j=0}^m \lambda_j g_j^*(\frac{w^j}{\lambda_j}) \\ \text{s.t.} \quad & \sum_{j=0}^m w^j = \mathbf{0}, \quad \lambda_0 = 1, \quad \theta^0 = u, \quad \lambda_j \geq 0, \quad j \in [m], \\ & w^0 \in \text{dom}(g_0^*), \quad \frac{w^j}{\lambda_j} \in \text{dom}(g_j^*), \quad j \in [m], \\ & \lambda_j c_i(\frac{\theta^j}{\lambda_j}) \leq 0, \quad j \in [m_0], \quad i \in [t]. \end{aligned} \tag{PERS}$$

We call this problem (PERS) as it is obtained by using a *perspectification approach* corresponding to problem (3.7). Moreover, (PERS) is a convex-concave programming, while it is not the case for (3.7).

Problem (PERS) can be seen as a relaxation of the dual reformulation of (ARC) when it has fixed-recourse. So, it is important to know the interpretation of such a relaxation for the primal problem, i.e., (ARC). The next theorem shows that (PERS) is actually equivalent to

the static robust counterpart (RC) in the separable fixed-recourse case when the uncertainty set is compact.

**Theorem 5.** Consider (RC) with separable fixed-recourse as follows:

$$\begin{aligned} \inf_{x \in \mathcal{X}, y} \quad & \sup_{u \in \mathcal{U}} \hat{f}_0(x, u) + g_0(y) \\ \text{s.t.} \quad & \sup_{u \in \mathcal{U}} \hat{f}_j(x, u) + g_j(y) \leq 0, \quad j \in [m]. \end{aligned} \quad (3.10)$$

Suppose that the uncertainty set  $\mathcal{U}$  is compact,  $\hat{f}_j$  is proper concave in  $u$ , and  $g_j$  is closed convex and real-valued, for each  $j \in [m_0]$ . If

$$\forall x \in \mathcal{X} \exists y \text{ such that } \sup_{u \in \mathcal{U}} \hat{f}_j(x, u) + g_j(y) < 0, \quad j \in [m],$$

then (PERS) and static robust counterpart (3.10) are equivalent.

*Proof.* Without loss of generality, since  $\mathcal{U}$  is compact, we assume that there exists some  $i$ , for which  $c_i(u) = \|u\|_2 - \rho$ , for some  $\rho > 0$ . By setting  $F_j(x) := \sup_u \{\hat{f}_j(x, u) \mid u \in \mathcal{U}\}$  for each  $j \in [m_0]$  and  $x \in \mathcal{X}$ , we can rewrite (3.10) as

$$\begin{aligned} \inf_{x \in \mathcal{X}, y \in \mathbb{R}^{n_y}} \quad & F_0(x) + g_0(y) \\ \text{s.t.} \quad & F_j(x) + g_j(y) \leq 0, \quad j \in [m]. \end{aligned} \quad (3.11)$$

By applying the Fenchel's duality over  $y$ , (3.11) is equivalent to

$$\begin{aligned} \inf_{x \in \mathcal{X}} \quad & \sup_{\lambda_j, w^j} \sum_{j=0}^m \lambda_j F_j(x) - \sum_{j=0}^m \lambda_j g_j^* \left( \frac{w^j}{\lambda_j} \right) \\ \text{s.t.} \quad & \sum_{j=0}^m w^j = \mathbf{0}, \quad \lambda_0 = 1, \quad \lambda_j \geq 0, \quad j \in [m], \\ & w^0 \in \text{dom}(g_0^*), \quad \frac{w^j}{\lambda_j} \in \text{dom}(g_j^*), \quad j \in [m]. \end{aligned} \quad (\text{RC-1})$$

Now, we show that, for a given  $x \in \mathcal{X}$ , the inner suprimums of (RC-1) and (PERS) have the same optimal value. To prove this claim, let  $x \in \mathcal{X}$  be fixed. Let  $(u, \lambda, \{w^j\}_{j=0}^m, \{\theta^j\}_{j=1}^m)$  be a feasible solution for the inner suprimum in (PERS).

If  $\lambda_j > 0$ , then

$$\begin{aligned} \lambda_j c_i\left(\frac{\theta^j}{\lambda_j}\right) \leq 0, \quad \forall i \in [t] &\Rightarrow c_i\left(\frac{\theta^j}{\lambda_j}\right) \leq 0, \quad \forall i \in [t] \\ &\Rightarrow \frac{\theta^j}{\lambda_j} \in \mathcal{U} \Rightarrow \hat{f}_j\left(x, \frac{\theta^j}{\lambda_j}\right) \leq F_j(x) \\ &\Rightarrow \lambda_j \hat{f}_j\left(x, \frac{\theta^j}{\lambda_j}\right) \leq \lambda_j F_j(x). \end{aligned}$$

If  $\lambda_j = 0$ , then  $\theta^j = \mathbf{0}$ . To prove this, as  $c_i(u) = \|u\|_2 - \rho$  for some  $\rho > 0$  and some  $i \in [t]$ , by taking  $\lambda_j = 0$  into account, we have

$$0 \geq \lambda_j c_i\left(\frac{\theta^j}{\lambda_j}\right) = \delta_{\text{dom}(c_i^*)}^*(\theta^j) = \sup_{\|\gamma\|_2 \leq 1} \{\gamma^\top \theta^j\} = \|\theta^j\|_2 \geq 0.$$

This implies  $\theta^j = \mathbf{0}$ . Hence, in this case

$$\lambda_j \hat{f}_j\left(x, \frac{\theta^j}{\lambda_j}\right) = -\delta_{\text{dom}((-f_j)^{*2}(x, \cdot))}^*(\mathbf{0}) = 0 = \lambda_j F_j(x),$$

where first equality comes from the definition of the partial concave perspective, and the second equality holds as  $\text{dom}((-f_k)^{*2}(x, \cdot)) \neq \emptyset$ . So,

$$\lambda_j \hat{f}_j\left(x, \frac{\theta^j}{\lambda_j}\right) \leq \lambda_j F_j(x), \quad j \in [m_0].$$

Summing over  $j$  yields  $\sum_j \lambda_j \hat{f}_j\left(x, \frac{\theta^j}{\lambda_j}\right) \leq \sum_j \lambda_j F_j(x)$ . Thus,

$$\sum_{j=0}^m \lambda_j \hat{f}_j\left(x, \frac{\theta^j}{\lambda_j}\right) - \sum_{j=0}^m \lambda_j g_j^*\left(\frac{w^j}{\lambda_j}\right) \leq \sum_{j=0}^m \lambda_j F_j(x) - \sum_{j=0}^m \lambda_j g_j^*\left(\frac{w^j}{\lambda_j}\right).$$

Therefore, the optimal value of the objective function of the inner supremum in (PERS) is less than or equal to that in (RC-1).

Conversely, let  $(\bar{\lambda}, \{\bar{w}^j\}_{j=0}^m)$  be a feasible solution for inner supremum of (RC-1). By choosing

$$\bar{u}^j \in \text{argmax}\{\hat{f}_j(x, u) \mid u \in \mathcal{U}\}, \quad j \in [m_0],$$

and setting

$$\bar{\theta}^j = \begin{cases} \bar{\lambda}_j \bar{u}^j, & \lambda_j > 0, \\ 0, & \lambda_j = 0, \end{cases} \quad j \in [m],$$

and then setting  $\bar{\theta}^0 = \bar{u}^0$ , the vector  $(\bar{u}^0, \bar{\lambda}, \{\bar{w}^j\}_{j=0}^m, \{\bar{\theta}^j\}_{j=1}^m)$  is feasible for (PERS). Furthermore, for  $\bar{\lambda}_j > 0$ ,

$$\bar{\lambda}_j F_j(x) = \bar{\lambda}_j \sup_{u \in \mathcal{U}} \hat{f}_j(x, u) = \bar{\lambda}_j \hat{f}_j(x, \bar{u}^j) = \bar{\lambda}_j \hat{f}_j(x, \frac{\bar{\theta}^j}{\bar{\lambda}_j}).$$

This equality is trivial for  $\bar{\lambda}_j = 0$ . Hence,

$$\sum_{j=0}^m \bar{\lambda}_j F_j(x) - \sum_{j=0}^m \bar{\lambda}_j g_j^*(\frac{\bar{w}^j}{\bar{\lambda}_j}) = \sum_{j=0}^m \bar{\lambda}_j \hat{f}_j(x, \frac{\bar{\theta}^j}{\bar{\lambda}_j}) - \sum_{j=0}^m \lambda_j g_j^*(\frac{\bar{w}^j}{\bar{\lambda}_j}).$$

This implies that the optimal value of the objective function of (RC-1) is less than or equal to that in (PERS). This completes the proof.  $\square$

Theorem 5 states that, under some assumptions, the upper bound obtained based on the perspective relaxation of the dual reformulation of (ARC) is the same as the robust counterpart, which is a conservative approximation. In other words, the perspectification approach yields an upper bound for (ARC); nevertheless, there are stronger upper bounds in the literature of adjustable robust optimization, such as *K-adaptability* or *finite adaptability* approaches [97, 112], which can straightforwardly be extended to nonlinear problems. One way to obtain a stronger upper bound is by applying a piece-wise constant decision rule to (ARC) using finite adaptability approach. In this approach, the uncertainty set is partitioned into subsets and a constant decision rule is obtained for each of the subsets. In the numerical experiments, we show how much stronger the upper bound obtained by finite adaptability compared to the one obtained from (PERS).

### 3.4 Lower Bound Calculation

In Section 3.2, we showed that the dual reformulation of (ARC) in the fixed-resource case is a convex programming on here-and-now decision variables. We have also shown how to

construct an upper bound. In this section, we design methods to approximate (ARC) from below. Let us set

$$v := \begin{pmatrix} \lambda \\ \{w^j\}_{j=0}^m \end{pmatrix} \in \mathbb{R}^m \times \mathbb{R}^{n_y(m+1)},$$

$$G(v) := -g_0^*(w^0) - \sum_{j=1}^m \lambda_j g_j^*\left(\frac{w^j}{\lambda_j}\right),$$

$$\mathcal{V} := \left\{ v = \begin{pmatrix} \lambda \\ \{w^j\}_{j=0}^m \end{pmatrix} \left| \begin{array}{l} \sum_{j=0}^m w^j = \mathbf{0}, \lambda \geq \mathbf{0}, \\ w^0 \in \text{dom}(g_0^*), \\ \frac{w^j}{\lambda_j} \in \text{dom}(g_j^*), j \in [m] \end{array} \right. \right\}.$$

Since  $\lambda_j g_j^*\left(\frac{w^j}{\lambda_j}\right)$  is jointly convex in  $(w^j, \lambda_j)$ , the set  $\mathcal{V}$  is convex and  $G$  is a concave function. The next proposition proves the convexity of the set  $\mathcal{V}$  and the concavity of the function  $G$ .

**Proposition 3.4.1.** *The set  $\mathcal{V}$  is convex, and  $G$  is a concave function on  $\mathcal{V}$ .*

*Proof.* We consider two points  $\bar{v} = \begin{pmatrix} \bar{\lambda} \\ \{\bar{w}^j\}_{j=0}^m \end{pmatrix}, \tilde{v} = \begin{pmatrix} \tilde{\lambda} \\ \{\tilde{w}^j\}_{j=0}^m \end{pmatrix} \in \mathcal{V}$  and  $\ell \in [0, 1]$ . Since  $\frac{\bar{w}^j}{\bar{\lambda}_j}, \frac{\tilde{w}^j}{\tilde{\lambda}_j} \in \text{dom}(g_j^*)$ , and  $\lambda_j g_j^*\left(\frac{w^j}{\lambda_j}\right)$  for each  $j$  is jointly convex in  $(w^j, \lambda_j)$ , we have the following possible cases:

**Case 1.**  $\ell \bar{\lambda}_j + (1 - \ell) \tilde{\lambda}_j > 0$ : In this case,

$$\begin{aligned} (\ell \bar{\lambda}_j + (1 - \ell) \tilde{\lambda}_j) g_j^*\left(\frac{\ell \bar{w}^j + (1 - \ell) \tilde{w}^j}{\ell \bar{\lambda}_j + (1 - \ell) \tilde{\lambda}_j}\right) &\leq \ell \bar{\lambda}_j g_j^*\left(\frac{\bar{w}^j}{\bar{\lambda}_j}\right) + (1 - \ell) \tilde{\lambda}_j g_j^*\left(\frac{\tilde{w}^j}{\tilde{\lambda}_j}\right) < \infty \\ \Rightarrow \frac{\ell \bar{w}^j + (1 - \ell) \tilde{w}^j}{\ell \bar{\lambda}_j + (1 - \ell) \tilde{\lambda}_j} &\in \text{dom}(g_j^*) \end{aligned}$$

**Case 2.**  $\ell \bar{\lambda}_j + (1 - \ell) \tilde{\lambda}_j = 0$ : In this case, if  $0 < \ell < 1$ , then  $\bar{\lambda}_j = 0 = \tilde{\lambda}_j$ , and so

$$\begin{aligned} (\ell \bar{\lambda}_j + (1 - \ell) \tilde{\lambda}_j) g_j^*\left(\frac{\ell \bar{w}^j + (1 - \ell) \tilde{w}^j}{\ell \bar{\lambda}_j + (1 - \ell) \tilde{\lambda}_j}\right) &= \delta_{\text{dom}(g_j)}^*(\ell \bar{w}^j + (1 - \ell) \tilde{w}^j) \\ &\leq \delta_{\text{dom}(g_j)}^*(\ell \bar{w}^j) + \delta_{\text{dom}(g_j)}^*((1 - \ell) \tilde{w}^j) \\ &< \infty. \end{aligned}$$



If  $\ell = 0$ , then  $\tilde{\lambda}_j = 0$ , and hence

$$\begin{aligned} (\ell\bar{\lambda}_j + (1-\ell)\tilde{\lambda}_j)g_j^* \left( \frac{\ell\bar{w}^j + (1-\ell)\tilde{w}^j}{\ell\bar{\lambda}_j + (1-\ell)\tilde{\lambda}_j} \right) &= \delta_{dom(g_j)}^*(\ell\bar{w}^j + (1-\ell)\tilde{w}^j) \\ &= \delta_{dom(g_j)}^*(\tilde{w}^j) < \infty. \end{aligned}$$

If  $\ell = 1$ , then  $\bar{\lambda}_j = 0$ , and thus

$$\begin{aligned} (\ell\bar{\lambda}_j + (1-\ell)\tilde{\lambda}_j)g_j^* \left( \frac{\ell\bar{w}^j + (1-\ell)\tilde{w}^j}{\ell\bar{\lambda}_j + (1-\ell)\tilde{\lambda}_j} \right) &= \delta_{dom(g_j)}^*(\ell\bar{w}^j + (1-\ell)\tilde{w}^j) \\ &= \delta_{dom(g_j)}^*(\bar{w}^j) < \infty \end{aligned}$$

So, in all above three cases, we get

$$\begin{aligned} (\ell\bar{\lambda}_j + (1-\ell)\tilde{\lambda}_j)g_j^* \left( \frac{\ell\bar{w}^j + (1-\ell)\tilde{w}^j}{\ell\bar{\lambda}_j + (1-\ell)\tilde{\lambda}_j} \right) &= \delta_{dom(g_j)}^*(\ell\bar{w}^j + (1-\ell)\tilde{w}^j) < \infty \\ \Rightarrow \frac{\ell\bar{w}^j + (1-\ell)\tilde{w}^j}{\ell\bar{\lambda}_j + (1-\ell)\tilde{\lambda}_j} &\in dom(g_j^*). \end{aligned}$$

Convexity in all other constraints of  $\mathcal{V}$  obviously holds. So,  $\ell\bar{v} + (1-\ell)\tilde{v} \in \mathcal{V}$  which shows that  $\mathcal{V}$  is a convex set. The function  $G$  on the convex set  $\mathcal{V}$  is a concave function due to the concavity of each  $-\lambda_j g_j^* \left( \frac{w^j}{\lambda_j} \right)$ .  $\square$

Also, let us set

$$\begin{aligned} F(x, u) &:= (\hat{f}_0(x, u), \dots, \hat{f}_m(x, u), 0, \dots, 0)^\top \in \mathbb{R}^{m+1} \times \mathbb{R}^{n_y(m+1)}, \\ \mathcal{L}(x, u, v) &:= (1, v^\top)F(x, u) + G(v). \end{aligned}$$

Thus, the dual formulation of (ARC) in the separable fixed-recourse case reads as

$$\inf_{x \in \mathcal{X}} \sup_{\substack{u \in \mathcal{U} \\ v \in \mathcal{V}}} \mathcal{L}(x, u, v). \quad (3.12)$$

Given  $\bar{x} \in \mathcal{X}$ , we define  $\mathcal{L}_{\bar{x}}(u, v) := (1, v^\top)F(\bar{x}, u) + G(v)$ . Clearly,  $\mathcal{L}_{\bar{x}}(u, v)$  is concave in  $u$

and concave in  $v$ . Therefore,

$$\sup \{ \mathcal{L}_{\bar{x}}(u, v) \mid u \in \mathcal{U}, v \in \mathcal{V} \}, \quad (3.13)$$

is a disjoint biconcave maximization problem. A common way to find a solution for such problems is by using alternating methods, which obtain a local optimizer. In these methods, a decision variable is divided into several blocks, and optimization can be performed explicitly in each block when the variables of other blocks are fixed (see Chapter 14 of [9] for more details). These methods also appears in the literature as block coordinate methods. The performance of the alternating method is closely related to finding the optimizers for each block.

### 3.4.1 Alternating Iterative Algorithm

As was mentioned above, we use a two-block alternating method to solve (3.13). In this method, we alternatively fix  $u$  to find  $v$  and fix  $v$  to find  $u$  until no improvement is achieved or the prescribed computational limit is reached. This method is described follow in detail.

#### Alternating Method

**Input:** initial value  $\bar{u}^{(0)} \in \mathcal{U}$

**Initialization:**

Set iteration counter  $k \leftarrow 0$ , choose  $\bar{v}^{(0)} \in \operatorname{argmax}_{v \in \mathcal{V}} \mathcal{L}_{\bar{x}}(\bar{u}^{(0)}, v)$ .

**Repeat**

Find optimal  $u$ :  $\bar{u}^{(k+1)} \in \operatorname{argmax}_{u \in \mathcal{U}} \mathcal{L}_{\bar{x}}(u, \bar{v}^{(k)})$ ,

Find optimal  $v$ :  $\bar{v}^{(k+1)} \in \operatorname{argmax}_{v \in \mathcal{V}} \mathcal{L}_{\bar{x}}(\bar{u}^{(k+1)}, v)$ ,

Update iteration counter  $k \leftarrow k + 1$ ,

**Until:** time limit is reached, or no improvement is possible.

**Return:**  $(\bar{u}^{(k)}, \bar{v}^{(k)})$ .

In Theorem 6 below, we discuss the convergence of the addressed alternating method. It is done assuming some appropriate conditions, under which the alternating method is well-defined and the sequence  $\{(u^k, v^k)\}_{k \geq 0}$  admits limit point(s).

**Theorem 6.** *Let  $\mathcal{L}_{\bar{x}}(\cdot, \cdot)$  be continuously differentiable and bounded above on Cartesian product of two closed convex sets  $\mathcal{U}$  and  $\mathcal{V}$ . Suppose that every sub-problem of the alternating method has an optimal solution and  $\{z^k \equiv (u^k, v^k)\}_{k \geq 0}$ , as the sequence generated by the alternating method, has at least a limit point. Then, every limit point of  $\{z^k\}_{k \geq 0}$  is a stationary point of problem (3.13).*

We first recall optimality condition for a constrained differentiable problem (for more details see e.g., [10]). Consider a (non-convex) problem of the form

$$\sup_y \{g(y) \mid y \in \mathcal{S}\}, \quad (3.14)$$

where  $g$  is a real-valued continuously differentiable function, and  $\mathcal{S}$  is a nonempty closed convex set. A vector  $y^* \in \mathcal{S}$  is called a *stationary point* of problem (3.14) if

$$\nabla g(y^*)^\top (y - y^*) \leq 0, \quad \forall y \in \mathcal{S},$$

where  $\nabla g(y^*)$  is the gradient of  $g$  at  $y^*$ .

**Lemma 6.1.** *Let  $g$  be a real-valued continuously differentiable function defined on the Cartesian product of two closed convex sets  $C_1 \subseteq \mathbb{R}^{n_1}$ ,  $C_2 \subseteq \mathbb{R}^{n_2}$ . Suppose that  $\bar{y} = (\bar{y}^1, \bar{y}^2) \in C_1 \times C_2$ . Then*

$$\nabla g(\bar{y})^\top (y - \bar{y}) \leq 0, \quad \forall y \in C_1 \times C_2, \quad (3.15)$$

*if and only if the following properties hold:*

- (i)  $\nabla_1 g(\bar{y})^\top (y^1 - \bar{y}^1) \leq 0, \quad \forall y^1 \in C_1,$
- (ii)  $\nabla_2 g(\bar{y})^\top (y^2 - \bar{y}^2) \leq 0, \quad \forall y^2 \in C_2,$

*where the vector  $y$  is partitioned into two component vectors  $y^1 \in \mathbb{R}^{n_1}$ ,  $y^2 \in \mathbb{R}^{n_2}$ , as  $y \equiv (y^1, y^2)$ , and  $\nabla_1 g(\bar{y}) = \left( \frac{\partial g}{\partial y^1}(\bar{y}) \right)$ , and  $\nabla_2 g(\bar{y}) = \left( \frac{\partial g}{\partial y^2}(\bar{y}) \right)$  denote the corresponding gradient vectors.*

*Proof.* ( $\Rightarrow$ ) Let  $y = (y^1, y^2) \in C_1 \times C_2$ . By setting  $y := (y^1, \bar{y}^2)$  and  $y := (\bar{y}^1, y^2)$  in inequality (3.15), inequalities (i) and (ii) are derived.

( $\Leftarrow$ ) Clearly, (i) and (ii) lead (3.15). □

Now we are ready to prove Theorem 6. The main line of reasoning can be found in [64] but given here for completeness.

**Proof of Theorem 6.** Suppose that  $z^* = (z^{1*}, z^{2*})$  is a limit point of the sequence  $\{z^k\}_{k \geq 0}$ . Without loss of generality, we assume that  $z^k = (u^k, v^k) \rightarrow (z^{1*}, z^{2*})$ . Our goal is to show that for any  $\zeta = (\zeta^1, \zeta^2) \in \mathcal{U} \times \mathcal{V}$ , we have

$$\nabla \mathcal{L}_{\bar{x}}(z^*)^\top (\zeta - z^*) \leq 0.$$

According to Lemma 6.1, the above inequality is equivalent to

$$\nabla_1 \mathcal{L}_{\bar{x}}(z^*)^\top (\zeta^1 - z^{1*}) \leq 0, \quad \forall \zeta^1 \in \mathcal{U}, \quad (3.16)$$

$$\nabla_2 \mathcal{L}_{\bar{x}}(z^*)^\top (\zeta^2 - z^{2*}) \leq 0, \quad \forall \zeta^2 \in \mathcal{V}, \quad (3.17)$$

where  $\nabla \mathcal{L}_{\bar{x}}(z^*) = (\nabla_1 \mathcal{L}_{\bar{x}}(z^*)^\top, \nabla_2 \mathcal{L}_{\bar{x}}(z^*)^\top)^\top$  is the gradient of  $\mathcal{L}_{\bar{x}}$  at  $z^*$ . By contradiction, suppose that there exists a vector  $\tilde{\zeta}^2 \in \mathcal{V}$ , such that

$$\nabla_2 \mathcal{L}_{\bar{x}}(z^*)^\top (\tilde{\zeta}^2 - z^{2*}) > 0. \quad (3.18)$$

Set  $r^k := \tilde{\zeta}^2 - v^k$ . As the sequence  $\{v^k\}_{k \geq 0}$  converges to  $z^{2*}$ , the sequence  $\{r^k\}_{k \geq 0}$  converges to  $\tilde{\zeta}^2 - z^{2*}$ . Thus, due to the continuity of the gradient, there exists  $N > 0$  such that for all  $k > N$  we have

$$\nabla_2 \mathcal{L}_{\bar{x}}(z^k)^\top r^k > 0.$$

So,  $d^k := (\mathbf{0}^\top, (r^k)^\top)^\top$  is an ascent direction of  $\mathcal{L}_{\bar{x}}$  at  $z^k$ . By backtracking line search [10, Lemma 4.3], for given parameter  $\alpha \in (0, 1)$ , there exists a step size  $t_k \in (0, 1)$  such that

$$\mathcal{L}_{\bar{x}}(z^k + t_k d^k) - \mathcal{L}_{\bar{x}}(z^k) \geq \alpha t_k \nabla \mathcal{L}_{\bar{x}}(z^k)^\top d^k, \quad \forall k > N.$$

Therefore

$$\mathcal{L}_{\bar{x}}(u^k, v^k + t_k r^k) - \mathcal{L}_{\bar{x}}(u^k, v^k) \geq \alpha t_k \nabla_2 \mathcal{L}_{\bar{x}}(z^k)^\top r^k > 0, \quad \forall k > N. \quad (3.19)$$

Since  $\mathcal{V}$  is convex, we have

$$v^k + t_k r^k = (1 - t_k)v^k + t_k \tilde{\zeta}^2 \in \mathcal{V}, \quad \forall k > N.$$

Hence,

$$\mathcal{L}_{\bar{x}}(u^{k+1}, v^{k+1}) \geq \mathcal{L}_{\bar{x}}(u^k, v^{k+1}) \geq \mathcal{L}_{\bar{x}}(u^k, v^k + t_k r^k) > \mathcal{L}_{\bar{x}}(u^k, v^k), \quad \forall k > N.$$

So, the sequence of function values  $\{\mathcal{L}_{\bar{x}}(u^k, v^k)\}$  is non-decreasing and also bounded above. Therefore, it is convergent. The last inequality and the convergence of  $\{\mathcal{L}_{\bar{x}}(u^k, v^k)\}$  implies

$$\lim_{k \rightarrow \infty} \mathcal{L}_{\bar{x}}(u^k, v^k + t_k r^k) - \mathcal{L}_{\bar{x}}(u^k, v^k) = 0.$$

The above equation and (4.17) gives

$$\nabla_2 \mathcal{L}_{\bar{x}}(z^*)^\top (\tilde{\zeta}^2 - z^{2*}) = 0,$$

which contradicts (3.18). This prove (3.17). The inequality (3.16) can be proved similarly.  $\square$

Theorem 6 provides conditions under which the limit points of the sequence obtained by the alternating method are helpful in solving problem (3.13). These conditions can be checked for (1.1). More specifically,  $\mathcal{L}$  is continuously differentiable if  $\hat{f}_j$  and  $g_j^*$  are so. Furthermore, it is bounded above if static robust counterpart (3.10) has an optimal solution. Finally,  $\mathcal{V}$  is a closed set when  $dom(g_j^*)$  is closed for all  $j$ .

**Remark 3.4.1.** In Theorem 6, it is established that all limit points of the sequence generated by the alternating method are stationary. Generally, stationarity is necessary for local optimality [10]. However, stationary points are not necessarily optimal solutions. Such a property requires (generalized) concavity assumption to hold. Under generalized concavity assumptions on problem (3.13), the alternating method globally converges; For more details, see [64, Proposition 6] and [124].  $\square$

Using this theorem, we can find a lower bound for (ARC) in the following way: starting from initial solution  $x^{(0)}$  and initial scenario  $u^{(0)}$ , we can find the limit points of  $\{z^k\}$ , denoted

by  $\bar{z}^{(0)}$ . Let us denote by  $\bar{\mathcal{W}}$  the set of limit points after each iteration. Limiting ourselves to  $\bar{\mathcal{W}}$ , we can find a here-and-now solution  $x^{(\ell)}$  using the following optimization problem

$$\begin{aligned} \inf_{x, \tau} \quad & \tau \\ \text{s.t.} \quad & \mathcal{L}(x, \bar{u}^{(i)}, \bar{v}^{(i)}) - \tau \leq 0, \quad 1 \leq i \leq |\bar{\mathcal{W}}|, \\ & x \in \mathcal{X}. \end{aligned} \tag{P-1}$$

By fixing this decision, we can find new limit point  $\bar{z}^{(\ell)}$  to be added to  $\bar{\mathcal{W}}$ . Algorithm 1 provides the pseudo-code of this procedure.

#### **Algorithm 1**

**Input:**  $\epsilon > 0$ , initial value  $x^{(0)} \in \mathcal{X}$ ,  $u^{(0)} \in \mathcal{U}$ .

**Initialization:** Set iteration counter  $\ell \leftarrow 0$ , and set  $\bar{\mathcal{W}} = \emptyset$ .

**Repeat:** Execute the following steps:

(**Step 1**) obtain  $(\bar{u}^{(\ell)}, \bar{v}^{(\ell)})$  as a stationary point of  $\mathcal{L}_{x^{(\ell)}}(u, v)$  by applying Alternating Method. Set  $\bar{\mathcal{W}} = \bar{\mathcal{W}} \cup \{(\bar{u}^{(\ell)}, \bar{v}^{(\ell)})\}$ .

(**Step 2**) Find  $(x^*, \tau^*)$  by solving (P-1).

Update iteration counter  $\ell \leftarrow \ell + 1$ , and set

$$x^{(\ell)} \leftarrow x^*,$$

$$\tau^{(\ell)} \leftarrow \tau^*,$$

**Until:**  $\|\tau^{(\ell)} - \tau^{(\ell-1)}\| \leq \epsilon$ .

**Return:**  $x^{(\ell)}, \tau^{(\ell)}$ .

It can be seen that the optimal value of (P-1) in Algorithm 1 is a lower bound for problem (3.12). Since, we add one more constraint to (P-1) in each iteration, the sequence of the lower bounds is non-decreasing.

Another way to generate lower bounds is to use finite-scenario approach. In the next section, we show how we can improve the lower bounds obtained by finite-scenario approach using the results we have.

### 3.4.2 Dual-Based Cutting Plane Algorithm

Based on the original form of (ARC), one can find a lower bound by using finite-scenario approach and only considering a finite subset  $\{u^1, \dots, u^\ell\}$  of  $\mathcal{U}$ . This idea leads to the following convex programming problem:

$$\begin{aligned}
 & \inf_{\substack{x \in \mathcal{X}, \tau \\ \{y^k\}_k}} \tau \\
 \text{s.t.} \quad & f_j(x, y^k, u^k) \leq 0, \quad j \in [m], k \in [\ell], \\
 & f_0(x, y^k, u^k) \leq \tau, \quad k \in [\ell],
 \end{aligned} \tag{3.20}$$

which is called finite-scenario approach of the (ARC) problem. A technique to obtain a finite set of scenarios is by (i) approximating (ARC) with a suitable decision rule and (ii) finding the active scenarios in the uncertainty set [66]. Since we are considering a nonlinear problem, we use constant decision rule. So, we first find an optimal solution  $(x^*, y^*)$  of (RC) with the optimal value  $t^*$ . After that, by fixing the obtained  $(x^*, y^*)$ , we take an active (binding) scenario on each constraint

$$\begin{aligned}
 & f_j(x^*, y^*, u) \leq 0, \quad j \in [m], \\
 & f_0(x^*, y^*, u) \leq t^*.
 \end{aligned} \tag{3.21}$$

The optimal value of the finite-scenario approach problem is a lower bound for the optimal objective value of the original (ARC) model since feasibility is fulfilled for only a subset of the uncertainty set.

In Theorem 7 we show how to construct a better lower bound by means of dual cuts.

**Theorem 7.** *Let  $\{u^1, \dots, u^\ell\} \subseteq \mathcal{U}$  and  $\{v^1, \dots, v^\ell\} \subseteq \mathcal{V}$ . Then optimal value of*

$$\begin{aligned}
 & \inf_{\substack{x \in \mathcal{X}, \tau \\ \{y^k\}_k}} \tau \\
 \text{s.t.} \quad & f_j(x, y^k, u^k) \leq 0, \quad j \in [m], k \in [\ell], \\
 & f_0(x, y^k, u^k) \leq \tau, \quad k \in [\ell], \\
 & \mathcal{L}(x, u^k, v^k) \leq \tau, \quad k \in [\ell],
 \end{aligned} \tag{3.22}$$

provides a lower bound for (ARC).

*Proof.* Let us denote the optimal value of (ARC) by  $Opt$ . Also, we have the following inequality due to the weak duality

$$\forall x \in \mathcal{X} \quad \sup_{u \in \mathcal{U}} \mathcal{T}(x, u) \geq \sup_{\substack{u \in \mathcal{U} \\ v \in \mathcal{V}}} \mathcal{L}(x, u, v),$$

where  $\mathcal{T}(x, u) := \inf_{y \in \mathbb{R}^{n_y}} \{f_0(x, y, u) : f_j(x, y, u) \leq 0, j \in [m]\}$ . Therefore,

$$\begin{aligned} Opt &= \inf_{\substack{x \in \mathcal{X} \\ \tau \in \mathbb{R}}} \tau \\ \text{s.t.} \quad & \sup_{u \in \mathcal{U}} \mathcal{T}(x, u) \leq \tau, \\ & \sup_{\substack{u \in \mathcal{U} \\ v \in \mathcal{V}}} \mathcal{L}(x, u, v) \leq \tau, \end{aligned}$$

where the second constraint is redundant. Let  $\bar{\mathcal{U}} \subseteq \mathcal{U}$  and  $\bar{\mathcal{V}} \subseteq \mathcal{V}$ . Then

$$\begin{aligned} Opt &\geq \inf_{x \in \mathcal{X}, \tau} \tau \\ \text{s.t.} \quad & \sup_{u \in \bar{\mathcal{U}}} \mathcal{T}(x, u) \leq \tau, \\ & \sup_{\substack{u \in \bar{\mathcal{U}} \\ v \in \bar{\mathcal{V}}}} \mathcal{L}(x, u, v) \leq \tau. \end{aligned}$$

So, if  $\bar{\mathcal{U}} = \{u^1, \dots, u^\ell\}$  and  $\bar{\mathcal{V}} = \{v^1, \dots, v^\ell\}$ , then we have

$$\begin{aligned} Opt &\geq \inf_{x \in \mathcal{X}, \tau} \tau \\ \text{s.t.} \quad & \inf_{y \in \mathbb{R}^{n_y}} \{f_0(x, y, u^k) : f_j(x, y, u^k) \leq 0, j \in [m]\} \leq \tau, \quad k \in [\ell], \\ & \mathcal{L}(x, u^k, v^k) \leq \tau, \quad k \in [\ell], \end{aligned} \tag{3.23}$$

which is equivalent to (3.22). □

Using Theorem 7, we develop Algorithm 2, which generates potentially better lower bounds compared to Algorithm 1 and finite-scenario approach.



**Algorithm 2**

(Step 1) Find  $(x^*, y^*)$  as a static solution.

(Step 2) Given  $(x^*, y^*)$ , find the active scenarios  $\{u^0, u^1, \dots, u^m\}$ .

(Step 3) Use Alternating Method, to find  $(\bar{u}^{(k)}, \bar{v}^{(k)})$  as a stationary point of  $\mathcal{L}_{x^*}(u, v)$  starting from  $u^k$  ( $k \in [m_0]$ ).

(Step 4) Given  $\{(\bar{u}^{(k)}, \bar{v}^{(k)})\}_k$ , solve

$$\begin{aligned} & \inf_{\substack{x \in \mathcal{X}, \tau \\ \{y^k\}_k}} \tau \\ & \text{s.t.} \quad f_0(x, y^k, \bar{u}^{(k)}) \leq \tau, \quad k \in [m_0], \\ & \quad \quad f_j(x, \bar{u}^{(k)}, y^k) \leq 0, \quad j \in [m], \quad k \in [m_0], \\ & \quad \quad \mathcal{L}(x, \bar{v}^{(k)}, \bar{u}^{(k)}) \leq \tau, \quad k \in [m_0]. \end{aligned}$$

We emphasize that the algorithms to construct a sequence of lower bounds are applicable for any nonlinear ARO problem.

## 3.5 Numerical Experiments

In this section, we illustrate the performance of the discussed algorithms. All the numerical results were carried out on a laptop featuring Intel(R) Core(TM) i5-3210M CPU, 2.50 GHz processor, and 8 GB of RAM. We implemented the algorithms in MATLAB (2022a) and used YALMIP toolbox [84] to pass the optimization problems to MOSEK as a solver [91]. All results of this section are presented with four decimals.

### 3.5.1 Problem Setting

We consider the following uncertain problem:

$$\inf_{x \in \mathcal{X}} \inf_{y \in \mathcal{Y}(x, u)} \hat{f}_0(x, u) + \|A_0 y - b^0\|_2 - (p^0)^\top y + q_0, \quad (3.24)$$

where

$$\mathcal{Y}(x, u) = \left\{ y : \hat{f}_j(x, u) + \|A_j y - b^j\|_2 - (p^j)^\top y + q_j \leq 0, \quad j \in [m] \right\} \subseteq \mathbb{R}^{n_y},$$

$A_j \in \mathbb{R}^{r \times n_y}$ ,  $b^j \in \mathbb{R}^r$ , and  $p^j \in \mathbb{R}^{n_y}$ . For  $j \in [m]$ , let us set

$$g_j(y) := \|A_j y - b^j\|_2 - (p^j)^\top y + q_j.$$

Thus, the perspective functions corresponding to the conjugate of  $g_j(y)$  is given by

$$\begin{aligned} \lambda_j > 0: \quad & \lambda_j g_j^* \left( \frac{w^j}{\lambda_j} \right) = \inf_{z^j} \left\{ \lambda_j \left( (b^j)^\top z^j - q_j \right) \mid \|z^j\|_2 \leq 1, (A_j)^\top z^j - p^j = \frac{w^j}{\lambda_j} \right\}, \\ \lambda_j = 0: \quad & \lambda_j g_j^* \left( \frac{w^j}{\lambda_j} \right) = \delta_{\text{dom}(g_j)}^*(w^j) = \delta_{\mathbb{R}^{n_y}}^*(w^j) = \delta_{\{0\}}(w^j). \end{aligned}$$

Now we consider (ARC) version of the uncertain problem (3.24). According to Theorem 2, after dualizing over the wait-and-see variable  $y$ , with some algebra (see Section 6.4 of [101]), we get the following equivalent dual reformulation:

$$\begin{aligned} \inf_{x \in \mathcal{X}} \quad & \sup_{u \in \mathcal{U}, \lambda_j, z^j} \sum_{j=0}^m \lambda_j \hat{f}_j(x, u) + \sum_{j=0}^m \left( \lambda_j q_j - \lambda_j (b^j)^\top z^j \right) \\ \text{s.t.} \quad & \sum_{j=0}^m \left( \lambda_j (A_j)^\top z^j - \lambda_j p^j \right) = \mathbf{0}, \\ & \|z^j\|_2 \leq 1, \quad \lambda_0 = 1, \quad \lambda_j \geq 0, \quad j \in [m_0]. \end{aligned} \tag{3.25}$$

Let us consider the parameters in a matrix form, i.e.,

$$\begin{aligned} A^\top &:= [(A_0)^\top \dots (A_m)^\top] \in \mathbb{R}^{n_y \times r(m+1)}, \\ P &:= [p^0 \dots p^m] \in \mathbb{R}^{n_y \times (m+1)}, \\ b^\top &:= [(b^0)^\top \dots (b^m)^\top] \in \mathbb{R}^{r(m+1)}, \\ \bar{z}^j &:= \lambda_j z^j, \quad \bar{z}^\top := [(\bar{z}^0)^\top \dots (\bar{z}^m)^\top] \in \mathbb{R}^{r(m+1)}. \end{aligned}$$

In addition, by setting

$$\begin{aligned} v^\top &:= (\lambda^\top, \bar{z}^\top), \\ \mathcal{V} &:= \left\{ v \mid A^\top \bar{z} - P \begin{pmatrix} 1 \\ \lambda \end{pmatrix} = \mathbf{0}, \lambda \geq \mathbf{0}, \|\bar{z}^0\|_2 \leq 1, \|[\bar{z}]^j\|_2 \leq \lambda_j, j \in [m] \right\}, \\ \mathcal{L}(x, u, v) &:= (1, \lambda^\top) \hat{f}(x, u) + (1, \lambda^\top) q - \bar{z}^\top b, \end{aligned}$$

where  $\hat{f}(x, u)$  is a vector-valued function with components  $\hat{f}_j(x, u)$ , and

$$[\bar{z}]^j = [\bar{z}]_{jr+1, \dots, (j+1)r},$$

we can write (3.25) as

$$\inf_{x \in \mathcal{X}} \sup_{u \in \mathcal{U}, v \in \mathcal{V}} \mathcal{L}(x, u, v).$$

We use the optimality gap to compare the quality of the lower bounds obtained by applying the finite-scenario approach (3.20), Algorithm 1, and Algorithm 2:

$$\text{OptGap} = \left( \frac{UB - LB}{|LB| + 10^{-4}} \right) \times 100,$$

where  $LB$  is the obtained lower bound, and  $UB$  is the best obtained upper bound for a given instance. Note that the constant  $10^{-4}$  is added to the denominator to avoid division over zero.

We consider two classes of randomly generated instances, each containing 100 instances.

Class One: In this class, we consider small-sized instances. We consider  $n_y = 2$ ,  $n_x = 2$ ,  $m = 2$ , and  $r = 5$ . Furthermore,

$$\mathcal{U} = \mathcal{U}_1 := \{u \in \mathbb{R}^2 : \|u\|_2 \leq 1\},$$

$$\mathcal{X} = \mathcal{X}_1 := \{x \in \mathbb{R}^2 : x_1 + 2x_2 \leq 3, 2x_1 + x_2 \leq 3, x_1, x_2 \geq 0\},$$

and

$$\hat{f}_j(x, u) := c^{j^\top} x + \alpha^{j^\top} u, \quad j \in [m_0].$$

In this class, we obtained upper bounds by solving the static robust counterpart problem, which is equivalent to the perspectification approach. Additionally, we employed the K-adaptability approach to obtain upper bounds. The K-adaptability approach involves splitting the uncertainty set  $\mathcal{U}$  into  $K$  partitions ( $\mathcal{U} = \cup_{k=1}^K \mathcal{U}_k$ ) and solves the following

problem

$$\begin{aligned} & \inf_{\substack{x \in \mathcal{X}, \tau \\ \{y^k\}_k}} \tau \\ \text{s.t.} \quad & \hat{f}_j(x, u) + \|A_j y^k - b^j\|_2 - (p^j)^\top y^k + q_j \leq 0, \quad j \in [m], k \in [K], \forall u \in \mathcal{U}_k, \\ & \hat{f}_0(x, u) + \|A_0 y^k - b^0\|_2 - (p^0)^\top y^k + q_0 \leq \tau, \quad k \in [K], \forall u \in \mathcal{U}_k. \end{aligned}$$

We set  $K = 8$  and partitioned the uncertainty set into eight regions, each being an octant.

Class Two: This class contains large-sized instances. We consider  $n_y = 100$ ,  $n_x = 100$ ,  $m = 5$ , and  $r = 120$ . Furthermore,

$$\begin{aligned} \mathcal{U} = \mathcal{U}_2 & := \{u \in \mathbb{R}^{20} : \|u\|_2 \leq 1\}, \\ \mathcal{X} = \mathcal{X}_2 & := \{x \in \mathbb{R}^{100} : \|x\|_2 \leq 1, e^\top x \leq 1, d^\top x \geq 0\}, \end{aligned}$$

where  $e \in \mathbb{R}^{n_x}$  is the vector of all ones,  $d \in \mathbb{R}^{n_x}$  is a random vector, and

$$\hat{f}_j(x, u) := c^{j^\top} x + \alpha^{j^\top} u, \quad j \in [m_0].$$

We use static approximation to obtain an upper bound on the optimal value of the instances in this class.

To generate random instances, for each  $j \in [m_0]$ , we randomly generate  $A_j$ ,  $p^j$ ,  $b^j$ ,  $\alpha^j$ ,  $c^j$ , and  $d$  by drawing their (entries) values from a standard normal distribution using a built-in MATLAB function “randn”.

### 3.5.2 Numerical results

In this section, we present the results of the numerical experiments.

Class One: We present the statistic on the optimality gaps of the finite-scenario approach (3.20), Algorithm 1, and Algorithm 2 in Table 3.1 (details can be found in Table 3.3). Since the upper bound obtained using the K-adaptability approach for the instances of this class is lower than the one from the perspective approach, which is equivalent to the static approximation, we report the gap using the former. As one can see, Algorithm 2 outperforms

the other methods on average.

**Table 3.1:** Statistic of optimality gaps of instances in Class One.

Method	Mean	Standard deviation
Algorithm 1	26.4925	75.3803
Algorithm 2	<b>22.2883</b>	<b>72.4447</b>
Finite-scenario approach	49.7797	336.4677

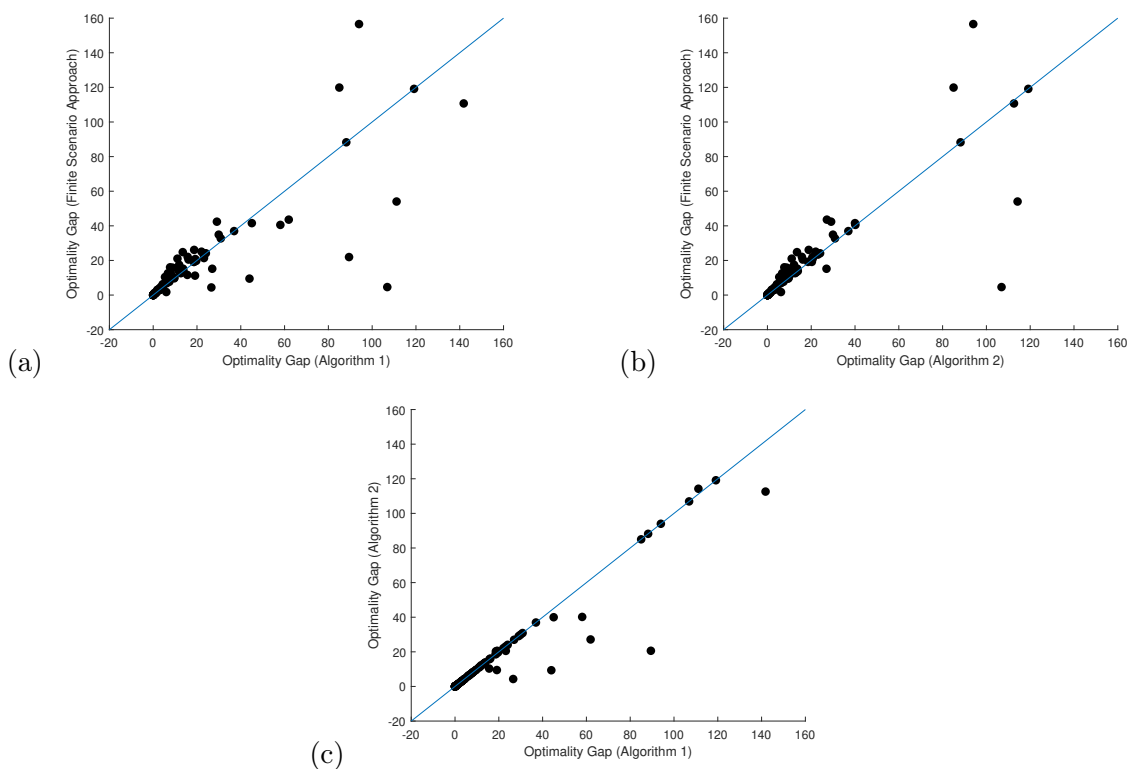
Figure 3.1 compares the optimality gaps of the solutions obtained by Algorithm 1, Algorithm 2, and the finite-scenario approach, where each point corresponds to an instance. As shown in Figures 3.1a and 3.1b, Algorithms 1 and 2 outperform the finite-scenario approach. More specifically, in 84 instances, Algorithm 2 generates better lower bounds, while the finite-scenario approach generates better lower bounds in only 15 instances. We should emphasize that if the scenarios considered in both approaches are the same, Theorem 7 shows that the lower bound obtained by Algorithm 2 should outperform the finite-scenario approach. However, these two methods do not generate the same scenarios. Additionally, Figure 3.1c shows that the solutions obtained by Algorithm 2 have a similar or better optimality gap to the ones obtained from Algorithm 1.

Next to the quality of the approaches, we also report their solution times.

The average solution times for Algorithm 1, Algorithm 2, and finite-scenario approach are 0.0500, 0.0483, and 0.2253 seconds, respectively. Figure 3.2 depicts a scatter plot comparing the solution times in each instance. Illustrated in Figures 3.2a and 3.2b, Algorithms 1 and 2 reach lower bounds more rapidly compared to the finite-scenario approach. In Figure 3.2c, each point represents an instance, with the horizontal and vertical axes indicating the solution times of Algorithm 1 and Algorithm 2, respectively. For most of the instances in this class, we observe that Algorithm 2 reached a solution faster than Algorithm 1.

Hitherto, we have seen that Algorithm 2 performs well in the instances in Class One. In what follows, we analyze the performance of the algorithms in the instances of Class Two.

Class Two: We present the statistic on the optimality gaps of the solutions obtained by different algorithms in Table 3.2 (details can be found in Table 3.4). As one can see,



**Figure 3.1:** The comparison of the optimality gaps of the solutions obtained by Algorithm 1, Algorithm 2, and finite-scenario approach for the instances in Class One.

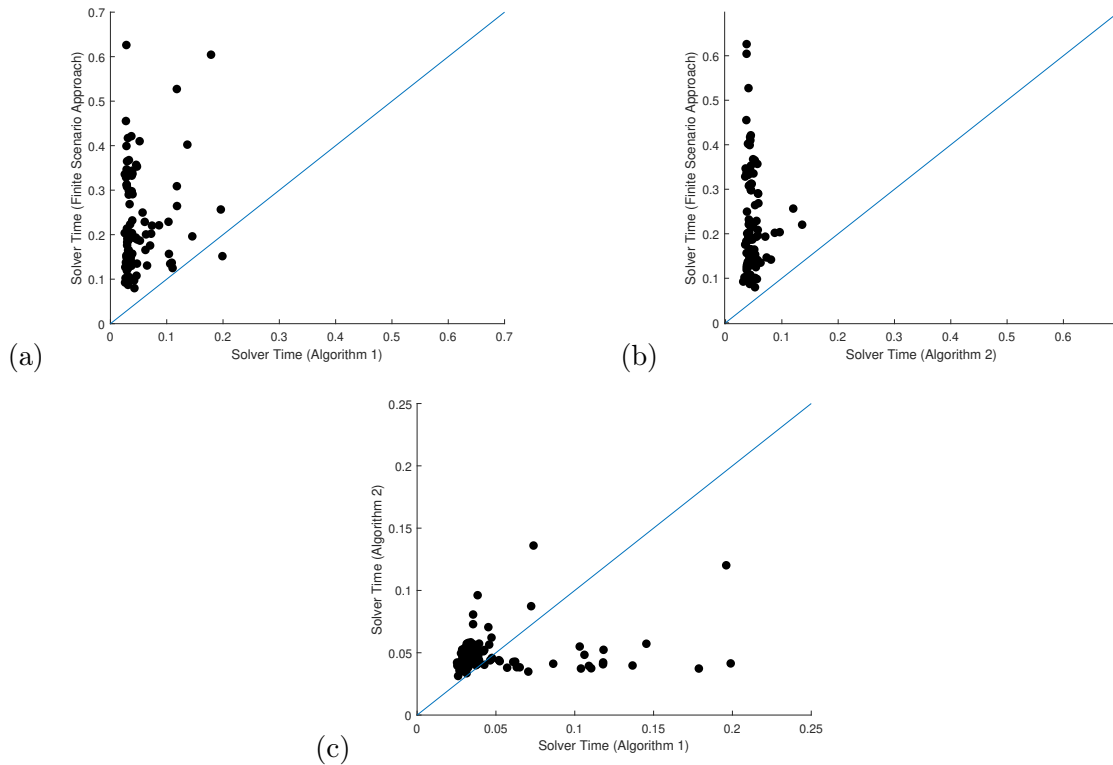
Algorithm 1 outperforms the other methods on average.

**Table 3.2:** Statistic of optimality gaps of instances in Class Two.

Method	Mean	Standard deviation
Algorithm 1	<b>56.5417</b>	<b>83.8414</b>
Algorithm 2	57.7425	88.1209
Finite-scenario approach	64.5980	114.8234

To have a clearer comparison, we illustrate the optimality gaps in Figure 3.3. Remarkably, both algorithms exhibit better performance in nearly all instances compared to the finite-scenario approach (Figures 3.3a and 3.3b). Furthermore, as one can see in Figure 3.3c, the optimality gap of the solutions obtained by Algorithms 1 and 2 are close, and in all instances (except two of them), Algorithm 1 provides a solution with a slightly lower optimality gap compared to Algorithm 2.

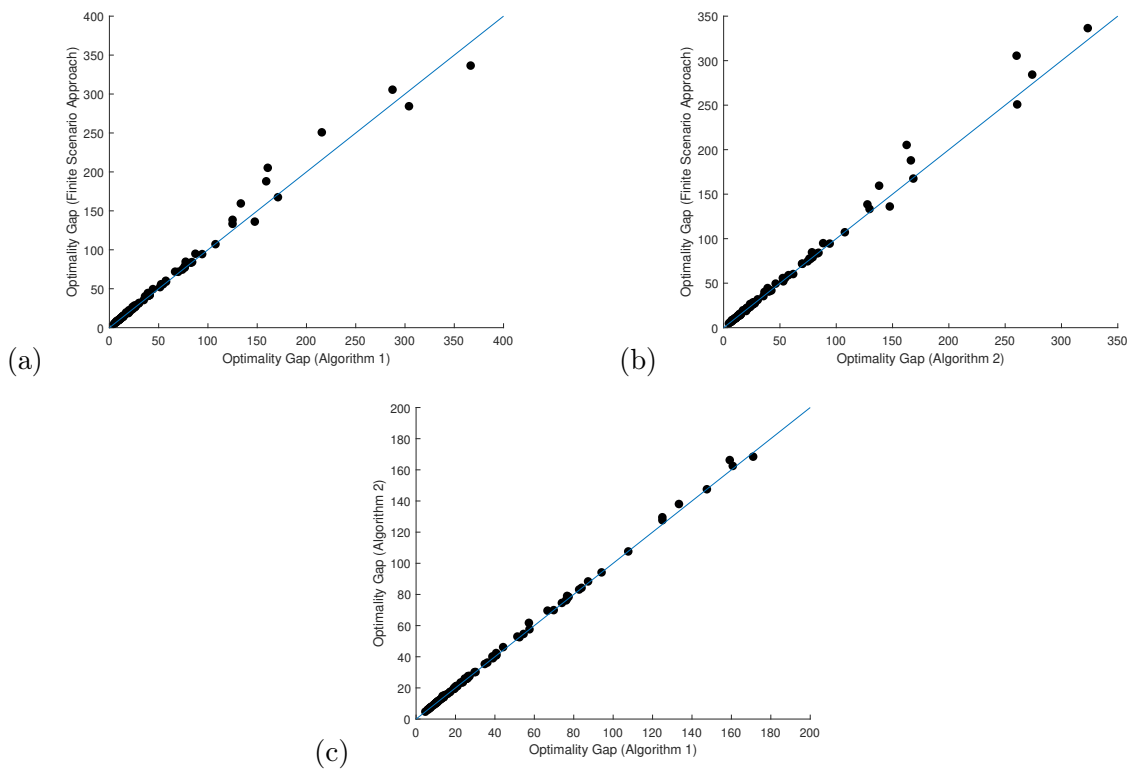
The average solution times for Algorithm 1, Algorithm 2, and finite-scenario approach are



**Figure 3.2:** The comparison of the solution time of Algorithm 1, Algorithm 2, and finite-scenario approach for the instances in Class One.

7.5085, 3.6327, and 0.5417 seconds, respectively. Figure 3.4 presents the scatter plot of the solution times of these approaches on each instance. In this class, the computation times of Algorithms 1 and 2 are higher than the finite-scenario approach because they solve more (sub-)optimization problems than the finite-scenario approach to reach a lower bound (as shown in Figures 3.4a and 3.4b). From Figure 3.4c, across a significant proportion of instances within this classification, it is evident that Algorithm 2 exhibited notable performance in achieving solutions faster than Algorithm 1.

Overall for Class Two, our findings demonstrate the competitive performance of both Algorithm 1 and Algorithm 2 on average.

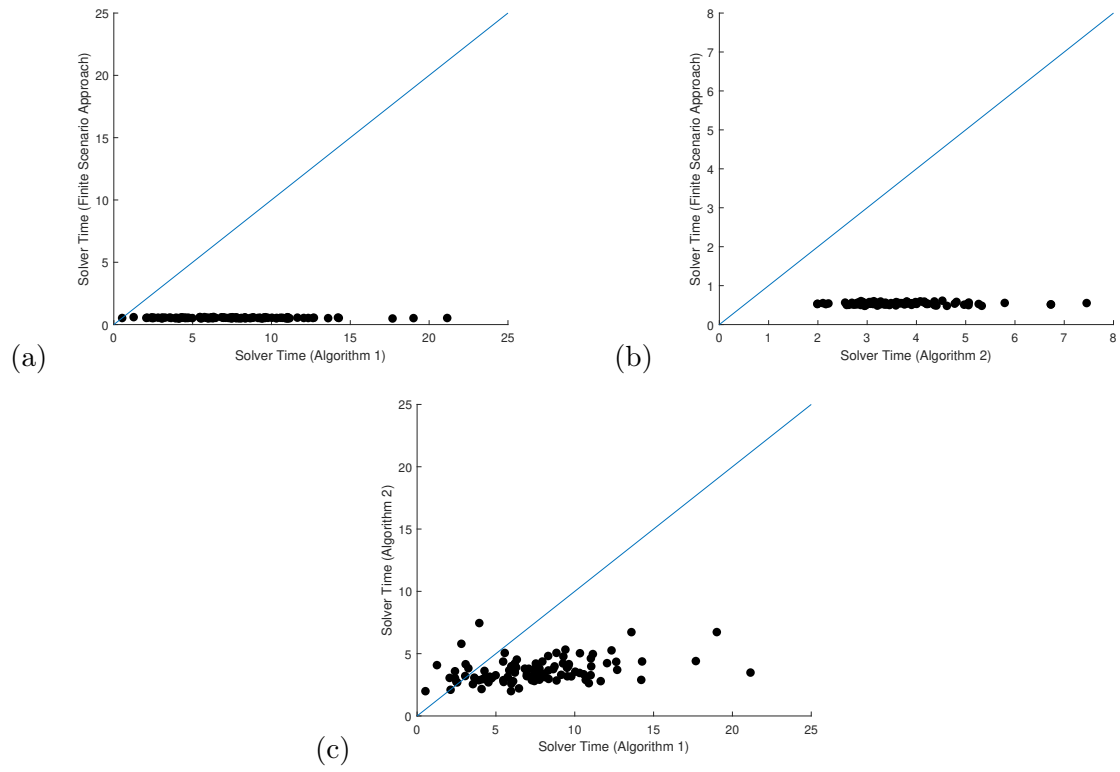


**Figure 3.3:** The comparison of the optimality gaps of the solutions obtained by Algorithm 1, Algorithm 2, and finite-scenario approach for the instances in Class Two.

### 3.6 Conclusions

This chapter studied a general nonlinear ARO model with objective and constraint uncertainty. We obtained an equivalent dual formulation by applying Fenchel’s duality on the wait-and-see variable, a nonlinear static robust optimization. We investigated when the dual formulation is convex in the decision variables. Also, we explored reaching upper and lower bounds for the original problem based on the dual formulation. Thanks to the equivalent dual reformulation, we presented and analyzed two algorithms. These algorithms aimed to find a lower bound on the optimal objective value of the general nonlinear ARO model. We demonstrated by numerical results that our algorithm could produce a locally robust solution with an acceptable optimality gap.





**Figure 3.4:** The comparison of solution times between Algorithm 1, Algorithm 2, and the finite-scenario approach for instances in Class Two.

**Table 3.3:** Detailed numerical results from Class One.

Case	$LB_1$	$LB_2$	$LB_3$	$UB_8$	$UB_1$	Case	$LB_1$	$LB_2$	$LB_3$	$UB_8$	$UB_1$
1	<b>259.1977</b>	<b>259.1977</b>	259.1946	260.0564	261.3407	51	<b>3.3127</b>	<b>3.3127</b>	3.3122	3.3225	3.3274
2	<b>9.3817</b>	<b>9.3817</b>	9.3658	9.6877	10.0478	52	0.4577	0.5206	<b>0.5252</b>	1.1068	3.6220
3	<b>6.9970</b>	6.9935	6.7271	7.5588	7.9666	53	<b>2.9971</b>	<b>2.9971</b>	<b>2.9971</b>	2.9971	3.0049
4	<b>2.7486</b>	<b>2.7486</b>	2.7103	2.9390	3.0800	54	<b>0.8657</b>	<b>0.8657</b>	0.8653	1.6291	1.9446
5	11.0089	<b>11.0090</b>	10.8160	11.4931	11.8715	55	<b>8.7115</b>	<b>8.7115</b>	8.7114	8.7119	8.7119
6	<b>15.3520</b>	<b>15.3520</b>	15.2586	16.3023	17.8020	56	0.8995	0.8865	<b>1.2330</b>	1.8994	3.1152
7	<b>-1.9373</b>	<b>-1.9373</b>	<b>-1.9373</b>	-1.9357	-1.9357	57	<b>3.3758</b>	3.3751	3.3751	3.3934	3.4227
8	<b>9.3109</b>	<b>9.3109</b>	9.3063	10.2061	11.1458	58	<b>2.8508</b>	<b>2.8508</b>	<b>2.8508</b>	2.8508	3.3457
9	10.6770	10.6762	<b>11.7674</b>	13.5560	15.4325	59	<b>3.4591</b>	3.4590	3.4419	3.5102	3.5223
10	<b>5.4643</b>	<b>5.4643</b>	5.3333	6.6699	8.1682	60	<b>4.7327</b>	<b>4.7327</b>	4.7325	10.3711	19.4164
11	<b>5.4997</b>	5.4970	5.4996	5.5553	5.7042	61	<b>2.1088</b>	<b>2.1088</b>	1.9812	2.2950	2.4424
12	<b>4.8946</b>	<b>4.8946</b>	4.4919	5.4370	5.6290	62	<b>2.4899</b>	<b>2.4899</b>	2.4305	2.6931	2.8786
13	-1.7771	<b>-1.2453</b>	-1.2531	-0.7450	-0.2057	63	<b>3.5704</b>	<b>3.5704</b>	3.5606	4.3928	5.2596
14	<b>15.0970</b>	<b>15.0970</b>	14.8494	15.7447	16.1069	64	<b>2.4027</b>	<b>2.4027</b>	<b>2.4027</b>	2.4027	2.4215
15	<b>3.1803</b>	<b>3.1803</b>	3.1792	3.1816	3.1816	65	<b>4.0104</b>	<b>4.0104</b>	3.9697	4.0845	4.1758
16	<b>32.7964</b>	<b>32.7964</b>	<b>32.7964</b>	32.7979	32.8014	66	<b>3.4362</b>	<b>3.4362</b>	3.3095	4.4646	5.5969
17	<b>0.1421</b>	<b>0.1421</b>	0.0323	1.1248	2.3101	67	<b>18.8952</b>	<b>18.8952</b>	17.9105	20.1629	21.3240
18	<b>16.5824</b>	<b>16.5824</b>	15.0801	18.8211	19.4832	68	<b>2.8023</b>	<b>2.8023</b>	<b>2.8023</b>	2.8025	2.9026
19	<b>3.6685</b>	<b>3.6685</b>	3.3255	4.7357	5.8387	69	<b>1.6128</b>	1.5944	1.5914	1.9228	1.9755
20	7.0912	<b>8.6052</b>	8.5969	8.9749	9.3419	70	<b>3.9882</b>	<b>3.9882</b>	3.9630	4.0116	4.0224
21	<b>49.5685</b>	<b>49.5685</b>	49.5664	50.9667	52.7088	71	<b>31.4786</b>	<b>31.4786</b>	31.3828	32.6877	33.7343
22	<b>7.3621</b>	<b>7.3621</b>	7.3552	8.8104	11.2659	72	<b>756.1827</b>	<b>756.1827</b>	756.1826	756.2022	756.2175
23	21.5784	<b>28.4078</b>	28.3726	31.0675	38.7033	73	<b>6.4255</b>	<b>6.4255</b>	6.2120	6.9449	7.2463
24	<b>12.5946</b>	<b>12.5946</b>	12.5854	15.6238	20.4960	74	<b>2.4011</b>	2.4010	2.4000	2.4714	2.6531
25	<b>14.1311</b>	<b>14.1311</b>	14.1301	14.1403	14.1403	75	<b>2.0766</b>	<b>2.0766</b>	2.0765	2.3492	2.4106
26	<b>52.2638</b>	<b>52.2638</b>	52.2504	53.0016	53.6820	76	3.0271	3.0215	<b>3.1531</b>	3.2083	3.5310
27	<b>-0.4110</b>	<b>-0.4110</b>	-0.4118	-0.4073	-0.3732	77	<b>2.2849</b>	<b>2.2849</b>	<b>2.2849</b>	2.3563	2.4955
28	<b>393.0423</b>	<b>393.0423</b>	377.7744	436.0650	452.3473	78	<b>1.3092</b>	<b>1.3092</b>	<b>1.3092</b>	1.3112	1.3126
29	<b>7.9651</b>	<b>7.9651</b>	7.6756	9.2498	10.8485	79	<b>7.8062</b>	<b>7.8062</b>	5.9023	15.1441	23.1083
30	<b>30.2205</b>	<b>30.2205</b>	28.6526	34.9758	38.2203	80	3.3318	<b>3.4906</b>	3.4497	3.8514	4.1593
31	<b>7.2094</b>	<b>7.2094</b>	7.1337	7.3583	7.3583	81	2.6656	<b>2.8999</b>	2.8545	3.1747	3.6565
32	<b>6.6596</b>	6.6595	6.6530	6.7380	6.9949	82	<b>7.9201</b>	<b>7.9201</b>	7.9107	7.9570	8.3552
33	<b>95.2210</b>	<b>95.2210</b>	94.9450	102.2506	110.4956	83	<b>4.1187</b>	4.1080	3.9211	4.6071	4.9426
34	<b>21.5530</b>	<b>21.5530</b>	21.5225	21.7216	21.9525	84	<b>3.4734</b>	<b>3.4734</b>	3.2243	3.7412	3.9954
35	<b>4.7314</b>	<b>4.7314</b>	4.6987	4.8901	4.9057	85	3.5410	<b>3.6188</b>	3.5939	4.3623	5.8730
36	18.2915	<b>18.2916</b>	18.2899	18.5365	19.1278	86	<b>11.5812</b>	11.5660	10.9026	13.7476	15.7727
37	5.8375	<b>16.7701</b>	16.7202	17.6550	18.3373	87	<b>10.9663</b>	<b>10.9663</b>	10.9658	10.9887	10.9990
38	5.6850	<b>7.2390</b>	6.4109	9.2046	11.3974	88	<b>14.1618</b>	<b>14.1618</b>	14.1615	14.1639	14.1641
39	<b>86.3898</b>	<b>86.3898</b>	86.3881	86.5066	86.5195	89	<b>5.0708</b>	5.0023	5.0490	6.0166	7.0369
40	<b>-0.3083</b>	<b>-0.3083</b>	-0.3084	-0.1944	0.0310	90	4.8473	4.8461	<b>4.8483</b>	5.5185	5.8588
41	<b>75.7971</b>	<b>75.7971</b>	75.7851	76.0008	76.6293	91	<b>9.3160</b>	<b>9.3160</b>	9.2139	9.6205	10.2391
42	<b>10.9392</b>	<b>10.9392</b>	10.8825	12.9686	18.0775	92	<b>45.8951</b>	<b>45.8951</b>	45.3398	48.8046	50.3318
43	<b>32.2960</b>	<b>32.2960</b>	32.2940	32.3780	32.3964	93	1.9520	1.9519	<b>3.8607</b>	4.0389	4.4740
44	<b>-0.6766</b>	<b>-0.6766</b>	<b>-0.6766</b>	-0.6766	-0.6185	94	<b>0.7561</b>	<b>0.7561</b>	0.6361	1.3990	2.2513
45	<b>1.1652</b>	<b>1.1652</b>	1.1425	1.3013	1.4361	95	-0.7423	<b>-0.6797</b>	-0.6980	-0.4079	0.0277
46	<b>2.5419</b>	<b>2.5419</b>	2.5418	2.8630	3.2250	96	<b>22.3622</b>	<b>22.3622</b>	22.2758	23.3531	23.9807
47	<b>1.0582</b>	<b>1.0582</b>	1.0431	1.3847	2.2834	97	3.6372	<b>5.7130</b>	5.6500	6.8904	7.9793
48	<b>3.6334</b>	<b>3.6334</b>	3.5831	4.1290	4.8998	98	<b>6.6315</b>	<b>6.6315</b>	6.3300	6.9888	7.1655
49	<b>22.1946</b>	<b>22.1946</b>	22.1167	24.3342	27.4000	99	<b>18.9985</b>	<b>18.9985</b>	18.9977	19.5478	20.0648
50	<b>3.3475</b>	<b>3.3475</b>	3.3208	3.4439	3.4843	100	<b>2.7358</b>	<b>2.7358</b>	2.7216	2.8101	2.8288

Note. The column Case contains the instance number, columns  $LB_1$ ,  $LB_2$ , and  $LB_3$  are the lower bounds obtained by Algorithm 1, 2, and the finite-scenario approach, respectively, and the columns  $UB_1$  and  $UB_8$  are the upper bounds obtained by perspective approach and 8-adaptability approach respectively. The accuracy digit is four. In this table, for each instance, the best lower bound is in boldface.

**Table 3.4:** Detailed numerical results from Class Two.

Case	$LB_1$	$LB_2$	$LB_3$	$UB_1$	Case	$LB_1$	$LB_2$	$LB_3$	$UB_1$
1	<b>3.9001</b>	3.8934	3.8854	4.8279	51	<b>-2.9744</b>	-2.9971	-3.0021	-1.9376
2	<b>9.6268</b>	9.6178	9.6015	10.4601	52	<b>3.5619</b>	3.5499	3.5264	4.6196
3	<b>1.6344</b>	1.6123	1.6121	2.8872	53	<b>-1.7460</b>	-1.7904	-1.8189	-0.3012
4	<b>10.8829</b>	10.8773	10.8506	11.6009	54	<b>3.2074</b>	3.1680	3.1807	4.5110
5	<b>12.2019</b>	12.1472	12.1530	13.5152	55	<b>-1.0219</b>	-1.0273	-1.0839	0.0784
6	<b>2.7891</b>	2.7762	2.7543	3.8914	56	<b>-5.9077</b>	-5.9099	-6.0075	-4.5074
7	<b>4.2798</b>	4.2738	4.2123	5.3923	57	<b>9.0173</b>	8.9964	9.0082	10.0026
8	<b>-1.6253</b>	-1.6291	-1.6727	-0.7387	58	<b>-5.3944</b>	-5.4257	-5.4375	-4.3260
9	<b>3.5420</b>	3.5370	3.5318	4.2847	59	<b>5.9016</b>	<b>5.9016</b>	5.8349	6.3315
10	<b>0.2770</b>	0.2365	0.1944	1.6372	60	<b>0.9850</b>	0.9656	0.8858	2.2989
11	<b>-1.3185</b>	-1.3505	-1.3593	-0.3106	61	<b>7.2530</b>	7.2352	7.2083	8.2546
12	<b>-2.0551</b>	-2.1027	-2.1003	-0.5346	62	0.5393	<b>0.5800</b>	0.5150	2.0891
13	<b>8.1494</b>	8.1280	8.1285	8.9722	63	<b>5.2223</b>	5.1960	5.1005	6.5493
14	<b>8.3511</b>	8.3408	8.3322	8.8412	64	<b>-8.0248</b>	-8.0617	-8.0568	-6.4983
15	<b>-1.9698</b>	-2.0357	-1.9960	-0.9572	65	<b>7.0972</b>	7.0572	6.9516	8.5001
16	<b>7.3853</b>	7.3711	7.3958	8.3874	66	<b>-0.4458</b>	-0.5322	-0.5025	1.1889
17	<b>16.0063</b>	15.9723	15.9882	17.1185	67	<b>-0.5969</b>	-0.6999	-0.6607	1.2182
18	<b>2.9578</b>	2.9267	2.9187	4.1038	68	<b>-1.1199</b>	-1.1293	-1.1896	-0.0657
19	<b>2.4222</b>	2.4132	2.4054	3.4077	69	<b>2.2536</b>	2.2530	2.1678	3.1326
20	<b>-1.6671</b>	-1.6730	-1.7801	-0.5029	70	<b>7.2135</b>	7.2085	7.1412	8.6105
21	<b>1.0836</b>	1.0649	1.0506	1.8063	71	<b>9.6860</b>	9.6677	9.6259	10.3959
22	<b>9.8687</b>	9.8084	9.8125	11.3247	72	<b>4.5704</b>	4.5428	4.5342	5.5110
23	<b>0.6145</b>	0.5376	0.5527	1.9393	73	<b>-3.7698</b>	-3.7862	-3.7970	-2.8077
24	<b>-9.4925</b>	-9.5133	-9.5277	-8.5633	74	<b>-5.0022</b>	-5.0044	-5.1230	-3.4930
25	<b>2.1541</b>	2.1525	2.0924	2.9331	75	<b>-1.3936</b>	-1.4183	-1.4138	-0.2234
26	<b>9.5053</b>	9.4669	9.4446	10.8387	76	<b>5.3090</b>	5.2698	5.2504	6.7112
27	<b>-1.2397</b>	-1.2863	-1.3041	0.8808	77	<b>-2.0718</b>	-2.0727	-2.0879	-1.5332
28	<b>10.8428</b>	10.8244	10.7536	11.8421	78	<b>16.0217</b>	16.0155	15.9792	16.7758
29	<b>4.6773</b>	4.6652	4.6285	5.9266	79	<b>8.1416</b>	8.1328	8.1075	8.9623
30	<b>9.9538</b>	9.9512	9.8376	11.3559	80	<b>0.7576</b>	0.7532	0.7278	1.4189
31	<b>0.6215</b>	0.6172	0.5307	1.6204	81	<b>4.5890</b>	4.5885	4.4751	5.6660
32	<b>4.5631</b>	4.5559	4.5147	5.2915	82	<b>6.3865</b>	6.3751	6.3194	7.4986
33	<b>7.0305</b>	6.9845	6.9944	8.0498	83	<b>6.1873</b>	6.1780	6.0546	7.2406
34	<b>9.9680</b>	9.8466	9.9100	11.3081	84	<b>2.9913</b>	2.9078	2.9352	4.7038
35	<b>0.8044</b>	0.7829	0.7239	2.0848	85	<b>10.6753</b>	10.6411	10.6368	11.7620
36	<b>-4.8340</b>	-4.9150	-4.8925	-3.5569	86	<b>-7.7191</b>	-7.7389	-7.7566	-6.2191
37	<b>4.9906</b>	4.9789	4.9619	5.9426	87	<b>11.0850</b>	11.0523	11.0027	12.4703
38	<b>-7.6116</b>	-7.6907	-7.7910	-6.0688	88	10.3312	10.2796	<b>10.3872</b>	12.3348
39	<b>15.8386</b>	15.8279	15.8049	16.7978	89	<b>8.9459</b>	8.9313	8.8954	10.1933
40	<b>0.3786</b>	0.3694	0.2224	1.7918	90	<b>7.6131</b>	7.6069	7.5995	8.3429
41	<b>-2.2577</b>	-2.2625	-2.3401	-0.9587	91	<b>4.7490</b>	4.6998	4.6915	5.9225
42	<b>0.8659</b>	0.8557	0.8168	1.9483	92	<b>10.1188</b>	10.0803	10.0249	11.2766
43	<b>12.2575</b>	12.2277	12.2335	13.1937	93	<b>-1.0054</b>	-1.0080	-1.0731	-0.2399
44	<b>4.7063</b>	4.7037	4.6871	5.1994	94	<b>-2.3783</b>	-2.3824	-2.5552	-1.1306
45	<b>7.1076</b>	7.0944	7.0682	8.0319	95	<b>1.3148</b>	1.2884	1.2675	2.9579
46	<b>-1.0208</b>	-1.0213	-1.3427	0.4856	96	<b>-2.4070</b>	-2.4142	-2.4486	-1.4270
47	<b>5.6107</b>	5.5709	5.5854	6.8776	97	<b>3.6744</b>	3.6589	3.6323	4.6726
48	<b>7.9571</b>	7.9486	7.9025	9.0604	98	<b>1.8353</b>	1.8244	1.7627	3.2556
49	<b>5.8116</b>	5.8047	5.7952	6.4533	99	<b>-5.3745</b>	-5.4029	-5.4231	-4.3001
50	<b>2.9529</b>	2.9124	2.8474	4.2576	100	<b>7.8266</b>	7.8138	7.7616	8.7882

Note. The column Case contains the instance number, columns  $LB_1$ ,  $LB_2$ , and  $LB_3$  are the lower bounds obtained by Algorithm 1, 2, and the finite-scenario approach, respectively, and the columns  $UB_1$  is the upper bound obtained by perspectification approach. In this table, for each instance, the best lower bound is in boldface.

# CHAPTER 4

## Quadratic Optimization Through the Lens of Adjustable Robust Optimization

Ralph Tyrrell Rockafellar:

*“The great watershed in optimization isn’t between linearity and nonlinearity, but convexity and non-convexity.”*

### 4.1 Introduction

Various practical problems in different domains, including financial mathematics [89], machine learning [47], resource allocation [71], computer vision [31], game theory [32], robotic systems [77], graph theory [60], and image processing [41], to mention few, can be formulated as quadratic optimization problems. Thus, developing efficient techniques to solve general quadratic optimization problems is of great importance.

Let us consider a quadratic optimization (QO) problem of the form:

$$\min_{x \in \mathcal{X}} x^\top Q x + c^\top x, \quad (\text{QO})$$

where  $\mathcal{X} \subseteq \mathbb{R}^{n_x}$  is a nonempty convex set,  $Q \in \mathbb{R}^{n_x \times n_x}$  is a real matrix, and  $c \in \mathbb{R}^{n_x}$  is a real vector. Without loss of generality, we assume that  $Q$  is a symmetric matrix. If  $Q$  is a positive semi-definite matrix, we have a convex QO, which is solvable in polynomial time [80, 98]. In contrast, even when  $Q$  has only one negative eigenvalue, (QO) is NP-hard [94]. Besides, identifying local minimizers of (QO) over a polyhedron is not simpler than finding global minimizers from a complexity perspective [1].

Due to the NP-hardness of indefinite QO problems, there has been a lot of research on constructing upper bounds via finding “good” solutions [18, 51], and lower bounds to identify the quality of a candidate solution, which are mainly based on linear or conic approximations [90, 102, 125]. A customary way to approximate a QO problem is by relaxing it into linear optimization problems, which is achieved through Reformulation-Linearization Techniques (RLT) [3, 110]. For an overview of RLTs, we refer the reader to the chapter [108] and the references therein.

Among the conic relaxations, copositive relaxations have been considered the most powerful as it was shown that they result in tight bounds [33, 43]. In such relaxations, the primary computational challenge shifts to deal with the copositive cone using tractable inner and outer approximations [42, 63, 78], or use a KKT-based branch-and-bound method [48].

Another important conic relaxation for QO problems is the positive semi-definite relaxations. In the last thirty years, the field of semi-definite optimization (SDO) has undergone significant and swift advancement [118]. The SDO framework has led to many semi-definite relaxations due to their efficiency; these relaxations are reviewed and compared in [8, 116, 131]. Moreover, [44, 45] develop branch-and-bound approaches based on semi-definite relaxations to solve a QO problem.

In addition to directly approximating QOs, a research direction is to reformulate them into other well-studied problems. In [70] and [120], the authors show how a QO problem is reformulated as a mixed-integer linear optimization (MILO) problem. Moreover, since any quadratic function can be written as the difference between two convex quadratic functions (see, e.g. [59] and [95] for different representations and their properties), a QO can be reformulated as a difference-of-convex (DC) optimization problem.

Next to methods developed for general QO problems, there are techniques to solve or approximate special classes. One class is when the matrix  $Q$  has a few negative eigenvalues. In [46], the authors propose a solution scheme that involves solving a series of convex QO problems over the original feasible region. Additionally, [86] introduces an alternative direction-based method to solve QO problems in this class.

Another class is standard QO problems, where the feasible region is the unit simplex. For

more details on lower bound approximations for this class of QO problems, we refer the reader to [34, 35, 37, 61, 62, 106].

In this paper, we focus on the relation between QO problems and adjustable robust optimization problems. The adjustable robust optimization (ARO) framework, initially introduced in [14], has gained significant attention among researchers due to its ability to handle decision-making problems in the presence of uncertain parameters. This approach involves adaptive decision-making by considering two types of decision variables: static and adjustable decisions. Static (or ‘here-and-now’) decisions are made based on available information, while adjustable (or ‘wait-and-see’) decisions are made in response to the actual values of uncertain parameters. In recent years, the ARO framework has been successfully applied to tackle complex optimization problems such as convex maximization [105] and bi-linear optimization [130].

To obtain an approximate solution for an ARO problem, various techniques, such as the finite scenario approach [66], partitioning method [24, 97], Fourier-Motzkin elimination [128], decision rules [58], can be employed, particularly in the case of linear ARO problems. By using these methods, one can estimate the optimal value or obtain an approximated solution for the original problem. For more information on ARO, we refer to the tutorial by [54] and the survey paper by [123].

However, while linear ARO has effective approximation tools, the literature sparsely covers non-linear ARO problems due to their inherent complexity. In [53], the authors show a class of non-linear ARO problems featuring a polyhedral uncertainty set that can be transformed into an equivalent linear ARO problem, thereby enabling the application of approximations techniques available for linear cases. In a recent study [76], the authors employ Fenchel’s duality to convert a non-linear ARO problem into its dual formulation and introduce a cutting-plane algorithm to find locally robust solutions.

In this chapter, we make a four-fold contribution to the literature to connect the two fields of quadratic optimization and adjustable robust optimization. First, we show that any QO problem can be reformulated as a disjoint bi-convex quadratic optimization problem. Using this new reformulation, we further show that any QO problem can be reformulated as an

ARO problem, where the objective functions and constraints are convex quadratic on the decision variables and linear on the uncertain parameter. Moreover, the ARO reformulation has right-hand-side uncertainty, implying that available ARO techniques are applicable to approximate it.

Second, we show how one can interpret an approximation of the ARO reformulation on the original QO problem. More specifically, we prove that applying a structured affine decision rule to approximate the ARO formulation is equivalent to applying an RLT to approximate the disjoint bi-convex reformulation.

Third, we design an algorithm to construct a bound on the optimal value of (QO). More specifically, we apply a decision-rule approximation to obtain a lower bound. Then, based on the solution and the structure of the ARO problem, we construct “good” feasible solutions. In the final step, we apply the mountain-climbing procedure to improve the quality of the solution.

Finally, we conduct an extensive numerical experience to illustrate the efficiency of our algorithm. Based on the numerical results, we see that the solution obtained from the algorithm is close to optimum and, in most cases, has the optimality gap of 1%. Regarding speed, our algorithm is computationally efficient and significantly outperforms the available off-the-self solvers.

The main results of this chapter have been appeared in

- [75] Khademi, A., & Marandi. (2024). Quadratic Optimization Through the Lens of Adjustable Robust Optimization. Submitted.

The rest of this chapter is structured as follows: in Section 4.1.1, we define the notation used throughout this chapter. Section 4.2 introduces the reformulation of a QO problem as a bi-convex optimization problem and outlines its equivalent ARO problem. In Section 4.3, we approximate this problem using available techniques and prove the equivalence to an RLT for the original QO problem. Subsequently, in Section 4.4, we design an algorithm that provides a near-optimal solution for a QO problem using the ARO reformulation. Section 4.5 presents numerical results, demonstrating the efficiency of our ARO-based algorithm, particularly for large-sized instances. Finally, in Section 4.6, we summarize our findings and

present our conclusions.

#### 4.1.1 Notation

In this section, we introduce notations used in this chapter. For a symmetric matrix  $B$ , we use  $B \geq 0$  ( $B > 0$ ) to show  $B$  is positive semi-definite (positive definite), i.e., that it has non-negative (positive) eigenvalues. The smallest and largest eigenvalues of a symmetric matrix  $B$  are denoted by  $\lambda_{\min}(B)$  and  $\lambda_{\max}(B)$ , respectively. For a given matrix  $B$ , and integers  $i$  and  $j$ , we denote by  $B_i$ ,  $B^j$ , and  $B_{ij}$ , the  $i$ -th row, the  $j$ -th column, and the  $ij$ -th entry of  $B$ , respectively. For a matrix  $B$ ,  $\text{vec}(B)$  denotes the vector formed by concatenating all of the rows of the matrix  $B$ . We use  $(\cdot)^\top$  to refer to the transpose operator for both matrices and vectors. We denote the  $n \times n$  identity matrix by  $I_n$ , the vector of all ones by  $e$ , and the  $i$ -th unit vector by  $e_i$ . To avoid overcomplicating notation, we do not specify the dimensions of  $e$  and  $e_i$  but make sure they are always evident from the context. We misuse the notation and denote the real number zero, the vector of all zeroes, and the matrix of all zeroes by  $0$ .

We use  $\mathbb{R}^n$  to refer to the  $n$ -dimensional real-valued Euclidean space, where  $\|\cdot\|_2$  is the Euclidean norm. The standard or unit simplex in  $\mathbb{R}^n$ , given by  $\{x \in \mathbb{R}^n : e^\top x = 1, x \geq 0\}$ , is denoted by  $\Delta$ .

## 4.2 New Reformulations for Quadratic Optimization Problems

In this section, we propose two reformulations for a quadratic optimization problem (QO). We first show how we can reformulate (QO) to a disjoint bi-convex quadratic optimization problem. Using this reformulation, we further provide an equivalent adjustable robust optimization problem. So, we start with the following theorem.

**Theorem 8.** *Let  $Q^+, -Q^- \geq 0$ , and  $\mathcal{X} \subseteq \mathbb{R}^{n_x}$  be an arbitrary set. Then,*

$$\min_{x \in \mathcal{X}} x^\top (Q^+ + Q^-)x + c^\top x \tag{4.1}$$



is equivalent to

$$\min_{x,y \in \mathbb{R}^{n_x}} \left\{ \frac{1}{2}x^\top Q^+ x + \frac{1}{2}y^\top Q^+ y + x^\top Q^- y + \frac{1}{2}c^\top x + \frac{1}{2}c^\top y : x, y \in \mathcal{X} \right\}. \quad (\text{Bi-QO})$$

*Proof.* It is clear that

$$\begin{aligned} & \min_{x \in \mathcal{X}} x^\top (Q^+ + Q^-)x + c^\top x \\ &= \min_{x,y \in \mathbb{R}^{n_x}} \left\{ \frac{1}{2}x^\top Q^+ x + \frac{1}{2}y^\top Q^+ y + x^\top Q^- y + \frac{1}{2}c^\top x + \frac{1}{2}c^\top y : x = y, x, y \in \mathcal{X} \right\} \\ &\geq \min_{x,y \in \mathbb{R}^{n_x}} \left\{ \frac{1}{2}x^\top Q^+ x + \frac{1}{2}y^\top Q^+ y + x^\top Q^- y + \frac{1}{2}c^\top x + \frac{1}{2}c^\top y : x, y \in \mathcal{X} \right\}, \end{aligned}$$

where the inequality is due to the fact that the feasible region of the last optimization problem is contained in the feasible region of the middle optimization problem.

To show “ $\leq$ ”, we use the negative semi-definiteness of  $Q^-$ . Let  $x, y \in \mathbb{R}^{n_x}$  be arbitrary. Because  $Q^- \leq 0$ , we have  $(x - y)^\top Q^- (x - y) \leq 0$ . Hence,  $x^\top Q^- x + y^\top Q^- y \leq 2x^\top Q^- y$ . This implies that for any  $x, y \in \mathbb{R}^{n_x}$ ,

$$x^\top (Q^+ + Q^-)x + y^\top (Q^+ + Q^-)y \leq x^\top Q^+ x + y^\top Q^+ y + 2x^\top Q^- y.$$

So,

$$x^\top (Q^+ + Q^-)x + y^\top (Q^+ + Q^-)y + c^\top x + c^\top y \leq x^\top Q^+ x + y^\top Q^+ y + 2x^\top Q^- y + c^\top x + c^\top y.$$

Now, by taking the minimum over  $x, y \in \mathcal{X}$ , we have

$$\begin{aligned} & \min_{x \in \mathcal{X}} \{x^\top (Q^+ + Q^-)x + c^\top x\} + \min_{y \in \mathcal{X}} \{y^\top (Q^+ + Q^-)y + c^\top y\} \\ &\leq \min_{x,y \in \mathcal{X}} \{x^\top Q^+ x + y^\top Q^+ y + 2x^\top Q^- y + c^\top x + c^\top y\}. \end{aligned}$$

The fact that

$$\min_{x \in \mathcal{X}} \{x^\top (Q^+ + Q^-)x + c^\top x\} = \min_{y \in \mathcal{X}} \{y^\top (Q^+ + Q^-)y + c^\top y\},$$

completes the proof.  $\square$

It is worth noting that the proof of Theorem 8 does not rely on the specific structure of the feasible set  $\mathcal{X}$ . If  $\mathcal{X}$  is convex, then the proposition asserts that any indefinite **QO** can be reformulated as a disjoint bi-convex quadratic optimization problem, where the variables  $x$  and  $y$  are linked only in the objective function.

**Remark 4.2.1.** In **(QO)**, we can assume, without loss of generality, that the matrix  $Q$  is symmetric; otherwise, we can replace the objective function with  $x^\top (\frac{Q^\top + Q}{2})x + c^\top x$ . Now, for a symmetric matrix  $Q$ , we know that the eigenvalues are real [92]. So, for an indefinite matrix  $Q$ , we can construct the matrices in Theorem 8 as follows:

Representation 1:

$$Q^+ := Q - (\lambda_{\min}(Q) - \epsilon)I, \text{ and } Q^- := (\lambda_{\min}(Q) - \epsilon)I,$$

Representation 2:

$$Q^+ := (\lambda_{\max}(Q) + \epsilon)I, \text{ and } Q^- := Q - (\lambda_{\max}(Q) + \epsilon)I,$$

where  $\epsilon$  is a small positive constant chosen to ensure that  $Q^+, -Q^- \geq 0$ . Later, we discuss which of these formulations can provide a better approximation.  $\square$

The next proposition aims to establish a relation between the optimal solutions of the **(QO)** and **(Bi-QO)** problems, showcasing how solutions from one problem can be used to obtain optimal solutions for the other.

**Proposition 4.2.1.** *Let  $Q = Q^+ + Q^-$  where  $Q \in \mathbb{R}^{n_x \times n_x}$ , and  $Q^+, -Q^- > 0$ . If  $x^*$  is an optimal solution of **(QO)**, then  $(x^*, x^*)$  is an optimal solution of **(Bi-QO)**. If  $(\hat{x}, \hat{y})$  is an optimal solution of **(Bi-QO)**, then  $\hat{x}$  and  $\hat{y}$  are optimal solutions of **(QO)**.*

*Proof.* Suppose  $x^*$  is an optimal solution to **(QO)**. It is clear that the tuple  $(x^*, x^*)$  is also an optimal solution to **(Bi-QO)**. To prove the reverse direction, assume that  $(\hat{x}, \hat{y})$  is an

optimal solution of (Bi-QO). Therefore,

$$\begin{aligned} \frac{1}{2} (\hat{x}^\top Q^+ \hat{x} + \hat{y}^\top Q^+ \hat{y} + c^\top \hat{x} + c^\top \hat{y}) + \hat{x}^\top Q^- \hat{y} &\leq \frac{1}{2} (\hat{x}^\top Q^+ \hat{x} + \hat{x}^\top Q^+ \hat{x} + c^\top \hat{x} + c^\top \hat{x}) + \hat{x}^\top Q^- \hat{x}, \\ \frac{1}{2} (\hat{x}^\top Q^+ \hat{x} + \hat{y}^\top Q^+ \hat{y} + c^\top \hat{x} + c^\top \hat{y}) + \hat{x}^\top Q^- \hat{y} &\leq \frac{1}{2} (\hat{y}^\top Q^+ \hat{y} + \hat{y}^\top Q^+ \hat{y} + c^\top \hat{y} + c^\top \hat{y}) + \hat{y}^\top Q^- \hat{y}. \end{aligned} \quad (4.2)$$

When these inequalities are combined, the result is:

$$2\hat{x}^\top Q^- \hat{y} \leq \hat{x}^\top Q^- \hat{x} + \hat{y}^\top Q^- \hat{y}. \quad (4.3)$$

Now, note that  $\hat{x}^\top (-Q^-) \hat{y} = \left( (-Q^-)^{\frac{1}{2}} \hat{x} \right)^\top \left( (-Q^-)^{\frac{1}{2}} \hat{y} \right)$ , where  $(-Q^-)^{\frac{1}{2}}$  is the square roots of the matrix  $(-Q^-)$ . Therefore, we can apply the Cauchy-Schwarz inequality, which implies that

$$\begin{aligned} 2\hat{x}^\top (-Q^-) \hat{y} &\leq 2 \left\| (-Q^-)^{\frac{1}{2}} \hat{x} \right\| \cdot \left\| (-Q^-)^{\frac{1}{2}} \hat{y} \right\| \\ &= 2\sqrt{\hat{x}^\top (-Q^-) \hat{x}} \sqrt{\hat{y}^\top (-Q^-) \hat{y}} \\ &\leq \hat{x}^\top (-Q^-) \hat{x} + \hat{y}^\top (-Q^-) \hat{y}, \end{aligned} \quad (4.4)$$

where the reason for the last inequality is that, for any two non-negative scalars  $a$  and  $c$ ,  $2\sqrt{ac} \leq (a + c)$ . Hence, we have

$$2\hat{x}^\top Q^- \hat{y} \geq \hat{x}^\top Q^- \hat{x} + \hat{y}^\top Q^- \hat{y}. \quad (4.5)$$

Thus, by (4.3) and (4.5), we obtain that

$$2\hat{x}^\top Q^- \hat{y} = \hat{x}^\top Q^- \hat{x} + \hat{y}^\top Q^- \hat{y}, \quad (4.6)$$

which is the same as

$$(\hat{x} - \hat{y})^\top Q^- (\hat{x} - \hat{y}) = 0. \quad (4.7)$$

From  $-Q^- > 0$ , we have  $\hat{x} - \hat{y} = 0$ , i.e.  $\hat{x} = \hat{y}$ . So,  $\hat{x}$  is an optimal solution of (QO).  $\square$

A straightforward result, which follows from the proof of the above proposition, is that when  $(\hat{x}, \hat{y})$  represents an optimal solution to problem (Bi-QO),  $\hat{x}$  and  $\hat{y}$  must be equal. The next corollary states this fact.

**Corollary 4.2.1.** *Let  $Q = Q^+ + Q^-$  where  $Q \in \mathbb{R}^{n_x \times n_x}$ , and  $Q^+, -Q^- > 0$ . If  $(x^*, y^*)$  is an optimal solution to problem (Bi-QO), then  $x^* = y^*$ .  $\square$*

From now on, let us restrict the feasible region of (QO) to polytopes. So, we assume that  $\mathcal{X} = \{x \in \mathbb{R}^{n_x} \mid Ax = b, x \geq 0\}$  for some  $A \in \mathbb{R}^{m_x \times n_x}$  and  $b \in \mathbb{R}^{m_x}$ , so that  $\mathcal{X}$  is compact. In the next theorem, we show that we can reformulate (QO) problem to an adjustable robust optimization problem.

**Theorem 9.** *Let  $Q = Q^+ + Q^-$  where  $Q \in \mathbb{R}^{n_x \times n_x}$ , and  $Q^+, -Q^- \geq 0$ . Assume that  $\mathcal{X} = \{x \in \mathbb{R}^{n_x} \mid Ax = b, x \geq 0\}$  is non-empty compact. Then, the optimal value of (QO) is equal to the optimal value of the following problem:*

$$\begin{aligned} & \max_{\tau \in \mathbb{R}} \tau \\ & \text{s.t. } \forall x \in \mathcal{X}, \exists (u_x, w_x) : \begin{cases} \frac{1}{2}x^\top Q^+ x + \frac{1}{2}c^\top x - \frac{1}{2}u_x^\top Q^+ u_x + b^\top w_x \geq \tau, \\ A^\top w_x - Q^+ u_x \leq Q^- x + \frac{1}{2}c. \end{cases} \end{aligned} \quad (\text{ARO-QO})$$

*Proof.* Based on the assumption, we have that (QO) is equivalent to

$$\min_{x \in \mathcal{X}} x^\top (Q^+ + Q^-) x + c^\top x,$$

which is, using Theorem 8, equivalent to

$$\min_{x, y \in \mathbb{R}^{n_x}} \left\{ \frac{1}{2}x^\top Q^+ x + \frac{1}{2}y^\top Q^+ y + x^\top Q^- y + \frac{1}{2}c^\top x + \frac{1}{2}c^\top y : x, y \in \mathcal{X} \right\}. \quad (4.8)$$

We can write (4.8) as

$$\min_{x \in \mathcal{X}} \left\{ \frac{1}{2}x^\top Q^+ x + \frac{1}{2}c^\top x + \min_{y \in \mathcal{X}} \frac{1}{2}y^\top Q^+ y + x^\top Q^- y + \frac{1}{2}c^\top y \right\}. \quad (4.9)$$

We consider the inner minimization problem over  $y$  for a given  $x \in \mathcal{X}$ . Since  $\mathcal{X}$  non-empty

compact, we can apply Dorn duality [55], and rewrite (4.9) as follows:

$$\begin{aligned} \min_{x \in \mathcal{X}} \frac{1}{2}x^\top Q^+ x + \frac{1}{2}c^\top x + \max_{u_x, w_x} -\frac{1}{2}u_x^\top Q^+ u_x + b^\top w_x \quad (4.10) \\ \text{s.t. } A^\top w_x - Q^+ u_x \leq Q^- x + \frac{1}{2}c. \end{aligned}$$

Let  $x \in \mathcal{X}$ . If the inner maximization is infeasible, its optimal value is  $-\infty$ , implying that (4.10) is unbounded. So, in this case, (QO) is unbounded, which contradicts the compactness of  $\mathcal{X}$ . So, for any  $x \in \mathcal{X}$ , there is a feasible  $(u_x, w_x)$  for the inner maximization. Thus, using the epigraph reformulation of the objective function, we can rewrite (4.10) as

$$\max_{\tau} \left\{ \tau \mid \forall x \in \mathcal{X}, \exists (u_x, w_x) : \begin{aligned} & \frac{1}{2}x^\top Q^+ x + \frac{1}{2}c^\top x - \frac{1}{2}u_x^\top Q^+ u_x + b^\top w_x \geq \tau, \\ & A^\top w_x - Q^+ u_x \leq Q^- x + \frac{1}{2}c. \end{aligned} \right\}, \quad (4.11)$$

which completes the proof. □

Problem (ARO-QO) is a quadratic ARO problem with fixed recourse and right-hand-side uncertainty. In this problem,  $\tau$  is the static variable,  $x \in \mathcal{X}$  is the uncertain parameter, and  $(u_x, w_x)$  is the adjustable variable. The adjustable variables can be seen as functions of  $x$ , and are known as decision policies [76, 123].

It is important to note that concave QO, i.e., when dealing with a negative semi-definite matrix  $Q$ , is characterized by its NP-hard nature. This complexity primarily arises from the crucial relationship between achieving optimality and enumerating the extreme points within the feasible region [93]. Predominant strategies for addressing concave QO problems typically involve cutting plane methods, branch and bound approaches, or iterative computational techniques [6, 2, 49, 96]. Furthermore, recent studies in this area have focused on establishing bounds from a robust optimization perspective [105], and some have adopted approaches based on gradient descent principles [17]; the application of these techniques has been instrumental in deriving high-quality bounds for the optimal solution.

In the subsequent corollary, we present the ARO reformulation for concave QO.

**Corollary 4.2.2.** *Let  $Q$  be a negative semi-definite matrix. Assume that*

$\mathcal{X} = \{x \in \mathbb{R}^{n_x} \mid Ax = b, x \geq 0\}$  is non-empty compac. Then, the optimal value of (QO) is equal to the optimal value of the following problem:

$$\begin{aligned} & \max_{\tau \in \mathbb{R}} \tau \\ & \text{s.t. } \forall x \in \mathcal{X}, \exists w_x : \begin{cases} \frac{1}{2}c^\top x + b^\top w_x \geq \tau, \\ A^\top w_x \leq Q^- x + \frac{1}{2}c. \end{cases} \end{aligned} \quad (4.12)$$

*Proof.* Immediate result from Theorem 9 by setting  $Q^+ := 0$  and  $Q^- := Q$ .  $\square$

Note that (4.12) is a linear adjustable robust optimization problem and all techniques in the literature can be used to solve or approximate it.

**Remark 4.2.2.** Tables 4.8 and 4.9 present the equivalent ARO formulations if the polytope  $\mathcal{X}$  is formulated in another form than canonical.  $\square$

To approximate (ARO-QO) problem, we can use customary techniques to deal with adjustable variables, such as eliminating the adjustable variables via Fourier-Motzkin Elimination or using decision rules to approximate the adjustable variables. In the next section, we focus on such approximation methods.

## 4.3 ARO Based Approximations

In this section, we show how the available techniques to approximate an ARO problem can be employed and what their interpretations are concerning (QO).

### 4.3.1 Decision Rules

In (ARO-QO) problem, the adjustable variables  $u_x$  and  $w_x$  are, in essence, functions of the uncertain parameter  $x$ . One of the popular methods to approximate an ARO problem is by restricting the adjustable variables to belong to a specific class of functions. For example, we can restrict them to be constants, resulting in a static formulation, or to be affine, known as affine decision rule (ADR), which is a good approximation for linear ARO problems (see, e.g., [25] and [26, 28]).

Since (ARO-QO) contains a non-linear convex term  $u_x^\top Q^+ u_x$ , using ADR to approximate  $u_x$  results in an intractable approximation. Therefore, to have a tractable approximation, we apply a hybrid decision rule. More specifically, we restrict  $u_x$  to be constant and  $w_x$  to be affine:

$$u_x := u \text{ and } w_x := z + Zx,$$

where  $u \in \mathbb{R}^{n_x}$ ,  $z \in \mathbb{R}^{m_x}$ , and  $Z \in \mathbb{R}^{m_x \times n_x}$  are static variables. Using this decision rule in (ARO-QO) leads to the following static robust counterpart, which gives a lower bound on the optimal value of (QO):

$$\max_{u, z, Z, \tau} \left\{ \tau \left| \begin{array}{l} \frac{1}{2}x^\top Q^+ x + \frac{1}{2}c^\top x - \frac{1}{2}u^\top Q^+ u + b^\top(z + Zx) \geq \tau, \quad \forall x \in \mathcal{X} \\ A^\top(z + Zx) - Q^+ u \leq Q^- x + \frac{1}{2}c, \quad \forall x \in \mathcal{X} \end{array} \right. \right\}, \quad (4.13)$$

where  $u$ ,  $z$ , and  $Z$  are simultaneously optimized together with the static decision variable  $\tau$ .

In the previous section, we demonstrated that the (QO) problem is equivalent to both the (Bi-QO) and (ARO-QO) problems. In Problem (4.13) we approximate (ARO-QO). In the rest of this section, we show that this approximation, in fact, is equivalent to applying a reformulation-linearization (RL) technique to (Bi-QO).

RL techniques have also been considered in the literature to approximate an ARO problem. More specifically, it is shown in [4] that using duality techniques, a linear ARO problem can be reformulated as a bi-linear optimization problem. The authors then show that using an RL technique to approximate the bi-linear optimization reformulation is equivalent to applying ADR to the original problem. In [130], the same results are shown for disjoint bi-linear problems with convex feasible regions.

Considering (Bi-QO), using the RL technique proposed in [109] and [110] results in the

following linear optimization problem:

$$\begin{aligned}
& \min_{\gamma, x, y} \frac{1}{2} (x^\top Q^+ x + y^\top Q^+ y + c^\top x + c^\top y) + \sum_{i,j=1}^{n_x} Q_{ij}^- \gamma_{ij} \\
& \text{s.t. } Ax = b, \\
& \quad Ay = b, \\
& \quad A\gamma = by^\top, \\
& \quad A\gamma^\top = bx^\top, \\
& \quad x \geq 0, y \geq 0, \gamma \geq 0.
\end{aligned} \tag{4.14}$$

In the next theorem, we show that (4.14) is the dual of the deterministic reformulation of the robust counterpart (4.13).

**Theorem 10.** *Assume that  $\mathcal{X}$  is a non-empty compact set. Then, the optimal value of (4.13) is equal to the optimal value of (4.14).*

*Proof.* We can rewrite (4.13) as

$$\max_{u, z, Z, \tau} \left\{ \tau \left| \begin{array}{l} \min_{x \in \mathcal{X}} \left\{ \frac{1}{2} x^\top Q^+ x + \left( \frac{1}{2} c^\top + b^\top Z \right) x \right\} + b^\top z - \frac{1}{2} u^\top Q^+ u \geq \tau, \\ \min_{x \in \mathcal{X}} \left\{ (-A^\top Z + Q^-)_i x \right\} + \left( \frac{1}{2} c + Q^+ u - A^\top z \right)_i \geq 0, \quad i = 1, \dots, n_x \end{array} \right. \right\}. \tag{4.15}$$

Since  $\mathcal{X}$  is a polytope, the inner minimizations are convex optimization problems. Since  $\mathcal{X}$  is a non-empty compact set, strong duality holds [39, 55]. Therefore, (4.15) is equivalent to

$$\begin{aligned}
& \max_{u, z, Z, \tau} \tau \\
& \text{s.t. } \max_{\alpha, \beta} \left\{ b^\top \beta - \frac{1}{2} \alpha^\top Q^+ \alpha \mid A^\top \beta - Q^+ \alpha \leq (b^\top Z)^\top + \frac{1}{2} c \right\} + b^\top z - \frac{1}{2} u^\top Q^+ u \geq \tau, \\
& \quad \max_{\theta^i} \left\{ b^\top \theta^i \mid A^\top \theta^i \leq \left( (-A^\top Z + Q^-)_i \right)^\top \right\} + \left( \frac{1}{2} c + Q^+ u - A^\top z \right)_i \geq 0, \quad i = 1, \dots, n_x.
\end{aligned} \tag{4.16}$$



We can omit the inner maximization operator in the above constraints. Thus, we have

$$\begin{aligned}
 & \max_{u,z,Z,\alpha,\beta,\theta} \quad b^\top \beta - \frac{1}{2} \alpha^\top Q^+ \alpha + b^\top z - \frac{1}{2} u^\top Q^+ u \\
 & \text{s.t.} \quad A^\top \beta - Q^+ \alpha \leq (b^\top Z)^\top + \frac{1}{2} c, \\
 & \quad b^\top \theta^i + \left(\frac{1}{2} c + Q^+ u - A^\top z\right)_i \geq 0, \quad i = 1, \dots, n_x, \\
 & \quad A^\top \theta^i \leq \left((-A^\top Z + Q^-)_i\right)^\top, \quad i = 1, \dots, n_x.
 \end{aligned} \tag{4.17}$$

Now, we show that (4.17) is the dual problem of (4.14). To do this, we first write (4.14) in the matrix form:

$$\begin{aligned}
 & \min_{vec(\gamma), x, y} \quad \frac{1}{2} \begin{pmatrix} x \\ y \\ vec(\gamma) \end{pmatrix}^\top \begin{pmatrix} Q^+ & 0 & 0 \\ 0 & Q^+ & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ vec(\gamma) \end{pmatrix} + \begin{pmatrix} \frac{c}{2} \\ \frac{c}{2} \\ vec(Q^-) \end{pmatrix}^\top \begin{pmatrix} x \\ y \\ vec(\gamma) \end{pmatrix} \\
 & \text{s.t.} \quad \begin{pmatrix} A & 0 & 0 \\ 0 & A & 0 \\ 0 & B & C \\ B & 0 & D \end{pmatrix} \begin{pmatrix} x \\ y \\ vec(\gamma) \end{pmatrix} = \begin{pmatrix} b \\ b \\ 0 \\ 0 \end{pmatrix}, \\
 & \quad \begin{pmatrix} x \\ y \\ vec(\gamma) \end{pmatrix} \geq 0,
 \end{aligned} \tag{4.18}$$

$$\text{where } B := - \begin{pmatrix} b_1 I_{n_x} \\ b_2 I_{n_x} \\ \vdots \\ b_{m_x} I_{n_x} \end{pmatrix}, C := \begin{pmatrix} A_{11} I_{n_x} & A_{12} I_{n_x} & \dots & A_{1n_n} I_{n_x} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m_x 1} I_{n_x} & A_{m_x 2} I_{n_x} & \dots & A_{m_x n_x} I_{n_x} \end{pmatrix}, \text{ and } D := \begin{pmatrix} A_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & A_1 \\ \vdots & & \vdots \\ A_{m_x} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & A_{m_x} \end{pmatrix}.$$

The dual of (4.18) is

$$\begin{aligned} \max_{Y, W} \quad & -\frac{1}{2} Y^\top \begin{pmatrix} Q^+ & 0 & 0 \\ 0 & Q^+ & 0 \\ 0 & 0 & 0 \end{pmatrix} Y + \begin{pmatrix} b \\ b \\ 0 \\ 0 \end{pmatrix}^\top W \\ \text{s.t.} \quad & - \begin{pmatrix} Q^+ & 0 & 0 \\ 0 & Q^+ & 0 \\ 0 & 0 & 0 \end{pmatrix} Y + \begin{pmatrix} A & 0 & 0 \\ 0 & A & 0 \\ 0 & B & C \\ B & 0 & D \end{pmatrix}^\top W \leq \begin{pmatrix} \frac{c}{2} \\ \frac{c}{2} \\ \text{vec}(Q^-) \end{pmatrix}. \end{aligned} \tag{4.19}$$

Setting

$$Y \equiv \begin{pmatrix} \alpha \\ u \\ Y^3 \end{pmatrix}, \quad W \equiv \begin{pmatrix} \beta \\ z \\ \text{vec}(\theta) \\ \text{vec}(Z) \end{pmatrix},$$

(4.19) is the matrix form of (4.17). Hence, (4.14) is the dual of the deterministic reformulation of (4.13).  $\square$

We have shown that applying the RL technique to the disjoint bi-convex reformulation

(Bi-QO) is equivalent to using a hybrid static-affine decision rule to approximate the adjustable robust reformulation (ARO-QO). As mentioned, the ADR approximation is shown to be an efficient approximation for a class of linear ARO problems. For example, for a linear ARO problem with a right-hand-side uncertainty, the ADR approximation is tight when the uncertainty set is simplex [19]. The translation of this setting for the original problem (QO) is to have a concave quadratic objective function with  $\mathcal{X}$  being a simplex. Even though this class seems not to be interesting (we know that enumerating the  $n_x$  number of vertices provides us with the optimal value), it generates insights into the quality of (4.14).

As mentioned in Remark 4.2.1, we can have multiple representations of  $Q$  based on  $Q^+$  and  $Q^-$ . Considering Representation 1, we see that  $Q^-$  is a diagonal matrix, but  $Q^+$  has a similar density as  $Q$ . Therefore, in (ARO-QO), all entries of  $u_x$  are linked together via  $Q^+u_x$ . However, in Representation 2,  $Q^+$  is a diagonal matrix, implying that the entries of  $u_x$  are only linked together via  $u_x^\top Q^+u_x$  and not in the constraints. In the numerical result section, we will use this representation.

### 4.3.2 Fourier–Motzkin Elimination

In linear ARO problems with fixed recourse, an adjustable variable may be eliminated by employing Fourier-Motzkin elimination (EME). This approach allows for the effective handling of problems involving a limited number of adjustable variables [128].

Note that for a given  $x \in \mathcal{X}$ , in (ARO-QO), we have the ability to eliminate the adjustable variable  $w_x \in \mathbb{R}^{m_x}$ . We assume without loss of generality that  $b \geq 0$ . Let  $k \in \{1, \dots, m_x\}$ . To eliminate  $w_{x_k}$ , the  $k$ -th component of the vector  $w_x$ , we first isolate it in the constraints:

$$\begin{cases} b_k w_{x_k} \geq \tau - \frac{1}{2}x^\top Q^+x - \frac{1}{2}c^\top x + \frac{1}{2}u_x^\top Q^+u_x - \sum_{\substack{j=1 \\ j \neq k}}^{m_x} b_j w_{x_j}, \\ A_{ki} w_{x_k} \leq (Q^-x + \frac{1}{2}c + Q^+u_x)_i - \sum_{\substack{j=1 \\ j \neq k}}^{m_x} A_{ji} w_{x_j}. \end{cases} \quad i = 1, \dots, m_x. \quad (4.20)$$

Since  $\mathcal{X} = \{x \mid Ax = b, x \geq 0\}$  is non-empty, so we cannot have  $b_k > 0$  and  $A_{ki} \leq 0$  for any  $i = 1, \dots, m_x$ .

If  $A_{ki} \neq 0$  and  $b_k > 0$ , then both sides of their respective constraints can be divided by  $A_{ki}$

and  $b_k$ . This yields an equivalent representation of the feasible region, involving the following constraints:

$$\begin{aligned}
w_{x_k} &\geq \frac{1}{b_k} \left( \tau - \frac{1}{2} x^\top Q^+ x - \frac{1}{2} c^\top x + \frac{1}{2} u_x^\top Q^+ u_x - \sum_{\substack{j=1 \\ j \neq k}}^{m_x} b_j w_{x_j} \right) && \text{if } b_k > 0, \\
0 &\geq \tau - \frac{1}{2} x^\top Q^+ x - \frac{1}{2} c^\top x + \frac{1}{2} u_x^\top Q^+ u_x - \sum_{\substack{j=1 \\ j \neq k}}^{m_x} b_j w_{x_j} && \text{if } b_k = 0, \\
w_{x_k} &\geq \frac{1}{A_{ki}} \left( (Q^- x + \frac{1}{2} c + Q^+ u_x)_i - \sum_{\substack{j=1 \\ j \neq k}}^{m_x} A_{ji} w_{x_j} \right) && \text{for } i = 1, \dots, m_x, \text{ where } A_{ki} < 0, \\
\frac{1}{A_{kr}} \left( (Q^- x + \frac{1}{2} c + Q^+ u_x)_r - \sum_{\substack{j=1 \\ j \neq k}}^{m_x} A_{jr} w_{x_j} \right) &\geq w_{x_k} && \text{for } r = 1, \dots, m_x, \text{ where } A_{kr} > 0, \\
(Q^- x + \frac{1}{2} c + Q^+ u_x)_s - \sum_{\substack{j=1 \\ j \neq k}}^{m_x} A_{js} w_{x_j} &\geq 0 && \text{for } s = 1, \dots, m_x, \text{ where } A_{ks} = 0.
\end{aligned}$$

After the adjustable variable  $w_{x_k}$  is eliminated, the feasible set becomes:

$$\begin{aligned}
\frac{1}{A_{kr}} \left( (Q^- x + \frac{1}{2} c + Q^+ u_x)_r - \sum_{\substack{j=1 \\ j \neq k}}^{m_x} A_{jr} w_{x_j} \right) &\geq \frac{1}{b_k} \left( \tau - \frac{1}{2} x^\top Q^+ x - \frac{1}{2} c^\top x + \frac{1}{2} u_x^\top Q^+ u_x - \sum_{\substack{j=1 \\ j \neq k}}^{m_x} b_j w_{x_j} \right) && \text{where } b_k > 0 \text{ and } A_{kr} > 0, \\
0 &\geq \tau - \frac{1}{2} x^\top Q^+ x - \frac{1}{2} c^\top x + \frac{1}{2} u_x^\top Q^+ u_x - \sum_{\substack{j=1 \\ j \neq k}}^{m_x} b_j w_{x_j} && \text{where } b_k = 0, \\
\frac{1}{A_{kr}} \left( (Q^- x + \frac{1}{2} c + Q^+ u_x)_r - \sum_{\substack{j=1 \\ j \neq k}}^{m_x} A_{jr} w_{x_j} \right) &\geq \frac{1}{A_{ki}} \left( (Q^- x + \frac{1}{2} c + Q^+ u_x)_i - \sum_{\substack{j=1 \\ j \neq k}}^{m_x} A_{ji} w_{x_j} \right) && \text{where } A_{ki} < 0 \text{ and } A_{kr} > 0, \\
(Q^- x + \frac{1}{2} c + Q^+ u_x)_s - \sum_{\substack{j=1 \\ j \neq k}}^{m_x} A_{js} w_{x_j} &\geq 0 && \text{where } A_{sk}^\top = 0.
\end{aligned}$$

By continuing the process of FME, the adjustable variable  $w_x$  (or some part of it) is eliminated, resulting in a problem with fewer adjustable variables but potentially many more constraints. If the number of constraints in (QO) is limited, then it is computationally efficient to eliminate  $w_x$ .

### 4.3.3 Finite Scenario Approach

One of the approximation approaches for the ARO problem is the Finite Scenario Approach (FSA). In this approach, we restrict ourselves to only finite scenarios of uncertainty set.

Instead of making decision rules, one can introduce a new single optimization variable to each scenario. This approach only assures feasibility for a subset of scenarios from the uncertainty set, so the objective value found with this approach only provides an upper bound to the optimal objective value of the ARO model. The (FSA) is computationally efficient because it only considers a finite number of scenarios, rather than trying to optimize over all possible scenarios. This reduces the number of variables and constraints in the optimization problem, making it easier to solve. By identifying a set of potential scenarios, we are able to utilize this technique that results in a deterministic convex optimization problem as below:

$$\max_{\{u^k\}_k, \{w^k\}_k, \tau} \left\{ \tau \left| \begin{array}{l} \frac{1}{2}(x^k)^\top Q^+ x^k + \frac{1}{2}c^\top x^k + b^\top w^k - \frac{1}{2}(u^k)^\top Q^+ u^k \geq \tau, \quad k = 1, \dots, |\mathcal{W}| \\ -Q^+ u^k + A^\top w^k \leq Q^- x^k + \frac{1}{2}c, \quad k = 1, \dots, |\mathcal{W}| \end{array} \right. \right\}, \quad (\text{FSA-QO})$$

where  $\mathcal{W} = \{x^1, \dots, x^r\}$  is a finite sub-set of  $\mathcal{X}$ .

The following proposition states that if the optimal value of (FSA-QO) for a finite subset of scenarios is identical to the optimal value of (QO), then the optimal solution for (QO) must be included within that subset.

**Proposition 4.3.1.** *Let  $Q = Q^+ + Q^-$  where  $Q \in \mathbb{R}^{n_x \times n_x}$ , and  $Q^+, -Q^- > 0$ . If the optimal value of (FSA-QO) for a given finite subset of scenarios is equal to the optimal value of (QO), then the finite subset of scenarios contains the optimal solution for (QO).*

*Proof.* Let  $(\bar{\tau}, \{\bar{u}^k\}_k, \{\bar{w}^k\}_k)$  be an optimal solution of problem (FSA-QO), and suppose that in this solution, the following constraint of problem (FSA-QO) is binding for some  $x^s \in \mathcal{W} \not\subseteq \mathcal{X}$ :

$$\frac{1}{2}(x^s)^\top Q^+ x^s + \frac{1}{2}c^\top x^s + b^\top \bar{w}^s - \frac{1}{2}(\bar{u}^s)^\top Q^+ \bar{u}^s = \bar{\tau}.$$

We claim that  $x^s$  is an optimal solution of (QO). To show this, we have

$$\begin{aligned} \bar{\tau} &= \frac{1}{2}(x^s)^\top Q^+ x^s + \frac{1}{2}c^\top x^s + \max_{w, u} \left\{ b^\top w - \frac{1}{2}u^\top Q^+ u : -Q^+ u + A^\top w \leq Q^- x^s + \frac{1}{2}c \right\} \\ &= \frac{1}{2}(x^s)^\top Q^+ x^s + \frac{1}{2}c^\top x^s + \min_z \left\{ \frac{1}{2}z^\top Q^+ z + \frac{1}{2}c^\top z + z^\top Q^- x^s \mid z \in \mathcal{X} \right\}, \end{aligned} \quad (4.21)$$

The validity of the last equality follows from the fact that strong duality holds. If we denote

the optimal solution of the above minimization problem by  $z^s$ , then we can conclude that  $(x^s, z^s)$  is optimal for (Bi-QO). Furthermore, according to Corollary 4.2.1, we have  $x^s = z^s$ , and therefore  $x^s$  is optimal for the original (QO) problem.  $\square$

## 4.4 Solution Method

In the previous section, we explained how to obtain a lower bound using the techniques from ARO literature. In this section, we provide an algorithm to obtain a feasible solution of construct an upper bound.

After solving the approximated problem (4.13), we use the obtained solution to extract worst-case scenarios from each constraint of the robust counterpart problem (4.13). Among these scenarios, we select the one that yields the best objective value for the original QO problem. After identifying the most favorable scenario, our attention is redirected to the bi-convex reformulation of the QO problem. Given the selected scenario, we employ the mounting claiming algorithm for (Bi-QO) to improve the quality of the solution. This process ultimately leads us to an upper bound for (QO) problem.

### Mountain Climbing Procedure

**Input:** Matrix  $Q$ , vector  $c$ , and starting point  $x^0$ .

**Initialization:** Decompose  $Q = Q^+ + Q^-$  such that  $Q^+, -Q^- \geq 0$ .

**Repeat:** Execute the following steps:

$$x^{(k+1)} \leftarrow \operatorname{argmin}_{x \in \mathbb{R}^{n_x}} \left\{ \frac{1}{2} x^\top Q^+ x + x^\top Q^- x^{(k)} + \frac{1}{2} c^\top x : x \in \mathcal{X} \right\}.$$

**Until:** No further improvement is possible.

**Output:** Solution candidate  $x^{(end)}$ .

By employing the ARO reformulation, bi-convex reformulation, and the mounting claiming method, we can efficiently explore and improve the solution space, thereby obtaining an upper bound that closely approaches the optimal value. This approach allows us to make significant progress in refining the solution quality while mitigating computational challenges

often associated with large-scale optimization problems.

We present an ARO-based approach designed to find a good candidate solution for the QO problem. The “ARO-QO Algorithm” provides the pseudo-code for executing this procedure.

#### ARO-Based Algorithm to Solve QO (ARO-QO Algorithm)

**Input:** Matrix  $Q$ , vector  $c$ , matrix  $A$ , and vector  $b$ .

**Initialization:** Decompose  $Q = Q^+ + Q^-$  such that  $Q^+, -Q^- \geq 0$ .

**(Step 1) Lower Bound:** Compute the lower bound for the approximated problem based on the ARO formulation of the QO and hybrid decision rule (see section 4.3).

**(Step 2) Generation of Worst-Case Scenarios:** Generate a finite set of worst-case scenarios by substituting the optimal decision rule into (4.13).

**(Step 3) Set Initial Point:** Select from these scenarios the one that yields the best objective value for the original QO problem. Denote this point by  $x^{(0)}$ .

**(Step 4) Improve the Initial Solution:** Execute the mountain climbing algorithm starting with the initial solution  $x^{(0)}$ :

$$x^{(k+1)} \leftarrow \operatorname{argmin} \left\{ \frac{1}{2} x^\top Q^+ x + x^\top Q^- x^{(k)} + \frac{1}{2} c^\top x : x \in \mathcal{X} \right\}.$$

**(Step 5) Termination:** Continue (Step 4) until no further improvement is observed.

**Output:** Final solution candidate  $x^* := x^{(end)}$ , and the corresponding upper-bound value  $UB := (x^*)^\top Q x^* + c^\top x^*$ .

## 4.5 Numerical Experiments

In this section, we conduct a comprehensive numerical experiment to evaluate the efficacy of ARO-QO Algorithm. The efficiency of a particular bound on the optimal value of a mathematical optimization problem is influenced by two key aspects: the precision of the generated bound and the required computational time.

We implement the numerical experiments using MATLAB 2022a. The computations are executed on a laptop equipped with an Intel(R) Core(TM) i5-3210M CPU at 2.50 GHz and 8 GB of RAM. We use YALMIP to pass optimization problems to suitable solvers [84].

We emphasize that the computational times reported in our experiments exclude the time required by YALMIP to build the model and pass it to solvers, and we merely consider the time consumed by the solvers themselves. In what follows, we present the numerical experiments, specifically focusing on concave quadratic minimization and standard quadratic optimization. All the instances and the code are available at: [\[Link\]](#).

We use state-of-the-art global solvers to solve the QO problems, namely Gurobi [65, version 10.0] and CPLEX [72, version 12.9]. Given that our bounds requires solving multiple linear optimization problems, we specifically employ Gurobi for this purpose. Moreover, MOSEK [91, version 10.1.15] is used to solve second-order cone optimization problems.

#### 4.5.1 Concave Quadratic Minimization

Let us consider a concave quadratic minimization over a polyhedron

$$\begin{aligned} \min_{x \geq 0} \quad & x^\top Qx + c^\top x \\ \text{s.t.} \quad & Ax \geq b, \end{aligned} \tag{4.22}$$

where  $Q \in \mathbb{R}^{n_x \times n_x}$ , and  $-Q \geq 0$ ,  $c \in \mathbb{R}^{n_x}$ ,  $A \in \mathbb{R}^{m_x \times n_x}$ , and  $b \in \mathbb{R}^{m_x}$  are given. From Corollary 4.2.2 and Table 4.8, we have the following linear ARO reformulation of (4.22):

$$\begin{aligned} \max_{\tau \in \mathbb{R}} \quad & \tau \\ \text{s.t.} \quad & \forall x \in \mathcal{X}, \exists w_x : \begin{cases} \frac{1}{2}c^\top x + b^\top w_x \geq \tau, \\ A^\top w_x \leq Qx + \frac{1}{2}c, \\ w_x \geq 0, \end{cases} \end{aligned} \tag{4.23}$$



where  $\mathcal{X} := \{x \in \mathbb{R}^{n_x} | Ax \geq b, x \geq 0\}$ . To obtain a lower bound, we consider the following decision rule:

$$w_x := \begin{pmatrix} z + Zx \\ w \end{pmatrix},$$

where for given  $r \in \{1, 2, \dots, m_x\}$ ,  $z \in \mathbb{R}^r$ ,  $Z \in \mathbb{R}^{r \times n_x}$ , and  $w \in \mathbb{R}^{(m_x - r)}$  are static variables. It is important to note that for  $r = m_x$ , we obtain a full affine decision rule, while for  $r = 0$ , we have a static decision rule. For other values, we have a partial affine decision rule. Each of these decision rule types has its own set of advantages and disadvantages, which we address later in this section.

In [105], the authors propose an approximation solution approach for solving a concave minimization problem via ARO by providing upper and lower bounds, where the lower bound is formulated as a second-order cone optimization problem. To solve this, we employed the MOSEK solver.

In this section, we compare the quality of the solution obtained by ARO-QO Algorithm with Gurobi and CPLEX, and [105] method. In all of our numerical experiments, we set a maximum time limit of 3,000 seconds.

We analyze the performance of the upper and lower bound in terms of the optimality gap, which is measured as follows:

$$\text{Gap}(\%) = \left( \frac{\text{UB} - \text{LB}}{|\text{UB}| + 10^{-4}} \right) \times 100,$$

where ‘LB’ is the lower bound and ‘UB’ is the upper bound for a given instance. The addition of the small constant  $10^{-4}$  in the denominator is to ensure the prevention of division by zero.

### Problem Instances

First, we consider the seven test instances from Section 4.3 of [105]. We undertake a detailed comparison of three versions of ARO-QO Algorithm (static, partial, and fully affine), [105] method, and global solvers Gurobi and CPLEX. In the lower bound approximation of the ARO-QO Algorithm, applying full static, partial affine (restricting the first  $r = \lfloor \frac{m_x}{7} \rfloor + 1$  of

$w_x$  to be affine and the remaining  $m_x - r$  to be constant), and full affine decision rules has distinct effects on the optimality gaps. Table 4.1 illustrates that increasing the number of affine decision rules correlates with tighter optimality gaps within the ARO-QO Algorithm. Particularly, the Affine ARO-QO Algorithm consistently achieves the smallest optimality gaps among its variants. However, this precision incurs longer solver time, notably in larger problems, such as Problem 5, which required 900.48 seconds, and Problems 6 and 7, where it exceeded the time limit. However, the Static and Partial ARO-QO Algorithms have the lowest optimality gap in Problem 7 and do so within a reasonable time.

In addition, it is noteworthy that the [105] method typically leads to larger optimality gaps compared to the ARO-QO algorithms, while the solver times for this method are longer than static ARO-QO Algorithms. Gurobi and CPLEX achieve optimality for Problems 1-6. However, CPLEX often requires more time, especially in larger problems. Both Gurobi and CPLEX reached their time limits on Problem 7, highlighting the difficulty in solving large-size problems.

**Table 4.1:** Optimality gaps and solver times of concave minimization instances from [105].

Problem	Static ARO-QO Algorithm		Partial ARO-QO Algorithm		Affine ARO-QO Algorithm		Selvi et al. method [105]		Gurobi		CPLEX	
	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time
#1 ( $m_x = 10, n_x = 20$ )	30.38	0.08	27.97	0.15	0.02	1.05	77.85	0.17	0.00	0.11	0.00	0.12
#2 ( $m_x = 10, n_x = 20$ )	9.13	0.08	8.94	0.15	0.01	1.06	34.77	0.15	0.00	0.12	0.00	0.11
#3 ( $m_x = 15, n_x = 10$ )	0.42	0.04	0.29	0.06	0.00	0.12	1.68	0.15	0.00	0.03	0.00	0.06
#4 ( $m_x = 62, n_x = 50$ )	0.12	0.27	0.09	0.48	0.00	11.46	1.10	1.32	0.00	0.56	0.00	1.56
#5 ( $m_x = 130, n_x = 100$ )	0.08	1.28	0.07	4.81	0.00	900.48	2.15	8.27	0.00	12.03	0.00	43.64
#6 ( $m_x = 240, n_x = 200$ )	0.02	8.98	0.02	44.83	-	3000*	1.52	59.60	0.00	528.99	0.00	1165.38
#7 ( $m_x = 280, n_x = 240$ )	0.04	30.47	0.04	104.90	-	3000*	5.71	80.90	15.22	3000*	14.17	3000*

Notes. In this table, the first column presents the problem numbers along with their corresponding dimensions. The symbol “-” indicates that it was not possible to determine bound within 3,000 seconds.

Even though the static policy yields the highest optimality gap among the three decision rules, it stands out for its minimal computation time required to derive both lower and upper bounds. This becomes the problem after applying a static decision rule that has no extra variables and constraints, and the upper bound is calculated independently of the solution of the lower bound. Notably, the computation of the upper bound in the static policy is independent of the lower bound solution. This independence is based on the structure of problem (4.23). In Step 2 of the ARO-QO Algorithm, where worst-case scenarios are generated from problem (4.23), an optimal decision rule is not required. Remarkably, in the seven instances, the calculated upper bound aligns with the global optimal value, and is obtained quickly, as reported in Table 4.7. When compared with alternative approaches, such

as applying partial or full affine decision rules or using the [105] method (which also achieves optimal upper bounds), the full static decision rule demonstrates a faster computation process to reach a candidate solution. This increased speed is attributed to the fact that mentioned methods, necessitate optimal solutions for the lower bound to determine the upper bound, which inherently increases their computational demand as opposed to the more streamlined process observed in the static policy.

After considering the seven test instances of [105], we randomly generate large-size instances. For a meaningful comparison of the mentioned approaches, we evaluate the quality of the bounds on the objective value of problem (4.22) using 15 groups of random instances, with the dimension  $n_x$  taking value in  $\{50, 100, \dots, 600, 700\}$  and the number of constraints  $m_x$  spanning a range in  $\{100, 150, \dots, 750, 800\}$ . Each group contains five instances of the same size, and these instances are generated in a similar manner to those created in [105].

For each group, ranging from #1 to #15, Table 4.2 lists the mean optimality gap and solver time, with standard deviations included in brackets (details can be found in Table 4.10).

Based on the above discussions we use a static decision rule for the lower bound approximation in ARO-QO Algorithm. We observe that our ARO-QO Algorithm maintains a consistent performance level across different problem complexities. CPLEX demonstrates reasonable performance up to Group 6. Furthermore, the method by [105] displays more consistent optimality gaps across all problem groups, despite them being significantly higher compared to those of Gurobi and CPLEX in the initial groups. Gurobi achieved very low optimality gaps in Groups 1-3, in particular, showing an increasing trend in solver times with higher problem groups, often reaching the 3,000-second limit. Gurobi for the instances in groups 14 and 15, and CPLEX for the instances in groups 8-15, it was not possible to determine the presence of a feasible solution within the time limit.

From a computation time perspective, in each group, the ARO-QO Algorithm demonstrates the lowest time to reach the bounds compared to other methods. Overall, the ARO-QO Algorithm showcases efficiency in computation time and maintains acceptable gaps in all groups. This underscores the ARO-QO Algorithm's proficiency in balancing time efficiency and gap management across these problems.

**Table 4.2:** Statistic of optimality gaps and solver times for randomly generated concave minimization instances.

Group	ARO-QO Algorithm		Selvi et al. method [105]		Gurobi		CPLEX	
	Gap	Time	Gap	Time	Gap	Time	Gap	Time
#1	1.54 [0.35]	0.54 [0.01]	13.32 [0.94]	2.20 [0.28]	<b>0.01</b> [0.00]	66.53 [44.43]	<b>0.01</b> [0.00]	16.93 [2.14]
#2	1.49 [0.35]	2.77 [0.17]	13.81 [0.88]	9.00 [0.40]	<b>0.01</b> [0.00]	776.08 [407.71]	0.02 [0.01]	382.64 [181.74]
#3	1.71 [0.36]	4.50 [0.13]	16.37 [0.97]	17.83 [1.46]	<b>0.02</b> [0.02]	1602.76 [1035.26]	0.12 [0.15]	1285.16 [1256.08]
#4	1.40 [0.37]	15.31 [1.21]	14.71 [2.03]	51.25 [4.76]	10.69 [10.33]	3000*	<b>0.22</b> [0.35]	2927.72 [161.62]
#5	1.33 [0.13]	24.66 [0.92]	17.02 [1.03]	101.50 [22.87]	25.93 [3.34]	3000*	<b>0.43</b> [6.26]	3000*
#6	<b>1.62</b> [0.19]	50.28 [9.55]	16.38 [1.18]	154.01 [10.49]	23.85 [3.49]	3000*	1365.22 [1636.18]	3000*
#7	<b>1.53</b> [0.28]	76.88 [6.19]	18.83 [2.21]	256.42 [30.29]	37.02 [2.57]	3000*	4104.90 [236.04]	3000*
#8	<b>1.70</b> [0.25]	95.42 [21.84]	16.21 [1.83]	295.57 [12.37]	118.90 [88.99]	3000*	-	3000*
#9	<b>1.89</b> [0.20]	164.98 [7.30]	19.38 [0.77]	556.61 [19.42]	171.95 [103.12]	3000*	-	3000*
#10	<b>1.79</b> [0.08]	143.81 [11.21]	16.92 [0.56]	256.74 [15.37]	2521.58 [5213.52]	3000*	-	3000*
#11	<b>1.64</b> [0.19]	362.73 [23.07]	19.24 [0.66]	504.47 [15.83]	5420.98 [4697.11]	3000*	-	3000*
#12	<b>1.57</b> [0.12]	263.61 [35.30]	17.17 [0.89]	424.63 [43.45]	8616.82 [5403.16]	3000*	-	3000*
#13	<b>1.48</b> [0.29]	617.78 [48.84]	18.89 [1.50]	821.51 [75.41]	10233.75 [1598.93]	3000*	-	3000*
#14	<b>1.79</b> [0.39]	343.31 [12.94]	17.33 [1.08]	503.12 [36.98]	-	3000*	-	3000*
#15	<b>1.48</b> [0.21]	906.65 [80.47]	19.21 [1.47]	1164.43 [515.86]	-	3000*	-	3000*

Notes: This table categorizes problems into groups in the first column. The subsequent columns display “mean [standard deviation]” values of the optimality gaps and solver time for each sub-group. The symbol “-” indicates that it was not possible to determine upper bounds for all instances of the corresponding group within the maximum the time limit.

#### 4.5.2 Standard Quadratic Optimization

Let us consider a standard quadratic optimization problem

$$\min_{x \in \Delta} x^\top \tilde{Q}x + c^\top x.$$

In general, a standard QO problem is NP-hard [34]. We remark that the quadratic function  $x^\top \tilde{Q}x + c^\top x$  over the unit-simplex can be described as a homogeneous quadratic function:  $x^\top Qx$ , where  $Q := \tilde{Q} + \frac{1}{2}ec^\top + \frac{1}{2}ce^\top$ . Hence, without loss of generality, the standard QO problem can be represented as follows:

$$\min_{x \in \Delta} x^\top Qx. \tag{StQO}$$

Let  $Q \in \mathbb{R}^{n_x \times n_x}$  be an indefinite symmetric matrix. The (StQO) problem is equivalent to the

following problem

$$\begin{aligned} & \max_{\tau \in \mathbb{R}} \tau \\ & \text{s.t. } \forall x \in \Delta, \exists (u_x \in \mathbb{R}^{n_x}, w_x \in \mathbb{R}) : \begin{cases} \frac{1}{2}x^\top Q^+ x - \frac{1}{2}u_x^\top Q^+ u_x + w_x \geq \tau, \\ -Q^+ u_x + e w_x \leq Q^- x, \end{cases} \end{aligned} \quad (\text{ARO-StQO})$$

where  $\tau$  is the static variable,  $x \in \Delta$  is the uncertain parameter, and  $(u_x, w_x) \in \mathbb{R}^{n_x} \times \mathbb{R}$  is the adjustable variable.

As mentioned in previous sections, we address two types of approximations of (ARO-StQO). First, the following problem is an approximation of (ARO-StQO) by applying the hybrid static and affine decision rule

$$\max_{z, u, z_0, \tau} \left\{ \tau \left| \begin{array}{ll} \frac{1}{2}x^\top Q^+ x + (z_0 + z^\top x) - \frac{1}{2}u^\top Q^+ u \geq \tau, & \forall x \in \Delta \\ -Q^+ u + e(z_0 + z^\top x) \leq Q^- x, & \forall x \in \Delta \end{array} \right. \right\}, \quad (\text{L1-StQO})$$

which is equivalent to the following deterministic convex quadratic optimization problem

$$\max_{z, u, z_0, \tau, \alpha, \beta, \theta} \left\{ \tau \left| \begin{array}{l} -\frac{1}{2}\alpha^\top Q^+ \alpha + \beta + z_0 - \frac{1}{2}u^\top Q^+ u \geq \tau, \\ e\beta - Q^+ \alpha \leq z, \\ Q^+ u + \theta - e z_0 \geq 0, \\ e\theta^\top \leq (-e z^\top + Q^-)^\top, \end{array} \right. \right\}. \quad (4.24)$$

Second, using Fourier–Motzkin elimination on (ARO-StQO) to eliminate  $w_x \in \mathbb{R}$ , we have

$$\begin{aligned} & \max_{\tau \in \mathbb{R}} \tau \\ & \text{s.t. } \forall x \in \Delta, \exists u_x : \frac{1}{2}x^\top Q^+ x - \frac{1}{2}u_x^\top Q^+ u_x + (Q^-)_i x + (Q^+)_i u_x \geq \tau, \quad i = 1, \dots, n_x. \end{aligned} \quad (\text{FME-StQO})$$

In (FME-StQO), applying a constant decision rule on  $u_x$  (i.e.,  $u_x = u$ ) result in

$$\begin{aligned} & \max_{\tau \in \mathbb{R}, u \in \mathbb{R}^{n_x}} \tau \\ & \text{s.t. } \frac{1}{2}x^\top Q^+ x - \frac{1}{2}u^\top Q^+ u + (Q^-)_i x + (Q^+)_i u \geq \tau, \quad \forall x \in \Delta \quad i = 1, \dots, n_x \end{aligned} \quad (\text{L2-StQO})$$

The lower bound obtained from problem (L2-StQO) is better than the one via (L1-StQO). It is imperative to note, however, that the computational effort associated with this superior bound may be elevated due to an augmented set of constraints.

We now offer a more detailed examination of the ARO-QO Algorithm employed for solving StQOs. Utilizing decision rules, we have successfully approximated the original problem. The optimal values extracted from each of these approximated problems serve as lower bounds. Subsequently, we discuss selected worst-case scenarios, which are derived based on the optimal solutions of these lower-bound problems.

**Scenario Based on L1-StQO.** Let  $(z^*, v^*, z_0^*, \tau^*, \alpha^*, \beta^*, \theta^*)$  be an optimal solution for (4.24) which is the deterministic reformulation of (L1-StQO). We select scenarios using the following optimization problems

$$\begin{cases} \bar{x}^0 \in \operatorname{argmin}_{x \in \Delta} \left\{ \frac{1}{2}x^\top Q^+ x + z^{*\top} x \right\}, \\ \bar{x}^i \in \operatorname{argmin}_{x \in \Delta} \left\{ (-e z^{*\top} + Q^-)_i x \right\}, \quad i = 1, \dots, n_x. \end{cases} \quad (4.25)$$

Note that, we do not need to solve linear optimization problems in (4.25) to find  $\{\bar{x}^i\}_{i=1}^{n_x}$  and just need to consider the extreme points of the unit-simplex set, i.e.,  $\{e_i\}_{i=1}^{n_x}$ . These points provide the natural upper bound (i.e.,  $e_i^\top Q e_i = Q_{ii}$ ), which exists in the literature, see [62, Lemma 2.1 part (iv)]. We choose the best scenario, denote it by  $x^{*1}$ , as the one with the lowest objective value, i.e.,

$$x^{*1} \in \operatorname{argmin}_x \left\{ x^\top Q x \mid x \in \{\bar{x}^i\}_{i=0}^{n_x} \right\}.$$

**Scenario Based on L2-StQO.** We can find scenarios from the uncertainty set  $\Delta$  according

to (L2-StQO) as follows

$$\hat{x}^i \in \operatorname{argmin}_{x \in \Delta} \left\{ \frac{1}{2} x^\top Q^+ x + Q_i^- x \right\}, \quad i = 1, \dots, n_x. \quad (4.26)$$

We denote by  $x^{*2}$  the scenario with the lowest objective value, i.e.,

$$x^{*2} \in \operatorname{argmin}_x \left\{ x^\top Q x \mid x \in \{\hat{x}^i\}_{i=1}^{n_x} \right\}.$$

In this subsection, our method will be compared with the global solvers Gurobi and CPLEX, as well as with the local solver IPOPT. It is worth noting that IPOPT, a local primal-dual-based interior point solver [115] is renowned for its time computational efficiency but functions exclusively as a local solver. To solve convex quadratic sub-problems in ARO-QO Algorithm, we employed MOSEK.

To implement our ARO-based method on StQOs, we need to compute lower bounds on the optimal objective value of StQO as discussed above. We consider two best scenarios,  $x^{*1}$  and  $x^{*2}$ , obtained from the lower bound approximations (L1-StQO) and (L2-StQO). By using these two initial points, we can improve the initial solutions, and the best solution obtained becomes the candidate solution, with its corresponding objective value serving as an upper bound.

### Problem Instances

It is of paramount importance to note that with a high probability, global solutions of a randomly generated StQO instances are located either at vertices or edges of the standard simplex [36]. In order to make a fair comparison based on this information, we do not generate naive random instances in our study, as our upper-bound methodology would be optimal in these cases. Instead, we concentrate on using instances from well-known datasets or employing their patterns to generate new instances, as outlined by [37, 83], and [104].

We analyze the performance of the upper bound in terms of the solution gap, which is

measured as follows:

$$\text{SGap}(\%) = \left( \frac{\text{UB} - \text{UB}^{(best)}}{|\text{UB}| + 10^{-4}} \right) \times 100,$$

where  $\text{UB}^{(best)}$  represents the best upper bound obtained from all approaches, and  $\text{UB}$  is the upper bound for a given instance.

### Detailed Results

As mentioned in Remark 4.2.1, multiple representations of  $Q$  can be obtained based on  $Q^+$  and  $Q^-$ . Here we utilize Representation 2 and apply the ARO-QO Algorithm to find upper bounds. Furthermore, we consider the upper bounds obtained from global solvers, when setting their time limit to the time taken by ARO-QO Algorithm. We use SGap1 and Time1 to refer this upper bound. We also set the time limit at the solvers to 3,000 seconds to which we refer by SGap 2 and Time2.

A statistical analysis of solution gaps for two classes of test problems is presented in Table 4.3 and 4.4. Both classes consist of 150 instances, with a dimension of  $n_x = 30$  for Class One and  $n_x = 50$  for Class Two.

**Table 4.3:** Statistic of solution gaps and solution times of (StQO) instances in Class One.

Class One	ARO-QO Algorithm		Gurobi				CPLEX				IPOPT	
	SGap	Time	SGap1	Time1	SGap2	Time2	SGap1	Time1	SGap2	Time2	SGap	Time
Mean	3.11	0.70	3.89	0.65	0.00	22.28	0.90	0.73	0.00	226.46	11.09	0.04
Standard deviation	6.12	0.60	10.22	0.45	0.00	111.73	1.94	0.61	0.04	714.40	20.78	0.02

Notes. Test instances from [37].

**Table 4.4:** Statistic of solution gaps and solution times of (StQO) instances in Class Two.

Class Two	ARO-QO Algorithm		Gurobi				CPLEX				IPOPT	
	SGap	Time	SGap1	Time1	SGap2	Time2	SGap1	Time1	SGap2	Time2	SGap	Time
Mean	3.33	1.08	8.56	1.07	0.05	233.36	1.88	1.11	0.02	721.48	14.03	0.09
Standard deviation	6.36	0.96	15.34	0.97	0.32	743.15	4.14	0.98	0.10	1258.62	21.98	0.05

Notes. Test instances from [37].

In Tables 4.3 and 4.4, we observe that within the time taken by ARO-QO Algorithm, CPLEX exhibits the best performance, and our approach better than Gurobi. As the time limit extends to a maximum of 3,000 seconds for global solvers Gurobi and CPLEX, they demonstrate superior performance, particularly in these two cases of small-sized instances,



achieving the best solutions. While IPOPT shows the fastest solution times, it presents significantly higher mean solution gap values in both classes. This situation reflects a trade-off between speed and accuracy. Currently, it is apparent that global solvers are generally effective in handling small-sized problems.

The next step involves comparing instances of larger sizes. We consider 12 groups of instances with the dimension  $n_x$  taking values in  $\{100, 300, 500, 700\}$ . Since the density of the matrices may also affect the performance of the considered solution methods, we examine three density values for each dimension - 50%, 75%, and 90% - for the matrix  $Q$  in the objective function. For these test problems, some are sourced from [83], while others are generated using the pattern described in [83] and [104].

In Table 4.5, the average solution gaps for each instance group are provided (the details can be found in Table 4.11).

**Table 4.5:** Statistic of solution gaps and solution times of generated large-sized (StQO) instances.

Group	ARO-QO Algorithm		Gurobi		CPLEX		IPOPT	
	SGap(%)	Time	SGap1	SGap2	SGap1	SGap2	SGap	Time
#1	2.84 [4.36]	2.53 [0.94]	4.50 [5.17]	<b>0.49</b> [0.60]	584.40 [92.30]	1.80 [2.13]	3.09 [2.73]	0.17 [0.02]
#2	3.46 [1.31]	3.24 [1.27]	4.27 [2.35]	4.04 [2.70]	582.33 [99.73]	<b>0.00</b> [0.00]	3.46 [2.77]	0.21 [0.04]
#3	<b>0.64</b> [1.43]	3.11 [0.98]	2.41 [3.34]	1.43 [1.96]	589.80 [119.47]	1.08 [2.40]	1.74 [2.49]	0.30 [0.09]
#4	3.33 [3.71]	12.83 [5.73]	-	5.37 [3.62]	644.54 [81.70]	<b>1.81</b> [3.50]	7.57 [4.71]	2.91 [0.64]
#5	<b>1.56</b> [2.56]	11.37 [3.59]	-	3.53 [2.60]	666.54 [80.77]	2.75 [2.52]	1.95 [1.99]	3.90 [1.58]
#6	0.48 [0.61]	8.22 [1.73]	-	2.34 [2.97]	676.43 [82.21]	1.27 [2.50]	<b>0.35</b> [0.55]	12.28 [3.92]
#7	<b>0.06</b> [0.14]	34.01 [19.16]	-	2.06 [1.61]	612.43 [65.44]	612.43 [65.44]	4.47 [3.78]	31.79 [16.97]
#8	1.08 [2.11]	17.78 [1.95]	-	<b>0.86</b> [1.91]	652.02 [71.03]	652.02 [71.03]	3.22 [2.79]	49.59 [17.47]
#9	<b>1.06</b> [0.68]	18.14 [3.42]	-	1.92 [2.21]	661.66 [74.03]	661.66 [74.03]	1.44 [1.38]	277.53 [165.23]
#10	2.43 [2.28]	47.26 [10.11]	-	<b>1.40</b> [2.08]	624.72 [152.55]	624.72 [152.55]	4.88 [1.89]	127.84 [17.81]
#11	2.35 [2.85]	29.85 [1.43]	-	<b>1.21</b> [1.69]	657.54 [157.00]	657.54 [157.00]	3.87 [2.21]	412.14 [149.47]
#12	<b>0.38</b> [0.59]	31.76 [2.75]	-	4.14 [1.82]	657.22 [162.33]	657.22 [162.33]	0.48 [0.98]	113.62 [19.52]
All Problems	<b>1.64</b> [2.36]	18.34 [15.19]	-	2.40 [2.52]	631.16 [105.36]	322.86 [333.58]	3.04 [3.07]	86.02 [140.14]

Notes. This table categorizes problems into groups in the first column. The subsequent columns display ‘mean [standard deviation]’ values of the solution gaps for each sub-group. SGap1 represents the solution gap with the time limit equals to that needed by the ARO-QO Algorithm, while SGap2 denotes the solution gap within a 3,000-second time frame. Instances, where Gurobi failed to find solutions within its allotted time, are marked with a “-”, indicating its inability to establish feasible solutions (upper bounds) for all instances in the respective group.

Table 4.5 presents a detailed evaluation of the solution gap (SGap) percentages obtained by various algorithms in multiple large-sized instances. The table outlines the mean and standard deviation of the SGap for each algorithm, providing a clear perspective on their average effectiveness.

The ARO-QO Algorithm demonstrates consistently low solution gap percentages in various large-sized groups, especially notable in groups 3, 5, 7, and 12. Within a 3,000-second span, Gurobi shows commendable performance in groups 1, 8, 10, and 11, while CPLEX excels in groups 2 and 4. The local solver IPOPT in class 6 has good performance. However, CPLEX exhibits less satisfactory performance in groups 7 to 12. Moreover, across all groups, the ARO-QO Algorithm significantly surpasses the global solvers in terms of efficiency, considering its shorter time requirement. A minimal SGap is indicative of the algorithm's proficiency in approximating solutions that are closer to the optimal or best-known solutions, an essential objective in optimization problems. Conclusively, the ARO-QO Algorithm stands out for having the lowest mean gap percentage across all evaluated problems, highlighting its superior performance.

Even though our lower bound were loose for small instances based on Table 4.11, it is evident that these bounds surpass the lower bounds in groups 10, 11, and 12, thereby outperforming the global solvers CPLEX and Gurobi overall.

## 4.6 Conclusions

We introduce a novel reformulation technique that enables the Quadratic Optimization problem (QO) to be recast as an Adjustable Robust Optimization problem (ARO). This process begins by demonstrating that any QO problem can be transformed into a disjoint bi-convex QO problem. Following this, we propose an equivalent ARO reformulation. Specifically, we illustrate that employing a so-called decision rule technique to approximate the ARO reformulation equates to using a linearization-relaxation technique on its bi-convex form. The ARO reformulation offers a new approach to solving non-convex QO problems by transferring the complexity from the original problem to its equivalent ARO counterpart. Specifically, in the concave QO problem, our ARO model transforms into a linear ARO, whereas in the indefinite QO problem, it becomes a non-linear ARO. Moreover, we develop an algorithm capable of identifying near-optimal solutions using our novel reformulations. We demonstrate the effectiveness of our ARO-based method in solving a class of quadratic optimization problems through numerical experiments, showing that it can yield high-quality

solutions with reasonable computational costs. This established connection between QO and ARO provides a new perspective on addressing the challenges of non-convex QO problems and opens up new possibilities for further research in the field of mathematical optimization.

**Table 4.6:** Comparison result on concave quadratic minimization test instances from [105].

Problem	Static ARO-QO Algorithm			Partial affine ARO-QO Algorithm			Full affine ARO-QO Algorithm			Selvi et al. method [105]			Gurobi			CPLEX		
	LB	UB	Gap	LB	UB	Gap	LB	UB	Gap	LB	UB	Gap	LB	UB	Gap	LB	UB	Gap
#1 ( $m_x = 10, n_x = 20$ )	-514.68	-394.75	30.38	-505.16	-394.75	27.97	-394.83	-394.75	0.02	-702.05	-394.75	77.85	-394.75	-394.75	0.00	-394.75	-394.75	0.00
#2 ( $m_x = 10, n_x = 20$ )	-965.52	-884.75	9.13	-963.87	-884.75	8.94	-884.83	-884.75	0.01	-1192.05	-884.75	34.73	-884.75	-884.75	0.00	-884.75	-884.75	0.00
#3 ( $m_x = 15, n_x = 10$ )	-4694.10	-4674.68	0.42	-4688.44	-4674.68	0.29	-4674.68	-4674.68	0.00	-4753.10	-4674.68	1.68	-4674.83	-4674.68	0.00	-4674.92	-4674.68	0.00
#4 ( $m_x = 62, n_x = 50$ )	-175920.41	-175705.59	0.12	-175869.24	-175705.59	0.09	-175705.59	-175705.59	0.00	-177638.35	-175705.59	1.10	-175707.22	-175705.59	0.00	-175707.22	-175705.59	0.00
#5 ( $m_x = 130, n_x = 100$ )	-693146.47	-692613.05	0.08	-693068.49	-692613.05	0.07	-692613.05	-692613.05	0.00	-707519.84	-692613.05	2.15	-692633.48	-692613.05	0.00	-692633.48	-692613.05	0.00
#6 ( $m_x = 240, n_x = 200$ )	-6022194.41	-6020787.42	0.02	-6021978.92	-6020787.42	0.02	NA	-	-	-6112433.52	-6020787.42	1.52	-6020887.35	-6020787.42	0.00	-6020887.35	-6020787.42	0.00
#7 ( $m_x = 280, n_x = 240$ )	-1856557.05	-1855739.98	0.04	-1856443.10	-1855733.06	0.04	NA	-	-	-1961723.39	-1855733.06	5.71	-1900962.70	-1649853.80	15.22	-2162137.04	-1855739.98	14.17

Notes. In this table, the first column presents the problem numbers along with their corresponding dimensions. For each problem, we applied the ARO-QO Algorithm: one with a full static decision rule, another with a partial affine decision rule, and the last one with a full affine decision rule on adjustable variables. In each approach, the ‘LB’ column represents the lower bound values, the ‘UB’ column displays the upper bounds.

**Table 4.7:** Time Results for concave quadratic minimization test instances from [105].

Problem	Static ARO-QO			Partial affine ARO-QO			Full affine ARO-QO			Selvi et al. method [105]			Global Solver	
	LB	UB	Time	LB	UB	Time	LB	UB	Time	LB	UB	Time	Gurobi	CPLEX
#1 ( $m_x = 10, n_x = 20$ )	0.02	0.06	0.08	0.07	0.08	0.15	0.94	0.11	1.05	0.07	0.10	0.17	0.11	0.12
#2 ( $m_x = 10, n_x = 20$ )	0.02	0.06	0.08	0.07	0.08	0.15	0.95	0.11	1.06	0.05	0.10	0.15	0.12	0.11
#3 ( $m_x = 15, n_x = 10$ )	0.01	0.03	0.04	0.01	0.05	0.06	0.04	0.08	0.12	0.06	0.09	0.15	0.03	0.06
#4 ( $m_x = 62, n_x = 50$ )	0.11	0.16	0.27	0.26	0.22	0.48	11.06	0.40	11.46	0.89	0.43	1.32	0.56	1.56
#5 ( $m_x = 130, n_x = 100$ )	0.88	0.40	1.28	4.25	0.56	4.81	899.42	1.06	900.48	7.07	1.20	8.27	12.03	43.64
#6 ( $m_x = 240, n_x = 200$ )	7.49	1.49	8.98	43.15	1.68	44.83	3000*	-	3000*	56.08	3.82	59.60	528.99	1165.38
#7 ( $m_x = 280, n_x = 240$ )	28.44	2.04	30.47	102.52	2.38	104.90	3000*	-	3000*	76.44	4.46	80.90	3000*	3000*

Notes. In this table, the ‘LB’ column represents the lower bound time, the ‘UB’ column displays the upper bound time, and the ‘Time’ column indicates the total corresponding solver times. Additionally, the columns for global solvers also report the time taken by Gurobi and CPLEX solvers to reach bounds.

**Table 4.8:** Other classes of ARO reformulations of indefinite QO problem.

Feasible Region	Type	ARO Problem
	I	
$\mathcal{X} = \{x \in \mathbb{R}^{n_x} \mid Ax = b, x \geq 0\}$	$\max_{\tau \in \mathbb{R}} \tau$	s.t. $\forall x \in \mathcal{X}, \exists(u_x, w_x) : \begin{cases} \frac{1}{2}x^\top Q^+ x + \frac{1}{2}c^\top x - \frac{1}{2}u_x^\top Q^+ u_x + b^\top w_x \geq \tau, \\ A^\top w_x - Q^+ u_x \leq Q^- x + \frac{1}{2}c. \end{cases}$
	II	
$\mathcal{X} = \{x \in \mathbb{R}^{n_x} \mid Ax = b\}$	$\max_{\tau \in \mathbb{R}} \tau$	s.t. $\forall x \in \mathcal{X}, \exists(u_x, w_x) : \begin{cases} \frac{1}{2}x^\top Q^+ x + \frac{1}{2}c^\top x - \frac{1}{2}u_x^\top Q^+ u_x + b^\top w_x \geq \tau, \\ A^\top w_x - Q^+ u_x = Q^- x + \frac{1}{2}c. \end{cases}$
	III	
$\mathcal{X} = \{x \in \mathbb{R}^{n_x} \mid Ax \geq b, x \geq 0\}$	$\max_{\tau \in \mathbb{R}} \tau$	s.t. $\forall x \in \mathcal{X}, \exists(u_x, w_x) : \begin{cases} \frac{1}{2}x^\top Q^+ x + \frac{1}{2}c^\top x - \frac{1}{2}u_x^\top Q^+ u_x + b^\top w_x \geq \tau, \\ A^\top w_x - Q^+ u_x \leq Q^- x + \frac{1}{2}c, \\ w_x \geq 0. \end{cases}$
	IV	
$\mathcal{X} = \{x \in \mathbb{R}^{n_x} \mid Ax \geq b\}$	$\max_{\tau \in \mathbb{R}} \tau$	s.t. $\forall x \in \mathcal{X}, \exists(u_x, w_x) : \begin{cases} \frac{1}{2}x^\top Q^+ x + \frac{1}{2}c^\top x - \frac{1}{2}u_x^\top Q^+ u_x + b^\top w_x \geq \tau, \\ A^\top w_x - Q^+ u_x = Q^- x + \frac{1}{2}c, \\ w_x \geq 0. \end{cases}$

**Table 4.9:** Other classes of ARO reformulations of concave QO minimization.

Feasible Region	Type	ARO Problem
I		
$\mathcal{X} = \{x \in \mathbb{R}^{n_x} \mid Ax = b, x \geq 0\}$		$\begin{aligned} & \max_{\tau \in \mathbb{R}} \tau \\ & \text{s.t. } \forall x \in \mathcal{X}, \exists w_x : \begin{cases} \frac{1}{2}c^\top x + b^\top w_x \geq \tau, \\ A^\top w_x \leq Q^-x + \frac{1}{2}c. \end{cases} \end{aligned}$
II		
$\mathcal{X} = \{x \in \mathbb{R}^{n_x} \mid Ax = b\}$		$\begin{aligned} & \max_{\tau \in \mathbb{R}} \tau \\ & \text{s.t. } \forall x \in \mathcal{X}, \exists w_x : \begin{cases} \frac{1}{2}c^\top x + b^\top w_x \geq \tau, \\ A^\top w_x = Q^-x + \frac{1}{2}c. \end{cases} \end{aligned}$
III		
$\mathcal{X} = \{x \in \mathbb{R}^{n_x} \mid Ax \geq b, x \geq 0\}$		$\begin{aligned} & \max_{\tau \in \mathbb{R}} \tau \\ & \text{s.t. } \forall x \in \mathcal{X}, \exists w_x : \begin{cases} \frac{1}{2}c^\top x + b^\top w_x \geq \tau, \\ A^\top w_x \leq Q^-x + \frac{1}{2}c, \\ w_x \geq 0. \end{cases} \end{aligned}$
IV		
$\mathcal{X} = \{x \in \mathbb{R}^{n_x} \mid Ax \geq b\}$		$\begin{aligned} & \max_{\tau \in \mathbb{R}} \tau \\ & \text{s.t. } \forall x \in \mathcal{X}, \exists w_x : \begin{cases} \frac{1}{2}c^\top x + b^\top w_x \geq \tau, \\ A^\top w_x = Q^-x + \frac{1}{2}c, \\ w_x \geq 0. \end{cases} \end{aligned}$

**Table 4.10:** Detailed result on concave quadratic minimization.

Problem (size)	ARO-QO Algorithm			Selvi et al. method [105]			Gurobi			CPLEX		
	LB	UB	Time	LB	UB	Time	LB	UB	Time	LB	UB	Time
# 1 ( $m_x = 100, n_x = 50$ )	-21226.05	-20906.11	0.55	-23918.61	-20906.11	2.19	-20907.98	-20906.11	46.51	-20908.66	-20906.11	20.31
# 2 ( $m_x = 100, n_x = 50$ )	-19999.47	-19801.12	0.54	-22281.13	-19801.12	2.21	-19801.12	-19801.12	142.38	-10803.73	-19801.12	16.31
# 3 ( $m_x = 100, n_x = 50$ )	-21924.71	-21516.59	0.53	-24468.84	-21514.54	1.93	-21518.39	-21516.59	36.52	-21520.18	-21516.59	16.50
# 4 ( $m_x = 100, n_x = 50$ )	-19796.92	-19504.56	0.55	-22192.55	-19504.56	2.01	-19505.00	-19504.46	37.70	-19508.39	-19504.56	14.43
# 5 ( $m_x = 100, n_x = 50$ )	-19938.54	-19588.64	0.55	-22004.72	-19617.58	2.65	-19619.28	-19617.58	69.52	-19620.01	-19617.58	17.07
# 6 ( $m_x = 150, n_x = 100$ )	-43503.68	-42817.73	2.90	-48917.86	-42817.73	8.97	-42821.91	-42817.73	398.47	-42824.07	-42817.73	421.30
# 7 ( $m_x = 150, n_x = 100$ )	-44023.17	-43288.39	2.67	-49478.74	-43295.39	9.09	-43298.91	-43295.39	1138.38	-43304.86	-43295.39	523.09
# 8 ( $m_x = 150, n_x = 100$ )	-44509.83	-43708.24	2.94	-50215.65	-43746.44	9.61	-43753.45	-43749.45	832.73	-43756.27	-43749.36	578.56
# 9 ( $m_x = 150, n_x = 100$ )	-40043.22	-39464.10	2.53	-44570.54	-39479.17	8.72	-39479.86	-39477.46	1195.30	-39489.34	-39479.17	228.30
# 10 ( $m_x = 150, n_x = 100$ )	-50419.80	-49987.08	2.79	-56409.90	-49987.08	8.59	-49991.33	-49987.08	315.52	-49992.08	-49987.08	161.97
# 11 ( $m_x = 200, n_x = 100$ )	-36365.81	-35824.95	4.47	-41378.67	-35824.95	18.51	-35828.39	-35824.95	484.97	-35829.88	-35824.95	366.40
# 12 ( $m_x = 200, n_x = 100$ )	-21924.71	-21516.59	4.32	-41252.42	-35828.10	18.44	-35839.10	-35835.66	726.90	-35841.34	-35835.67	501.42
# 13 ( $m_x = 200, n_x = 100$ )	-34758.56	-34055.57	4.45	-39847.86	-34055.40	19.59	-34057.88	-34055.76	1636.05	-34135.30	-34055.76	301.26
# 14 ( $m_x = 200, n_x = 100$ )	-35864.55	-35262.45	4.58	-41288.01	-35289.22	16.38	-35310.24	-35289.70	3.000*	-35410.68	-35289.49	3.000*
# 15 ( $m_x = 200, n_x = 100$ )	-39351.10	-38553.98	4.67	-45260.71	-38619.98	16.24	-38628.29	-38624.93	2165.88	-38631.30	-38624.93	2256.70
# 16 ( $m_x = 250, n_x = 200$ )	-114818.87	-113258.29	14.88	-129138.55	-113258.34	48.23	-113269.93	-113258.34	3.000*	-113282.68	-113258.34	3.000*
# 17 ( $m_x = 250, n_x = 200$ )	-92364.54	-90802.03	13.38	-103508.71	-90833.68	47.94	-109407.37	-90185.44	3.000*	-91059.89	-90848.24	3.000*
# 18 ( $m_x = 250, n_x = 200$ )	-102592.36	-101763.08	16.13	-115061.49	-101763.08	48.82	-102331.34	-101763.08	3.000*	-101797.00	-101763.08	2638.61
# 19 ( $m_x = 250, n_x = 200$ )	-110848.56	-109342.56	16.27	-124919.22	-109342.56	52.02	-121382.85	-109342.56	3.000*	-109354.94	-109342.56	3.000*
# 20 ( $m_x = 250, n_x = 200$ )	-124939.05	-122854.09	15.89	-145278.15	-122854.09	59.25	-148108.59	-122854.09	3.000*	-123861.93	-122854.09	3.000*
# 21 ( $m_x = 300, n_x = 200$ )	-90379.80	-89076.48	24.68	-104968.70	-89054.41	83.60	-115882.61	-89076.79	3.000*	-89728.32	-89076.86	3.000*
# 22 ( $m_x = 300, n_x = 200$ )	-83366.49	-82259.34	26.20	-95965.30	-82256.29	97.87	-102793.52	-82259.34	3.000*	-82661.52	-82259.34	3.000*
# 23 ( $m_x = 300, n_x = 200$ )	-82027.18	-81016.91	24.37	-93726.28	-81008.71	138.95	-97987.79	-81016.91	3.000*	-81178.61	-81016.91	3.000*
# 24 ( $m_x = 300, n_x = 200$ )	-84667.10	-83459.67	24.30	-98686.53	-83457.97	102.20	-105898.98	-83459.67	3.000*	-83953.38	-83459.67	3.000*
# 25 ( $m_x = 300, n_x = 200$ )	-81894.50	-80966.99	23.75	-94430.77	-80966.99	82.63	-102639.56	-80966.99	3.000*	-81059.07	-80966.99	3.000*
# 26 ( $m_x = 350, n_x = 300$ )	-171137.64	-168437.60	50.91	-194844.52	-168438.92	170.15	-210038.17	-168194.52	3.000*	-5605813.93	-168194.52	3.000*
# 27 ( $m_x = 350, n_x = 300$ )	-164141.28	-161444.77	44.66	-186697.76	-161462.55	158.70	-205946.65	-161142.30	3.000*	-5128394.30	-161464.49	3.000*
# 28 ( $m_x = 350, n_x = 300$ )	-158477.21	-156089.67	59.39	-184476.06	-156064.84	146.55	-196949.54	-156089.67	3.000*	-372519.82	-156089.67	3.000*
# 29 ( $m_x = 350, n_x = 300$ )	-163020.17	-159968.26	37.23	-186943.98	-159857.19	149.84	-192162.18	-159559.82	3.000*	-295635.77	-159985.69	3.000*
# 30 ( $m_x = 350, n_x = 300$ )	-179027.10	-176570.47	59.21	-203873.39	-176564.79	144.81	-211848.25	-176570.47	3.000*	-694837.69	-176570.47	3.000*
# 31 ( $m_x = 400, n_x = 300$ )	-130525.34	-128324.26	80.92	-151936.50	-1238322.77	262.92	-171188.37	-128323.25	3.000*	-5164971.42	-128310.81	3.000*
# 32 ( $m_x = 400, n_x = 300$ )	-122185.75	-120302.84	71.80	-142516.23	-120310.31	230.07	-165499.48	-120287.00	3.000*	-5232838.82	-120308.58	3.000*
# 33 ( $m_x = 400, n_x = 300$ )	-120417.62	-118873.22	76.78	-139741.15	-118868.50	220.00	-163195.52	-118573.02	3.000*	-5201602.88	-118873.22	3.000*
# 34 ( $m_x = 400, n_x = 300$ )	-121135.65	-118904.01	70.01	-140455.45	-118869.00	281.14	-166977.00	-118898.47	3.000*	-5223379.15	-118898.58	3.000*
# 35 ( $m_x = 400, n_x = 300$ )	-135783.77	-134180.30	84.87	-157376.81	-134180.24	287.99	-182483.43	-134135.73	3.000*	-5207186.87	-134180.24	3.000*
# 36 ( $m_x = 450, n_x = 400$ )	-239708.24	-236691.08	69.61	-267620.18	-236691.08	292.47	-308141.60	-236632.22	3.000*	-1.02268 × 10 <sup>9</sup>	NA	3.000*
# 37 ( $m_x = 450, n_x = 400$ )	-212020.31	-21007.32	97.31	-252589.43	-215007.32	308.21	-401192.82	-213758.53	3.000*	-1.02521 × 10 <sup>9</sup>	NA	3.000*
# 38 ( $m_x = 450, n_x = 400$ )	-231523.46	-227609.94	94.57	-265032.88	-227619.86	307.32	-322874.67	-169857.26	3.000*	-1.02639 × 10 <sup>9</sup>	NA	3.000*
# 39 ( $m_x = 450, n_x = 400$ )	-225364.42	-221535.08	86.43	-258200.84	-221537.39	291.19	-623239.89	-169708.85	3.000*	-1.02626 × 10 <sup>9</sup>	NA	3.000*
# 40 ( $m_x = 450, n_x = 400$ )	-248061.41	-243641.36	129.44	-286303.36	-243639.15	278.68	-353455.02	-161208.64	3.000*	-1.02815 × 10 <sup>9</sup>	NA	3.000*
# 41 ( $m_x = 500, n_x = 400$ )	-162882.47	-160300.05	176.57	-191340.70	-160278.14	559.34	-604066.17	-159909.39	3.000*	-1.02268 × 10 <sup>9</sup>	NA	3.000*
# 42 ( $m_x = 500, n_x = 400$ )	-172886.97	-169806.25	164.70	-202411.67	-169805.29	586.96	-267361.99	-169267.40	3.000*	-1.02521 × 10 <sup>9</sup>	NA	3.000*
# 43 ( $m_x = 500, n_x = 400$ )	-161267.75	-158235.56	163.39	-187056.39	-158223.56	555.60	-425931.05	-110899.09	3.000*	-1.02626 × 10 <sup>9</sup>	NA	3.000*
# 44 ( $m_x = 500, n_x = 400$ )	-173508.69	-170210.36	156.92	-203828.67	-170199.68	534.90	-350152.46	-169666.30	3.000*	-1.02626 × 10 <sup>9</sup>	NA	3.000*
# 45 ( $m_x = 500, n_x = 400$ )	-177296.71	-173532.58	161.32	-208668.91	-173433.65	546.23	-404418.89	-173127.83	3.000*	-1.02815 × 10 <sup>9</sup>	NA	3.000*
# 46 ( $m_x = 550, n_x = 500$ )	-254200.02	-250228.59	140.01	-292466.45	-250227.15	699.22	-386062.82	-165730.74	3.000*	-2.00381 × 10 <sup>9</sup>	NA	3.000*
# 47 ( $m_x = 550, n_x = 500$ )	-283033.14	-278009.50	143.87	-326626.69	-277934.32	263.27	-13261942.44	-111018.29	3.000*	-1.99773 × 10 <sup>9</sup>	NA	3.000*
# 48 ( $m_x = 550, n_x = 500$ )	-267476.07	-262965.87	144.32	-305398.64	-262965.87	270.19	-738574.46	-153507.88	3.000*	-2.00129 × 10 <sup>9</sup>	NA	3.000*
# 49 ( $m_x = 550, n_x = 500$ )	-267667.57	-264084.34	160.95	-309881.77	-264016.01	244.89	-449419.26	-197858.04	3.000*	-2.00369 × 10 <sup>9</sup>	NA	3.000*
# 50 ( $m_x = 550, n_x = 500$ )	-293060.47	-287527.81	129.89	-335443.85	-287498.36	236.11	-423238.78	-191541.98	3.000*	-2.00595 × 10 <sup>9</sup>	NA	3.000*
# 51 ( $m_x = 600, n_x = 500$ )	-217416.92	-213705.89	366.84	-256282.73	-213683.16	515.61	-1127074.64	-166459.89	3.000*	-2.00381 × 10 <sup>9</sup>	NA	3.000*
# 52 ( $m_x = 600, n_x = 500$ )	-230918.43	-227145.96	336.32	-270398.12	-227144.69	514.71	-525624.12	-151916.19	3.000*	-1.99773 × 10 <sup>9</sup>	NA	3.000*
# 53 ( $m_x = 600, n_x = 500$ )	-211637.44	-207961.86	389.37	-247423.31	-207918.89	484.56	-11832043.00	-116131.71	3.000*	-2.00129 × 10 <sup>9</sup>	NA	3.000*
# 54 ( $m_x = 600, n_x = 500$ )	-240876.42	-236788.53	379.21	-283777.93	-236783.30	517.46	-8878852.73	-111854.65	3.000*	-2.00369 × 10 <sup>9</sup>	NA	3.000*
# 55 ( $m_x = 600, n_x = 500$ )	-221077.81	-217845.10	341.90	-257822.62	-217845.30	490.03	-8732517.01	-108571.64	3.000*	-2.00595 × 10 <sup>9</sup>	NA	3.000*
# 56 ( $m_x = 650, n_x = 600$ )	-391433.86	-384919.90	326.00	-455449.56	-384880.07	483.69	-4971784.16	-134292.50	3.000*	-3.45702 × 10 <sup>9</sup>	NA	3.000*
# 57 ( $m_x = 650, n_x = 600$ )	-403904.96	-398018.32	239.91	-463747.58	-398018.32	445.72	-14150233.89	-146383.96	3.000*	-3.45533 × 10 <sup>9</sup>	NA	3.000*
# 58 ( $m_x = 650, n_x = 600$ )	-329953.17	-324832.60	254.45	-381469.39	-324788.61	416.11	-3241488.76	-127909.30	3.000*	-3.45744 × 10 <sup>9</sup>	NA	3.000*
# 59 ( $m_x = 650, n_x = 600$ )	-344749.19	-339970.74	251.05	-394538.13	-339907.86	406.65	-18755129.41	-134434.84	3.000*	-3.46041 × 10 <sup>9</sup>	NA	3.000*
# 60 ( $m_x = 650, n_x = 600$ )	-344458.62	-338774.49	246.62	-397930.12	-338767.55	370.98	-20611553.36	-150155.61	3.000*	-3.46327 × 10 <sup>9</sup>	NA	3.000*
# 61 ( $m_x = 700, n_x = 600$ )	-282346.92	-278274.80	636.29	-332396.86	-278196.53	910.18	-12550163.40	-146157.71	3.000*	-3.45702 × 10 <sup>9</sup>	NA	3.000*
# 62 ( $m_x = 700, n_x = 600$ )	-255125.93	-252049.75	588.72	-297435.47	-252019.04	734.42	-13717479.02	-135118.87	3.000*	-3.45533 × 10 <sup>9</sup>	NA	3.000*
# 63 ( $m_x = 700, n_x = 600$ )	-268578.12	-265180.15	551.67	-310071.57	-265179.26	751.89	-17742893.76	-136924.49	3.000*	-3.45744 × 10 <sup>9</sup>	NA	3.000*
# 64 ( $m_x = 700, n_x = 600$ )	-271077.81	-265854.53	678.98	-321255.53	-265784.91	847.24	-13671678.23	-137967.52	3.000*	-3.46041 × 10 <sup>9</sup>	NA	3.000*
# 65 ( $m_x = 700, n_x = 600$ )	-265786.65	-261964.40	633.24	-312133.44	-261960.96	863.81	-13511151.74	-134274.22	3.000*	-3.46327 × 10 <sup>9</sup>	NA	3.000*
# 66 ( $m_x = 750, n_x = 700$ )	-453938.25	-445745.52	334.48	-522337.64	-445739.49	500.72	-305					

Table 4.11: Detailed numerical results on standard quadratic optimization.

Problem	ARO-QO Algorithm				Gurobi				CPLEX				IPOPT	
	LB1	LB2	UB	Time	LB1	UB1	LB2	UB2	LB1	UB1	LB2	UB2	UB	Time
#1 100(0.50)	-141.3034	-70.4688	-6.1407	3.21	-8.3740	-6.0793	-6.8351	-6.1300	-19568.7553	1.4379	-9.4376	-5.8325	-6.0793	0.18
#2 100(0.50)	-129.2121	-64.4946	-6.0452	1.41	-206.8990	-6.2006	-7.0153	-6.2006	-20689.9017	0.1291	-9.6186	-6.0611	-6.2006	0.17
#3 100(0.50)	-128.5047	-64.1669	-5.9070	3.73	-93.9099	-5.9070	-7.1628	-5.9070	-20289.6140	1.9297	-9.2160	-5.9789	-5.7568	0.15
#4 100(0.50)	-132.9374	-66.3307	-5.3757	2.34	-105.4804	-5.4065	-7.0147	-5.9348	-19603.0830	0.3966	-9.1551	-5.9122	-5.7282	0.19
#5 100(0.50)	-132.0628	-65.9230	-6.1510	1.96	-208.1827	-5.5674	-7.1317	-6.0872	-20818.2732	1.4012	-9.4403	-6.0872	-5.7499	0.15
#6 100(0.75)	-97.3824	-48.6858	-6.1247	4.13	-8.9226	-6.1421	-7.4236	-6.2123	-29864.6740	1.4379	-9.2783	-6.2409	-6.1421	0.22
#7 100(0.75)	-97.9984	-49.0243	-6.2670	4.24	-150.9282	-6.2129	-8.0329	-6.2129	-30875.3437	0.1291	-9.4080	-6.4196	-6.2129	0.17
#8 100(0.75)	-103.0299	-51.5410	-6.4087	4.13	-8.4681	-6.4087	-7.5404	-6.4087	-30985.4966	1.9297	-9.3935	-6.6311	-6.6184	0.27
#9 100(0.75)	-97.5577	-48.7893	-6.2600	2.04	-151.9978	-6.2442	-7.4113	-6.2442	-29785.6185	0.3966	-9.5376	-6.5648	-6.2600	0.19
#10 100(0.75)	-96.2869	-48.1865	-6.4669	1.68	-156.2187	-6.2761	-7.5161	-6.2761	-31184.9292	1.4012	-9.2898	-6.7674	-6.3080	0.20
#11 100(0.90)	-79.3832	-39.8410	-6.9327	2.62	-180.1334	-6.9327	-7.6421	-6.9327	-36746.4932	0.1291	-9.2769	-6.9327	-6.9327	0.24
#12 100(0.90)	-86.1553	-43.2040	-6.8661	1.97	-174.1012	-6.4303	-7.7573	-6.6220	-36937.4664	1.9297	-9.6367	-6.8661	-6.6465	0.42
#13 100(0.90)	-82.2343	-41.2319	-6.7313	2.72	-9.3498	-6.3954	-7.6163	-6.5053	-35500.0261	0.3966	-9.5383	-6.7313	-6.7313	0.24
#14 100(0.90)	-80.3149	-40.3119	-6.8193	4.33	-182.8205	-6.8193	-7.7276	-6.8193	-37305.5831	1.4012	-9.5492	-6.8193	-6.8193	0.22
#15 100(0.90)	-83.1796	-41.7040	-6.5117	3.92	-8.6935	-6.7201	-7.6462	-6.7201	-36100.2716	1.7340	-9.3520	-6.3773	-6.3773	0.39
#16 300(0.50)	-346.1800	-172.8466	-6.8268	20.93	-613.7949	-6.1414	-9.0272	-6.4021	-184138.4571	1.1983	-9.7702	-6.8268	-6.3178	2.07
#17 300(0.50)	-363.1453	-181.2841	-6.4665	10.28	-602.3853	-6.2261	-9.0093	-6.2261	-180715.6028	0.4993	-9.7162	-6.8293	-6.2681	3.46
#18 300(0.50)	-349.3163	-174.3961	-6.4054	8.44	-616.3506	-6.1621	-8.8538	-6.1621	-184905.1896	0.4407	-9.7923	-6.4981	-6.5653	2.43
#19 300(0.50)	-341.5578	-170.5206	-6.6315	8.83	-607.0261	0.9849	-8.9102	-6.3765	-182107.8395	1.5055	-9.7605	-6.1386	-5.8708	3.07
#20 300(0.50)	-345.3875	-172.4347	-6.1377	16.68	-184419.6908	NA	-8.9316	-6.6613	-184419.6908	1.3906	-9.7736	-6.6613	-6.1736	3.53
#21 300(0.75)	-176.3619	-88.1453	-6.8032	10.48	-921.1536	-6.9338	-9.0566	-6.9338	-276346.0771	1.1983	-9.7876	-6.8216	-6.8054	6.41
#22 300(0.75)	-175.4038	-87.6363	-6.8759	8.76	-272391.0266	NA	-8.9426	-6.4181	-272391.0266	0.4993	-9.8275	-6.7489	-6.8759	2.75
#23 300(0.75)	-174.4136	-87.1647	-7.0573	7.89	-277034.1229	NA	-9.1796	-6.7585	-277034.1229	0.4407	-9.8511	-7.0574	-6.7585	4.50
#24 300(0.75)	-177.9974	-88.9396	-6.6903	16.78	-912.3190	-6.9003	-8.9031	-6.9003	-273695.7025	1.5055	-9.8216	-6.6438	-7.0843	2.63
#25 300(0.75)	-175.5445	-87.7248	-6.9182	12.94	-921.0848	-6.6895	-8.9447	-6.6895	-276325.4527	1.3906	-9.8807	-6.6779	-6.6895	3.23
#26 300(0.90)	-143.7195	-71.9071	-6.9269	8.22	-330123.7775	NA	-9.0177	-6.9986	-330123.7775	1.1983	-1100.4094	-6.9986	-6.9986	8.11
#27 300(0.90)	-144.6976	-72.3644	-7.0452	9.76	-327243.9761	NA	-8.7635	-7.1325	-327243.9761	0.4993	-9.9144	-7.1256	-7.0452	12.17
#28 300(0.90)	-145.0177	-72.5449	-7.0388	7.48	-330364.3039	NA	-9.1546	-6.9562	-330364.3039	0.4407	-1101.2111	-7.0013	-7.0013	10.57
#29 300(0.90)	-143.8378	-71.9478	-7.1740	5.74	-327262.3799	NA	-9.0733	-6.9284	-327262.3799	1.5055	-1090.8714	-6.7852	-7.1740	11.89
#30 300(0.90)	-143.0871	-71.5811	-7.0907	9.91	-329821.4820	NA	-8.9564	-6.6353	-329821.4820	1.3906	-1099.4017	-7.0982	-7.0982	18.68
#31 500(0.50)	-556.7245	-278.0625	-6.6145	67.67	-1020.2667	-6.3477	-191.4577	-6.3477	-510133.3318	1.2865	-510133.3318	1.2865	-6.5915	19.84
#32 500(0.50)	-566.5476	-282.9713	-6.6280	31.42	-506289.0935	NA	-9.0589	-6.6493	-506289.0935	0.3183	-506289.0935	0.3183	-6.5356	18.31
#33 500(0.50)	-591.9429	-295.6179	-6.7359	23.70	-501373.7349	NA	-8.8296	-6.6624	-510373.7349	1.2783	-510373.7349	1.2783	-6.2281	41.77
#34 500(0.50)	-610.1031	-304.6871	-6.6229	25.60	-494369.5262	NA	-275.1602	-6.4848	-494369.5259	1.6020	-494369.5259	1.6020	-6.0928	56.92
#35 500(0.50)	-572.0808	-285.7164	-6.5621	21.67	-506389.7396	NA	-180.8172	-6.3804	-506389.7396	1.2845	-506389.7396	1.2845	-6.3461	22.11
#36 500(0.75)	-229.8191	-114.8414	-6.8795	18.03	-762239.6625	NA	-8.9277	-7.2121	-762239.6625	1.2865	-762239.6625	1.2865	-6.7796	71.14
#37 500(0.75)	-230.8269	-115.3499	-6.9959	20.02	-762805.0408	NA	-9.0610	-7.0364	-762805.0408	0.3183	-762805.0408	0.3183	-6.9959	41.26
#38 500(0.75)	-232.7850	-116.2777	-7.2736	17.02	-752040.9615	NA	-8.8130	-7.2737	-752040.9615	1.2783	-752040.9615	1.2783	-7.1736	31.93
#39 500(0.75)	-233.0709	-116.4148	-7.0284	18.91	-743920.3594	NA	-9.0686	-7.0285	-743920.3594	1.6020	-743920.3594	1.6020	-6.7045	38.31
#40 500(0.75)	-232.9520	-116.3886	-7.1457	14.91	-761558.1498	NA	-9.0567	-6.8525	-761558.1498	1.2845	-761558.1498	1.2845	-6.8502	65.33
#41 500(0.90)	-184.1009	-92.0577	-7.1770	16.11	-912849.6789	NA	-8.9426	-7.2882	-912849.6789	1.2865	-912849.6789	1.2865	-7.0698	269.61
#42 500(0.90)	-185.8445	-92.9335	-7.1591	20.84	-915735.5099	NA	-8.9811	-7.2046	-915735.5099	0.3183	-915735.5099	0.3183	-7.2841	193.03
#43 500(0.90)	-190.0275	-94.9696	-7.2765	22.76	-897979.1777	NA	-9.0566	-7.0170	-897979.1777	1.2783	-897979.1777	1.2783	-7.1358	558.81
#44 500(0.90)	-187.4554	-93.6838	-7.1644	15.61	-891704.7924	NA	-8.8940	-7.2341	-891704.7924	1.6020	-891704.7924	1.6020	-7.0817	132.88
#45 500(0.90)	-184.8534	-92.4199	-7.1304	15.40	-910365.1227	NA	-9.0567	-6.8725	-910365.1227	1.2845	-910365.1227	1.2845	-7.2025	233.34
#46 700(0.50)	-787.9514	-393.6440	-6.8559	49.49	-992790.3462	NA	-581.7589	-7.0743	-992790.3462	1.9870	-992790.3462	1.9870	-6.6087	111.28
#47 700(0.50)	-782.0549	-390.6996	-6.8288	59.65	-991913.5841	NA	-687.3291	-6.5275	-991913.5841	0.8314	-991913.5841	0.8314	-6.6004	122.00
#48 700(0.50)	-792.3280	-395.8237	-6.9251	53.28	-989990.5545	NA	-688.0122	-7.2341	-989990.5545	0.0685	-989990.5545	0.0685	-6.7706	120.14
#49 700(0.50)	-793.4050	-396.3786	-6.6120	37.83	-990734.2155	NA	-582.0707	-6.4569	-990737.2155	0.4787	-990737.2155	0.4787	-6.3972	157.87
#50 700(0.50)	-773.1635	-386.2475	-6.4638	36.06	-995054.2807	NA	-699.3185	-6.7537	-995054.2807	1.8646	-995054.2807	1.8646	-6.5121	127.89
#51 700(0.75)	-275.0992	-137.4745	-6.9744	29.52	-1485228.3068	NA	-1449.6412	-7.4271	-1485228.3068	1.9870	-1485228.3068	1.9870	-7.0766	401.49
#52 700(0.75)	-275.6102	-137.7144	-7.0034	27.70	-1491678.2390	NA	-1270.8686	-7.2887	-1491678.2390	0.8314	-1491678.2390	0.8314	-7.0057	634.64
#53 700(0.75)	-272.7944	-136.3043	-7.1935	30.72	-1484429.9024	NA	-889.5585	-7.1530	-1484429.9024	0.0685	-1484429.9024	0.0685	-6.8480	263.00
#54 700(0.75)	-276.2158	-138.0321	-7.1483	31.50	-1488303.0870	NA	-1454.5193	-7.1306	-1488303.0870	0.4787	-1488303.0870	0.4787	-7.2318	468.92
#55 700(0.75)	-273.1807	-136.5040	-7.5046	29.79	-1491721.6015	NA	-703.8169	-7.2129	-1491721.6015	1.8546	-1491721.6015	1.8546	-7.1273	292.65
#56 700(0.90)	-219.2574	-109.6259	-7.2968	29.91	-1780157.9382	NA	-1199.5401	-6.8760	-1780157.9382	1.9870	-1780157.9382	1.9870	-7.3375	123.17
#57 700(0.90)	-219.4832	-109.7182	-7.3353	32.24	-1792548.9534	NA	-1208.4181	-7.0783	-1792548.9534	0.8314	-1792548.9534	0.8314	-7.3220	128.02
#58 700(0.90)	-221.5076	-110.7260	-7.2549	34.67	-1778302.0147	NA	-1044.4092	-7.1261	-1778302.0147	0.0685	-1778302.0147	0.0685	-7.0970	128.38
#59 700(0.90)	-221.4515	-110.7163	-7.2375	33.91	-1784629.5061	NA	-1068.3227	-6.8956	-1784629.5060	0.4787	-1784629.5060	0.4787	-7.2374	82.89
#60 700(0.90)	-220.9667	-110.4617	-7.2504	28.08	-1785773.3406	NA	-1053.8475	-7.0932	-1785773.3406	1.8546	-1785773.3406	1.8546	-7.3478	105.62

Notes. This table presents problem numbers, dimensions, and matrix densities in the first column. For the ARO-QO Algorithm, ‘LB1’, ‘LB2’, ‘UB’, and ‘Time’ represent the lower bounds with partial decision rules on the ARO version, the upper bound values, and the computation times of all solvers, respectively. The results for Gurobi and CPLEX are divided into four subcolumns: the first two show bounds within the time limit required by the ARO-QO Algorithm, while the last two display bounds within a fixed 3,000-second limit. The IPOPT columns detail the upper bounds and the solver times achieved by IPOPT solvers. The ‘NA’ indicates the absence of a feasible solution within the given time.

# CHAPTER 5

## Conclusions and Future Research

Pliny the Elder:

*“The only certainty is that nothing is certain.”*

In this thesis, two noteworthy contributions are presented in the field of mathematical programming.

The first contribution introduces an innovative dual-based cutting plane method as a solution methodology for nonlinear Adjustable Robust Optimization (ARO) problems. This research introduces novel tools and techniques for robust optimization to address sequential decision-making under uncertainty, specifically in nonlinear cases. Indeed, we studied a general nonlinear ARO model with uncertainty in the objective and constraints. We obtained an equivalent dual formulation by applying Fenchel’s duality on the wait-and-see variable. We investigated when the dual formulation is convex in the decision variables and explored reaching upper and lower bounds for the original problem based on the dual formulation. Thanks to the equivalent dual reformulation, we presented and analyzed two algorithms. These algorithms aimed to find a lower bound on the optimal objective value of the general nonlinear ARO model. We demonstrated by numerical results that our algorithm could produce a locally robust solution with an acceptable optimality gap.

Secondly, we have introduced a novel reformulation technique that allows us to exactly reformulate the indefinite Quadratic programming Problem (QP) as a maximization adjustable robust optimization problem. This reformulation provides a new approach to approximately solving indefinite QP problems by transferring the complexity of the original problem to the nonlinearity of the equivalent ARO problem. We have demonstrated the effectiveness



of our ARO-based method in solving indefinite quadratic programming problems through numerical experiments and shown that it can provide high-quality solutions with reasonable computational costs. The established connection between QP and ARO, provides a new perspective on the challenges of indefinite QP problems and opens up new possibilities for further research in the field of mathematical optimization.

## 5.1 Future Directions

The findings outlined in this thesis offer exciting possibilities for advancing the methodology in nonlinear adjustable robust optimization and exploring problems in applied domains that have had limited exposure to sophisticated robust modeling techniques. This research prompts an immediate inquiry into integrating decision rules within nonlinear adjustable robust optimization and approximating this complex problem. Although incorporating decision rules presents both theoretical and computational challenges, it holds the promise of capturing the underlying structure more effectively.

Another future research avenue might explore how ARO can be applied to deriving new reformulations for disjoint biconvex programming problems. Disjoint biconvex problems are known to be complex and challenging to solve. We hope to develop innovative reformulations that can improve solvability and efficiency by leveraging the power of ARO techniques. This future research has the potential to advance the field of disjoint biconvex programming and contribute to the development of more robust optimization techniques.

## Bibliography

- [1] A. A. AHMADI AND J. ZHANG, *On the complexity of finding a local minimizer of a quadratic function over a polytope*, *Mathematical Programming*, 195 (2022), pp. 783–792.
- [2] A. ANDRIANOVA, A. KOREPANOVA, AND I. HALILOVA, *One algorithm for branch and bound method for solving concave optimization problem*, in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, 2016, p. 012005.
- [3] K. M. ANSTREICHER, *Semidefinite programming versus the reformulation-linearization technique for nonconvex quadratically constrained quadratic programming*, *Journal of Global Optimization*, 43 (2009), pp. 471–484.
- [4] A. ARDESTANI-JAAFARI AND E. DELAGE, *Linearized robust counterparts of two-stage robust optimization problems with applications in operations management*, *INFORMS Journal on Computing*, 33 (2021), pp. 1138–1161.
- [5] A. N. ARSLAN AND B. DETIENNE, *Decomposition-based approaches for a class of two-stage robust binary optimization problems*, *INFORMS Journal on Computing*, 34 (2022), pp. 857–871.
- [6] C. AUDET, P. HANSEN, AND G. SAVARD, *Essays and Surveys in Global Optimization*, vol. 7, Springer Science & Business Media, New York, 2005.
- [7] A. AUSLENDER AND M. TEBoulLE, *Asymptotic cones and functions in optimization and variational inequalities*, Springer Science & Business Media, New York, 2006.
- [8] X. BAO, N. V. SAHINIDIS, AND M. TAWARMALANI, *Semidefinite relaxations for quadratically constrained quadratic programming: A review and comparisons*, *Mathematical programming*, 129 (2011), pp. 129–157.
- [9] A. BECK, *First-order methods in optimization*, SIAM, Philadelphia, 2017.

- 
- [10] ———, *Introduction to Nonlinear Optimization: Theory, Algorithms, and Applications with Python and MATLAB, Second Edition*, SIAM, Philadelphia, 2023.
- [11] W. BEN-AMEUR, *Between fully dynamic routing and robust stable routing*, in 2007 6th International Workshop on Design and Reliable Communication Networks, IEEE, 2007, pp. 1–6.
- [12] A. BEN-TAL, D. DEN HERTOOG, AND J.-P. VIAL, *Deriving robust counterparts of nonlinear uncertain inequalities*, *Mathematical programming*, 149 (2015), pp. 265–299.
- [13] A. BEN-TAL, L. EL GHAOUI, AND A. NEMIROVSKI, *Robust optimization*, Princeton university press, Princeton, 2009.
- [14] A. BEN-TAL, A. GORYASHKO, E. GUSLITZER, AND A. NEMIROVSKI, *Adjustable robust solutions of uncertain linear programs*, *Mathematical programming*, 99 (2004), pp. 351–376.
- [15] A. BEN-TAL AND A. NEMIROVSKI, *Robust convex optimization*, *Mathematics of operations research*, 23 (1998), pp. 769–805.
- [16] ———, *Robust solutions of uncertain linear programs*, *Operations research letters*, 25 (1999), pp. 1–13.
- [17] A. BEN-TAL AND E. ROOS, *An algorithm for maximizing a convex function based on its minimum*, *INFORMS Journal on Computing*, 34 (2022), pp. 3200–3214.
- [18] M. BENTOBACHE, M. TELLI, AND A. MOKHTARI, *New LP-based local and global algorithms for continuous and mixed-integer nonconvex quadratic programming*, *Journal of Global Optimization*, 82 (2022), pp. 659–689.
- [19] D. BERTSIMAS AND H. BIDKHORI, *On the performance of affine policies for two-stage adaptive optimization: a geometric perspective*, *Mathematical Programming*, 153 (2015), pp. 577–594.
- [20] D. BERTSIMAS AND C. CARAMANIS, *Adaptability via sampling*, in 2007 46th IEEE Conference on Decision and Control, IEEE, 2007, pp. 4717–4722.

- [21] D. BERTSIMAS AND F. J. DE RUITER, *Duality in two-stage adaptive linear optimization: Faster computation and stronger bounds*, INFORMS Journal on Computing, 28 (2016), pp. 500–511.
- [22] D. BERTSIMAS AND D. DEN HERTOOG, *Robust and adaptive optimization*, Dynamic Ideas LLC, Belmont, MA, 2022.
- [23] D. BERTSIMAS AND I. DUNNING, *Multistage robust mixed-integer optimization with adaptive partitions*, Operations Research, 64 (2016), pp. 980–998.
- [24] ———, *Multistage robust mixed-integer optimization with adaptive partitions*, Operations Research, 64 (2016), pp. 980–998.
- [25] D. BERTSIMAS AND V. GOYAL, *On the power and limitations of affine policies in two-stage adaptive optimization*, Mathematical programming, 134 (2012), pp. 491–531.
- [26] 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 (2015), pp. 281–319.
- [27] D. BERTSIMAS, D. D. HERTOOG, J. PAUPHILET, AND J. ZHEN, *Robust convex optimization: A new perspective that unifies and extends*, Mathematical Programming, 200 (2023), p. 877–918.
- [28] D. BERTSIMAS, D. A. IANCU, AND P. A. PARRILO, *Optimality of affine policies in multistage robust optimization*, Mathematics of Operations Research, 35 (2010), pp. 363–394.
- [29] D. BERTSIMAS, D. A. IANCU, AND P. A. PARRILO, *A hierarchy of near-optimal policies for multistage adaptive optimization*, IEEE Transactions on Automatic Control, 56 (2011), pp. 2809–2824.
- [30] D. BERTSIMAS AND S. SHTERN, *A scalable algorithm for two-stage adaptive linear optimization*, arXiv:1807.02812, (2018).
- [31] S. BHANJA, D. KARUNARATNE, R. PANCHUMARTHY, S. RAJARAM, AND S. SARKAR, *Non-boolean computing with nanomagnets for computer vision applications*, Nature

- nanotechnology, 11 (2016), pp. 177–183.
- [32] I. M. BOMZE, *Regularity versus degeneracy in dynamics, games, and optimization: a unified approach to different aspects*, SIAM review, 44 (2002), pp. 394–414.
- [33] ———, *Copositive relaxation beats lagrangian dual bounds in quadratically and linearly constrained quadratic optimization problems*, SIAM Journal on Optimization, 25 (2015), pp. 1249–1275.
- [34] I. M. BOMZE AND E. DE KLERK, *Solving standard quadratic optimization problems via linear, semidefinite and copositive programming*, Journal of Global Optimization, 24 (2002), pp. 163–185.
- [35] I. M. BOMZE, M. LOCATELLI, AND F. TARDELLA, *New and old bounds for standard quadratic optimization: dominance, equivalence and incomparability*, Mathematical Programming, 115 (2008), pp. 31–64.
- [36] I. M. BOMZE, W. SCHACHINGER, AND R. ULLRICH, *The complexity of simple models—a study of worst and typical hard cases for the standard quadratic optimization problem*, Mathematics of Operations Research, 43 (2018), pp. 651–674.
- [37] P. BONAMI, A. LODI, J. SCHWEIGER, AND A. TRAMONTANI, *Solving quadratic programming by cutting planes*, SIAM Journal on Optimization, 29 (2019), pp. 1076–1105.
- [38] O. BONI AND A. BEN-TAL, *Adjustable robust counterpart of conic quadratic problems*, Mathematical Methods of Operations Research, 68 (2008), pp. 211–233.
- [39] S. P. BOYD AND L. VANDENBERGHE, *Convex optimization*, Cambridge university press, 2004.
- [40] D. J. BREUER, N. LAHRICHI, D. E. CLARK, AND J. C. BENNEYAN, *Robust combined operating room planning and personnel scheduling under uncertainty*, Operations research for health care, 27 (2020), p. 100276.
- [41] S. R. BULO, M. PELILLO, AND I. M. BOMZE, *Graph-based quadratic optimization: A fast evolutionary approach*, Computer Vision and Image Understanding, 115 (2011),

- pp. 984–995.
- [42] S. BUNDFUSS AND M. DUR, *An adaptive linear approximation algorithm for copositive programs*, SIAM Journal on Optimization, 20 (2009), pp. 30–53.
- [43] S. BURER, *On the copositive representation of binary and continuous nonconvex quadratic programs*, Mathematical Programming, 120 (2009), pp. 479–495.
- [44] S. BURER AND D. VANDENBUSSCHE, *A finite branch-and-bound algorithm for nonconvex quadratic programming via semidefinite relaxations*, Mathematical Programming, 113 (2008), pp. 259–282.
- [45] ———, *Globally solving box-constrained nonconvex quadratic programs with semidefinite-based finite branch-and-bound*, Computational Optimization and Applications, 43 (2009), pp. 181–195.
- [46] X. CEN AND Y. XIA, *A new global optimization scheme for quadratic programs with low-rank nonconvexity*, INFORMS Journal on Computing, 33 (2021), pp. 1368–1383.
- [47] H. CEVIKALP AND R. POLIKAR, *Local classifier weighting by quadratic programming*, IEEE Transactions on Neural Networks, 19 (2008), pp. 1832–1838.
- [48] J. CHEN AND S. BURER, *Globally solving nonconvex quadratic programming problems via completely positive programming*, Mathematical Programming Computation, 4 (2012), pp. 33–52.
- [49] A. CHINCHULUUN, P. M. PARDALOS, AND R. ENKHBAT, *Global minimization algorithms for concave quadratic programming problems*, Optimization, 54 (2005), pp. 627–639.
- [50] P. L. COMBETTES, *Perspective functions: Properties, constructions, and examples*, Set-Valued and Variational Analysis, 26 (2018), pp. 247–264.
- [51] T. H. CUONG, Y. LIM, AND N. D. YEN, *On a solution method in indefinite quadratic programming under linear constraints*, Optimization, (2022), pp. 1–26.
- [52] F. J. DE RUITER, J. ZHEN, AND D. DEN HERTOOG, *Dual approach for two-stage robust nonlinear optimization*, Operations Research, (2022).

- [53] F. J. DE RUITER, J. ZHEN, AND D. DEN HERTOOG, *Dual approach for two-stage robust nonlinear optimization*, Operations Research, 71 (2023), pp. 1794–1799.
- [54] E. DELAGE AND D. A. IANCU, *Robust multistage decision making*, in The operations research revolution, INFORMS, 2015, pp. 20–46.
- [55] W. S. DORN, *Duality in quadratic programming*, Quarterly of applied mathematics, 18 (1960), pp. 155–162.
- [56] B. DU, H. ZHOU, AND R. LEUS, *A two-stage robust model for a reliable  $p$ -center facility location problem*, Applied Mathematical Modelling, 77 (2020), pp. 99–114.
- [57] L. EL GHAOUI, F. OUSTRY, AND H. LEBRET, *Robust solutions to uncertain semidefinite programs*, SIAM Journal on Optimization, 9 (1998), pp. 33–52.
- [58] O. EL HOUSNI AND V. GOYAL, *On the optimality of affine policies for budgeted uncertainty sets*, Mathematics of Operations Research, 46 (2021), pp. 674–711.
- [59] M. FAMPA, J. LEE, AND W. MELO, *On global optimization with indefinite quadratics*, EURO Journal on Computational Optimization, 5 (2017), pp. 309–337.
- [60] L. E. GIBBONS, D. W. HEARN, P. M. PARDALOS, AND M. V. RAMANA, *Continuous characterizations of the maximum clique problem*, Mathematics of Operations Research, 22 (1997), pp. 754–768.
- [61] Y. G. GÖKMEN AND E. A. YILDIRIM, *On standard quadratic programs with exact and inexact doubly nonnegative relaxations*, Mathematical Programming, 193 (2022), pp. 365–403.
- [62] J. GONDZIO AND E. A. YILDIRIM, *Global solutions of nonconvex standard quadratic programs via mixed integer linear programming reformulations*, Journal of Global Optimization, 81 (2021), pp. 293–321.
- [63] J. GOUVEIA, T. K. PONG, AND M. SAEED, *Inner approximating the completely positive cone via the cone of scaled diagonally dominant matrices*, Journal of Global Optimization, 76 (2020), pp. 383–405.

- [64] L. GRIPPO AND M. SCIANDRONE, *On the convergence of the block nonlinear gauss–seidel method under convex constraints*, Operations research letters, 26 (2000), pp. 127–136.
- [65] GUROBI OPTIMIZATION, LLC: *Gurobi optimizer reference manual. version 10.0.0*, 2023.
- [66] M. J. HADJIYIANNIS, P. J. GOULART, AND D. KUHN, *A scenario approach for estimating the suboptimality of linear decision rules in two-stage robust optimization*, in 2011 50th IEEE Conference on Decision and Control and European Control Conference, IEEE, 2011, pp. 7386–7391.
- [67] G. A. HANASUSANTO, D. KUHN, AND W. WIESEMANN, *K-adaptability in two-stage robust binary programming*, Operations Research, 63 (2015), pp. 877–891.
- [68] H. HASHEMI DOULABI, P. JAILLET, G. PESANT, AND L.-M. ROUSSEAU, *Exploiting the structure of two-stage robust optimization models with exponential scenarios*, INFORMS Journal on Computing, 33 (2021), pp. 143–162.
- [69] J.-B. HIRIART-URRUTY AND C. LEMARÉCHAL, *Fundamentals of convex analysis*, Springer Science & Business Media, Berlin, 2004.
- [70] J. HU, J. E. MITCHELL, AND J.-S. PANG, *An lpcc approach to nonconvex quadratic programs*, Mathematical programming, 133 (2012), pp. 243–277.
- [71] T. IBARAKI AND N. KATOH, *Resource allocation problems: algorithmic approaches*, MIT press, Cambridge, 1988.
- [72] IBM ILOG CPLEX, *optimizer 12.10.0.0*, Webpage at IBM: <https://www.ibm.com/analytics/cplex-optimizer>, (2020).
- [73] S. KAMMAMMETTU AND Z. LI, *Two-stage robust optimization of water treatment network design and operations under uncertainty*, Industrial & Engineering Chemistry Research, 59 (2019), pp. 1218–1233.
- [74] G. Y. KE, *Managing reliable emergency logistics for hazardous materials: A two-stage robust optimization approach*, Computers & Operations Research, 138 (2022),



- p. 105557.
- [75] A. KHADEMI AND A. MARANDI, *Quadratic optimization through the lens of adjustable robust optimization*, Optimization Online, (2024).
- [76] A. KHADEMI, A. MARANDI, AND M. SOLEIMANI-DAMANEH, *A new dual-based cutting plane algorithm for nonlinear adjustable robust optimization*, Journal of Global Optimization, (2023).
- [77] F. KHADIVAR, K. CHATZILYGEROUDIS, AND A. BILLARD, *Self-correcting quadratic programming-based robot control*, IEEE Transactions on Systems, Man, and Cybernetics: Systems, (2023).
- [78] S. KIM, M. KOJIMA, AND K.-C. TOH, *A geometrical analysis on convex conic reformulations of quadratic and polynomial optimization problems*, SIAM Journal on Optimization, 30 (2020), pp. 1251–1273.
- [79] J. KOUSHKI, K. MIETTINEN, AND M. SOLEIMANI-DAMANEH, *LR-NIMBUS: an interactive algorithm for uncertain multiobjective optimization with lightly robust efficient solutions*, Journal of Global Optimization, 83 (2022), pp. 843–863.
- [80] M. K. KOZLOV, S. P. TARASOV, AND L. G. KHACHIYAN, *The polynomial solvability of convex quadratic programming*, USSR Computational Mathematics and Mathematical Physics, 20 (1980), pp. 223–228.
- [81] J. LEE, D. SKIPPER, AND E. SPEAKMAN, *Gaining or losing perspective*, Journal of Global Optimization, 82 (2022), pp. 835–862.
- [82] E. LIANG AND Z. YUAN, *Adjustable robust optimal control for industrial 2-mercaptobenzothiazole production processes under uncertainty*, Optimization and Engineering, (2022), pp. 1–38.
- [83] G. LIUZZI, M. LOCATELLI, AND V. PICCIALLI, *A new branch-and-bound algorithm for standard quadratic programming problems*, Optimization methods and software, 34 (2019), pp. 79–97.

- [84] J. LÖFBERG, *YALMIP: A toolbox for modeling and optimization in MATLAB*, in In Proceedings of the CACSD Conference, Taipei, Taiwan, 2004.
- [85] M. LU AND Z.-J. M. SHEN, *A review of robust operations management under model uncertainty*, Production and Operations Management, 30 (2021), pp. 1927–1943.
- [86] H. LUO, X. BAI, G. LIM, AND J. PENG, *New global algorithms for quadratic programming with a few negative eigenvalues based on alternative direction method and convex relaxation*, Mathematical Programming Computation, 11 (2019), pp. 119–171.
- [87] A. MARANDI AND D. DEN HERTOG, *When are static and adjustable robust optimization problems with constraint-wise uncertainty equivalent?*, Mathematical programming, 170 (2018), pp. 555–568.
- [88] A. MARANDI AND G.-J. VAN HOUTUM, *Robust location-transportation problems with integer-valued demand*, Optimization Online, (2020).
- [89] H. M. MARKOWITZ, *Profilio selection*, J. Financ., 7 (1952), pp. 77–91.
- [90] J. E. MITCHELL, J.-S. PANG, AND B. YU, *Convex quadratic relaxations of nonconvex quadratically constrained quadratic programs*, Optimization Methods and Software, 29 (2014), pp. 120–136.
- [91] MOSEK APS, *The MOSEK optimization toolbox for MATLAB manual. version 9.3.*, 2021.
- [92] M. O’NAN, *Linear Algebra*, Eagle mathematics series, Harcourt Brace Jovanovich, New York, 1971.
- [93] P. M. PARDALOS AND G. SCHNITGER, *Checking local optimality in constrained quadratic programming is np-hard*, Operations Research Letters, 7 (1988), pp. 33–35.
- [94] P. M. PARDALOS AND S. A. VAVASIS, *Quadratic programming with one negative eigenvalue is NP-hard*, Journal of Global optimization, 1 (1991), pp. 15–22.
- [95] J. PARK, *Sparsity-preserving difference of positive semidefinite matrix representation of indefinite matrices*, arXiv:1609.06762, (2016).

- [96] A. T. PHILLIPS AND J. B. ROSEN, *A parallel algorithm for constrained concave quadratic global minimization*, *Mathematical Programming*, 42 (1988), pp. 421–448.
- [97] K. POSTEK AND D. D. HERTOOG, *Multistage adjustable robust mixed-integer optimization via iterative splitting of the uncertainty set*, *INFORMS Journal on Computing*, 28 (2016), pp. 553–574.
- [98] J. RENEGAR, *A mathematical view of interior-point methods in convex optimization*, SIAM, Philadelphia, 2001.
- [99] R. T. ROCKAFELLAR, *Convex analysis*, vol. 36, Princeton university press, Princeton, 1970.
- [100] W. ROMEIJNDERS AND K. POSTEK, *Piecewise constant decision rules via branch-and-bound based scenario detection for integer adjustable robust optimization*, *INFORMS Journal on Computing*, 33 (2021), pp. 390–400.
- [101] K. ROOS, M. BALVERT, B. L. GORISSEN, AND D. DEN HERTOOG, *A universal and structured way to derive dual optimization problem formulations*, *INFORMS Journal on Optimization*, 2 (2020), pp. 229–255.
- [102] B. ROSTAMI, F. ERRICO, AND A. LODI, *A convex reformulation and an outer approximation for a large class of binary quadratic programs*, *Operations Research*, 71 (2023), pp. 471–486.
- [103] A. ROY, S. DABADGHAO, AND A. MARANDI, *Value of intermediate imaging in adaptive robust radiotherapy planning to manage radioresistance*, *Annals of Operations Research*, (2022), pp. 1–22.
- [104] A. SCOZZARI AND F. TARDELLA, *A clique algorithm for standard quadratic programming*, *Discrete Applied Mathematics*, 156 (2008), pp. 2439–2448.
- [105] A. SELVI, A. BEN-TAL, R. BREKELMANS, AND D. DEN HERTOOG, *Convex maximization via adjustable robust optimization*, *INFORMS Journal on Computing*, 34 (2022), pp. 2091–2105.

- [106] A. SELVI, D. DEN HERTOG, AND W. WIESEMANN, *A reformulation-linearization technique for optimization over simplices*, *Mathematical Programming*, 197 (2023), pp. 427–447.
- [107] A. SHAPIRO, D. DENTCHEVA, AND A. RUSZCZYNSKI, *Lectures on stochastic programming: modeling and theory*, SIAM, Philadelphia, 2021.
- [108] H. SHERALI AND L. LIBERTI, *Reformulation-linearization methods for global optimization. pardalos p, floudas c, eds., encyclopedia of optimization, 3263–3268*, 2008.
- [109] H. D. SHERALI AND A. ALAMEDDINE, *A new reformulation-linearization technique for bilinear programming problems*, *Journal of Global optimization*, 2 (1992), pp. 379–410.
- [110] H. D. SHERALI AND C. H. TUNCBILEK, *A reformulation-convexification approach for solving nonconvex quadratic programming problems*, *Journal of Global Optimization*, 7 (1995), pp. 1–31.
- [111] A. L. SOYSTER, *Convex programming with set-inclusive constraints and applications to inexact linear programming*, *Operations research*, 21 (1973), pp. 1154–1157.
- [112] A. SUBRAMANYAM, C. E. GOUNARIS, AND W. WIESEMANN, *K-adaptability in two-stage mixed-integer robust optimization*, *Mathematical Programming Computation*, 12 (2020), pp. 193–224.
- [113] A. TAKEDA, S. TAGUCHI, AND R. TÛTÛNCÛ, *Adjustable robust optimization models for a nonlinear two-period system*, *Journal of Optimization Theory and Applications*, 136 (2008), pp. 275–295.
- [114] P. VAYANOS, D. KUHN, AND B. RUSTEM, *Decision rules for information discovery in multi-stage stochastic programming*, in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, IEEE, 2011, pp. 7368–7373.
- [115] A. WÄCHTER AND L. T. BIEGLER, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, *Mathematical programming*, 106 (2006), pp. 25–57.

- [116] A. L. WANG AND F. KILIŒ-KARZAN, *On the tightness of SDP relaxations of QCQPs*, Mathematical Programming, 193 (2022), pp. 33–73.
- [117] L. WEI, A. GÓMEZ, AND S. KÜÇÜKYAVUZ, *Ideal formulations for constrained convex optimization problems with indicator variables*, Mathematical Programming, 192 (2022), pp. 57–88.
- [118] H. WOLKOWICZ, R. SAIGAL, AND L. VANDENBERGHE, *Handbook of semidefinite programming: theory, algorithms, and applications*, vol. 27, Springer Science & Business Media, Kluwer Academic Publishers, Boston, 2012.
- [119] D. WOOLNOUGH, V. JEYAKUMAR, AND G. LI, *Exact conic programming reformulations of two-stage adjustable robust linear programs with new quadratic decision rules*, Optimization Letters, 15 (2021), pp. 25–44.
- [120] W. XIA, J. C. VERA, AND L. F. ZULUAGA, *Globally solving nonconvex quadratic programs via linear integer programming techniques*, INFORMS Journal on Computing, 32 (2020), pp. 40–56.
- [121] P. XIDONAS, R. STEUER, AND C. HASSAPIS, *Robust portfolio optimization: A categorized bibliographic review*, Annals of Operations Research, 292 (2020), pp. 533–552.
- [122] G. XU AND S. BURER, *A copositive approach for two-stage adjustable robust optimization with uncertain right-hand sides*, Computational Optimization and Applications, 70 (2018), pp. 33–59.
- [123] İ. YANIKOĒLU, B. L. GORISSEN, AND D. DEN HERTOĒ, *A survey of adjustable robust optimization*, European Journal of Operational Research, 277 (2019), pp. 799–813.
- [124] N. ZADEH, *Note—a note on the cyclic coordinate ascent method*, Management Science, 16 (1970), pp. 642–644.
- [125] M. ZAMANI, *New bounds for nonconvex quadratically constrained quadratic programming*, Journal of Global Optimization, 85 (2023), pp. 595–613.

- [126] B. ZENG AND L. ZHAO, *Solving two-stage robust optimization problems using a column-and-constraint generation method*, *Operations Research Letters*, 41 (2013), pp. 457–461.
- [127] X. ZHANG AND X. LIU, *A two-stage robust model for express service network design with surging demand*, *European Journal of Operational Research*, 299 (2022), pp. 154–167.
- [128] J. ZHEN, D. DEN HERTOOG, AND M. SIM, *Adjustable robust optimization via fourier-motzkin elimination*, *Operations Research*, 66 (2018), pp. 1086–1100.
- [129] J. ZHEN, D. KUHN, AND W. WIESEMANN, *A unified theory of robust and distributionally robust optimization via the primal-worst-equals-dual-best principle*, *Operations Research*, (2023).
- [130] J. ZHEN, A. MARANDI, D. DE MOOR, D. DEN HERTOOG, AND L. VANDENBERGHE, *Disjoint bilinear optimization: A two-stage robust optimization perspective*, *INFORMS Journal on Computing*, 34 (2022), pp. 2410–2427.
- [131] X. J. ZHENG, X. L. SUN, AND D. LI, *Convex relaxations for nonconvex quadratically constrained quadratic programming: matrix cone decomposition and polyhedral approximation*, *Mathematical programming*, 129 (2011), pp. 301–329.

**ABBAS KHADEMI** (Shiraz, 1993) received his Bachelor's degree in Mathematics and Applications from Shiraz University, graduating as the 1st Rank student in 2016. He then pursued a Master's degree in Applied Mathematics (Operations Research and Optimization) from the University of Tehran, again graduating as the 1st Rank student in 2018. Following that, he embarked on a PhD journey in Applied Mathematics during which he visited as a visiting researcher at Eindhoven University of Technology from mid-March to mid-September 2021, within the duration of his PhD period.

This thesis presents two significant contributions to mathematical programming: Firstly, a novel dual-based cutting plane method is proposed for solving nonlinear Adjustable Robust Optimization (ARO) problems, addressing uncertainty in sequential decision-making. The study involves a general nonlinear ARO model with uncertain objectives and constraints. Fenchel's duality is applied to the wait-and-see variable, yielding an equivalent dual formulation. Algorithms are developed to find lower bounds on the optimal objective value, showcasing the approach's efficacy in producing locally robust solutions with acceptable optimality gaps. Secondly, a new reformulation technique transforms Quadratic Optimization (QO) problems into maximization adjustable robust optimization formulations. This innovative approach shifts the complexity from the original QO problem to the equivalent ARO problem. Numerical experiments validate the effectiveness of the ARO-based method in solving non-convex QO problems, offering high-quality solutions at reasonable computational costs. This connection between QO and ARO presents a fresh perspective on addressing non-convex QO challenges and opens avenues for future research in mathematical optimization.

February 6, 2024