Optimization Online

Quadratic Optimization Through the Lens of Adjustable Robust Optimization

Abbas Khademi
School of Mathematics, Statistics and Computer Science, College of Science, University of Tehran, Tehran, Iran, abbaskhademi@ut.ac.ir

Ahmadreza Marandi
Eindhoven University of Technology, Eindhoven Artificial Intelligence Systems Institute, The Netherlands
Department of Industrial Engineering and Innovation Sciences, Eindhoven University of Technology, Eindhoven, The Netherlands, a.marandi@tue.nl

Abstract. Quadratic optimization (QO) has been studied extensively in the literature due to its applicability in many practical problems. While practical, it is known that QO problems are generally NP-hard. So, researchers developed many approximation methods to find good solutions. In this paper, we go beyond the norm and analyze QO problems using robust optimization techniques. To this end, we first show that any QO problem can be reformulated as a disjoint bi-convex QO problem. Then, we provide an equivalent adjustable robust optimization (ARO) reformulation and leverage the methods available in the literature on ARO to approximate this reformulation. More specifically, we show that using a so-called decision rule technique to approximate the ARO reformulation is interpreted as using a linearization-relaxation technique on its bi-convex reformulation problem. Additionally, we design an algorithm that can find a close-to-optimal solution based on our new reformulations. Our numerical results demonstrate the efficiency of our algorithm, particularly for large-sized instances, compared with the off-the-shelf solvers.

Key words: Quadratic Optimization, Adjustable Robust Optimization, Duality, Affine Decision Rule, Linearization-Relaxation Technique

History: January 16, 2024

1. Introduction

Various practical problems in different domains, including financial mathematics (Markowitz 1952), machine learning (Cevikalp and Polikar 2008), resource allocation (Ibaraki and Katoh 1988), computer vision (Bhanja et al. 2016), game theory (Bomze 2002), robotic systems (Khadivar et al. 2023), graph theory (Gibbons et al. 1997), and image processing (Bulo et al. 2011), 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:

$$\min_{x\in\mathcal{X}} x^\top Qx + c^\top x, \tag{QO}$$

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 (Kozlov et al. 1980, Renegar 2001). In contrast, even when Q has only one negative eigenvalue, (QO) is NP-hard (Pardalos and Vavasis 1991). Besides, identifying local minimizers of (QO) over a polyhedron is not simpler than finding global minimizers from a complexity perspective (Ahmadi and Zhang 2022).

Due to the NP-hardness of indefinite QO problems, there has been a lot of research on constructing upper bounds via finding "good" solutions (Bentobache et al. 2022, Cuong et al. 2022), and lower bounds to identify the quality of a candidate solution, which are mainly based on linear or conic approximations (Mitchell et al. 2014, Rostami et al. 2023, Zamani 2023). A customary way to approximate a QO problem is by relaxing it into linear optimization problems, which is achieved through Reformulation-Linearization Techniques (RLT) (Anstreicher 2009, Sherali and Tuncbilek 1995). For an overview of RLTs, we refer the reader to the chapter (Sherali and Liberti 2009) 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 (Bomze 2015, Burer 2009). In such relaxations, the primary computational challenge shifts to deal with the copositive cone using tractable inner and outer approximations (Bundfuss and Dur 2009, Gouveia et al. 2020, Kim et al. 2020), or use a KKT-based branch-and-bound method (Chen and Burer 2012).

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 (Wolkowicz et al. 2012). Due to their efficiency, the SDO framework has led to many semi-definite relaxations; these relaxations are reviewed and compared in (Bao et al. 2011, Wang and Kılınç-Karzan 2022, Zheng et al. 2011). Moreover, Burer and Vandenbussche (2008, 2009) 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. Hu et al. (2012) and Xia et al. (2020) 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., Fampa et al. (2017) and Park (2016) 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 Cen and Xia (2021), the authors propose a solution scheme that involves solving a series of convex QO problems over the original feasible region. Additionally, Luo et al. (2019) 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 (Bomze and De Klerk

2002, Bomze et al. 2008, Bonami et al. 2019, Gökmen and Yıldırım 2022, Gondzio and Yıldırım 2021, Selvi et al. 2023).

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 Ben-Tal et al. (2004), 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 (Selvi et al. 2022) and bi-linear optimization (Zhen et al. 2022).

To obtain an approximate solution for an ARO problem, various techniques, such as the finite scenario approach (Hadjiyiannis et al. 2011), partitioning method (Bertsimas and Dunning 2016, Postek and Hertog 2016), Fourier-Motzkin elimination (Zhen et al. 2018), and decision rules (El Housni and Goyal 2021) 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 Delage and Iancu (2015) and the survey paper by Yanıkoğlu et al. (2019).

While there has been a lot of research in approximating linear ARO problems, the literature sparsely covers non-linear ARO problems due to their inherent complexity. De Ruiter et al. (2023) shows 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 the existing approximations techniques for linear cases. In a recent study, Khademi et al. (2024) employed 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 paper, 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 rest of the paper is structured as follows: in Section 1.1, we define the notation used throughout the paper. Section 2 introduces the reformulation of a QO problem as a bi-convex optimization problem and outlines its equivalent ARO problem. In Section 3, we approximate this problem using available techniques and prove the equivalence to an RLT for the original QO problem. Subsequently, in Section 4, we design an algorithm that provides a near-optimal solution for a QO problem using the ARO reformulation. Section 5 presents numerical results, demonstrating the efficiency of our ARO-based algorithm, particularly for large-sized instances. Finally, in Section 6, we summarize our findings and present our conclusions.

1.1. Notation

In this section, we introduce notations used in the paper. For a symmetric matrix B, we use $B \succeq 0$ ($B \succ 0$) to show B is positive semi-definite (positive definite), i.e., 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, $vec(B)$ 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 Δ.

2. New Reformulations for Quadratic Optimization Problems

This section proposes two reformulations for a quadratic optimization problem (QO). We first show how we can reformulate (QO) as 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 1. *Let $Q^+, -Q^- \succeq 0$, and $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ be an arbitrary set. Then,*

$$\min_{x \in \mathcal{X}} x^\top (Q^+ + Q^-) x + c^\top x \tag{1}$$

is equivalent to

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

Proof. It is clear that

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

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 $Q^-\preceq 0$, 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

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

The fact that

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

completes the proof. □

It is worth noting that the proof of Theorem 1 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 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 (O'Nan 1971). So, for an indefinite matrix Q, we can construct the matrices in Theorem 1 in many ways including the following representations:

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 ϵ is a small positive constant chosen to ensure that $Q^+, -Q^- \succeq 0$. Later, we discuss the effect of choosing Q^+ and Q^- on the quality of the approximations. □

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

PROPOSITION 1. *Let $Q = Q^+ + Q^-$ where $Q \in \mathbb{R}^{n_x \times n_x}$, and $Q^+, -Q^- \succ 0$. If x^* is an optimal solution of (QO), then (x^*, x^*) is an optimal solution of (Bi-QO). Moreover, if $(\hat{x}, \hat{y})$ is an optimal solution of (Bi-QO), then $\hat{x}$ and $\hat{y}$ are both optimal solutions of (QO).*

Proof. Suppose x^* is an optimal solution to (QO). It is clear that (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). Thus,

$$\begin{aligned}
&\tfrac{1}{2}\left(\hat{x}^\top Q^+ \hat{x} + \hat{y}^\top Q^+ \hat{y} + c^\top \hat{x} + c^\top \hat{y}\right) + \hat{x}^\top Q^- \hat{y} \le \tfrac{1}{2}\left(\hat{x}^\top Q^+ \hat{x} + \hat{x}^\top Q^+ \hat{x} + c^\top \hat{x} + c^\top \hat{x}\right) + \hat{x}^\top Q^- \hat{x},\\
&\tfrac{1}{2}\left(\hat{x}^\top Q^+ \hat{x} + \hat{y}^\top Q^+ \hat{y} + c^\top \hat{x} + c^\top \hat{y}\right) + \hat{x}^\top Q^- \hat{y} \le \tfrac{1}{2}\left(\hat{y}^\top Q^+ \hat{y} + \hat{y}^\top Q^+ \hat{y} + c^\top \hat{y} + c^\top \hat{y}\right) + \hat{y}^\top Q^- \hat{y}.
\end{aligned} \tag{2}$$

Summing these inequalities results in

$$2\hat{x}^\top Q^- \hat{y} \le \hat{x}^\top Q^- \hat{x} + \hat{y}^\top Q^- \hat{y}. \tag{3}$$

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

$$\begin{aligned}
2\hat{x}^\top(-Q^-)\hat{y} &\le 2\left\|(-Q^-)^{\frac{1}{2}}\hat{x}\right\| \cdot \left\|(-Q^-)^{\frac{1}{2}}\hat{y}\right\|\\
&= 2\sqrt{\hat{x}^\top(-Q^-)\hat{x}}\,\sqrt{\hat{y}^\top(-Q^-)\hat{y}}\\
&\le \hat{x}^\top(-Q^-)\hat{x} + \hat{y}^\top(-Q^-)\hat{y},
\end{aligned} \tag{4}$$

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

$$2\hat{x}^\top Q^- \hat{y} \ge \hat{x}^\top Q^- \hat{x} + \hat{y}^\top Q^- \hat{y}. \tag{5}$$

Thus, by (3) and (5), we obtain that

$$2\hat{x}^\top Q^- \hat{y} = \hat{x}^\top Q^- \hat{x} + \hat{y}^\top Q^- \hat{y}, \tag{6}$$

which is equivalent

$$(\hat{x}-\hat{y})^\top Q^-(\hat{x}-\hat{y}) = 0. \tag{7}$$

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

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

From now on, let us restrict the feasible region of (QO) to polytopes i.e., $\mathcal{X} = \{x \in \mathbb{R}^{n_x} \mid Ax = b,\ x \geq 0\}$ 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 2. *Let $Q = Q^+ + Q^-$ where $Q \in \mathbb{R}^{n_x \times n_x}$, and $Q^+, -Q^- \succeq 0$. 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:*

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

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 1, equivalent to

$$\min_{x, y \in \mathbb{R}^{n_x}} \left\{ \tfrac{1}{2}x^\top Q^+ x + \tfrac{1}{2}y^\top Q^+ y + x^\top Q^- y + \tfrac{1}{2}c^\top x + \tfrac{1}{2}c^\top y :\ x, y \in \mathcal{X} \right\}. \tag{8}$$

We can write (8) as

$$\min_{x \in \mathcal{X}} \left\{ \tfrac{1}{2}x^\top Q^+ x + \tfrac{1}{2}c^\top x + \min_{y \in \mathcal{X}} \tfrac{1}{2}y^\top Q^+ y + x^\top Q^- y + \tfrac{1}{2}c^\top y \right\}. \tag{9}$$

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 (Dorn 1960), and rewrite (9) as follows:

$$\begin{aligned} \min_{x \in \mathcal{X}} \tfrac{1}{2}x^\top Q^+ x + \tfrac{1}{2}c^\top x + \max_{u_x, w_x} & -\tfrac{1}{2}u_x^\top Q^+ u_x + b^\top w_x \\ \text{s.t. } & A^\top w_x - Q^+ u_x \leq Q^- x + \tfrac{1}{2}c. \end{aligned} \tag{10}$$

Let $x \in \mathcal{X}$. If the inner maximization is infeasible, its optimal value is $-\infty$, implying that (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 (10) as

$$\max_{\tau} \left\{ \tau \;\middle|\; \forall x \in \mathcal{X},\ \exists (u_x, w_x) : \begin{array}{l} \frac{1}{2}x^\top Q^+ x + \frac{1}{2}c^\top x - \frac{1}{2}u_x^\top Q^+ u_x + b^\top w_x \geq \tau, \\ A^\top w_x - Q^+ u_x \leq Q^- x + \frac{1}{2}c. \end{array} \right\}, \tag{11}$$

which completes the proof. $\square$

Problem (ARO-QO) is a quadratic ARO problem with fixed recourse and right-hand-side uncertainty. In this problem, τ 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 (Khademi et al. 2024, Yanıkoğlu et al. 2019).

It is important to note that concave QO, i.e., when dealing with a negative semi-definite matrix Q, is NP-hard. This complexity primarily arises from the crucial relationship between achieving optimality and enumerating the extreme points within the feasible region (Pardalos and Schnitger 1988). Predominant strategies for addressing concave QO problems typically involve cutting plane methods, branch and bound approaches, or iterative computational techniques (Audet et al. 2005, Andrianova et al. 2016, Chinchuluun et al. 2005, Phillips and Rosen 1988). Furthermore, recent studies in this area have focused on establishing bounds from a robust optimization perspective (Selvi et al. 2022), and some have adopted approaches based on gradient descent principles (Ben-Tal and Roos 2022); 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 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 compact. Then, the optimal value of (QO) is equal to the optimal value of the following problem:*

$$\begin{aligned} &\max_{\tau \in \mathbb{R}} \ \tau \\ &\text{s.t. } \forall x \in \mathcal{X},\ \exists w_x : \begin{cases} \frac{1}{2} c^\top x + b^\top w_x \geq \tau, \\ A^\top w_x \leq Q^- x + \frac{1}{2} c. \end{cases} \end{aligned} \tag{12}$$

Proof. From Theorem 2 by setting $Q^+ := 0$ and $Q^- := Q$. □

Note that (12) is a linear adjustable robust optimization problem and all techniques in the literature can be used to solve or approximate it.

REMARK 2. In Table 6 of Appendix B, we present the equivalent ARO formulations if the polytope $\mathcal{X}$ is formulated in another form than canonical. □

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.

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).

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., Bertsimas and Goyal (2012) and Bertsimas et al. (2015, 2010)).

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, we apply a hybrid decision rule to have a tractable approximation. 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):

$$\max_{u,z,Z,\tau} \left\{ \tau \;\middle|\; \begin{array}{ll} \frac{1}{2}x^\top Q^+ x + \frac{1}{2}c^\top x - \frac{1}{2}u^\top Q^+ u + b^\top(z + Zx) \geq \tau, & \forall x \in \mathcal{X} \\ A^\top(z + Zx) - Q^+ u \leq Q^- x + \frac{1}{2}c, & \forall x \in \mathcal{X} \end{array} \right\}, \tag{13}$$

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

In the previous section, we demonstrated that the (QO) problem is equivalent to both the (Bi-QO) and (ARO-QO) problems. In the rest of this section, we show that (13) is equivalent to applying a reformulation-linearization (RL) technique to (Bi-QO).

The literature has also considered RL techniques to approximate an ARO problem. More specifically, it is shown in Ardestani-Jaafari and Delage (2021) that a linear ARO problem can be reformulated as a bi-linear optimization problem using duality techniques. 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 Zhen et al. (2022), the same results are shown for disjoint bi-linear problems with convex feasible regions.

Considering (Bi-QO), using the RL technique proposed in Sherali and Alameddine (1992) and Sherali and Tuncbilek (1995) results in the following linear optimization problem:

$$\begin{aligned} \min_{\gamma,x,y} \quad & \tfrac{1}{2}\left(x^\top Q^+ x + y^\top Q^+ y + c^\top x + c^\top y\right) + \sum_{i,j=1}^{n_x} Q^-_{ij}\gamma_{ij} \\ \text{s.t.} \quad & Ax = b, \\ & Ay = b, \\ & A\gamma = by^\top, \\ & A\gamma^\top = bx^\top, \\ & x \geq 0,\; y \geq 0,\; \gamma \geq 0. \end{aligned} \tag{14}$$

In the next theorem, we show that (14) is the dual of the deterministic reformulation of the robust counterpart (13).

THEOREM 3. *Assume that $\mathcal{X}$ is a non-empty compact set. Then, the optimal value of* (13) *is equal to the optimal value of* (14).

Proof. We can rewrite (13) as

$$\max_{u,z,Z,\tau}\left\{\tau \;\middle|\; \begin{array}{ll} \min\limits_{x\in\mathcal{X}}\left\{\frac{1}{2}x^\top Q^+ x + (\frac{1}{2}c^\top + b^\top Z)x\right\} + b^\top z - \frac{1}{2}u^\top Q^+ u \geq \tau, & \\ \min\limits_{x\in\mathcal{X}}\left\{(-A^\top Z + Q^-)_i x\right\} + (\frac{1}{2}c + Q^+ u - A^\top z)_i \geq 0, & i = 1,\dots,n_x \end{array}\right\}. \tag{15}$$

Since $\mathcal{X}$ is a polytope, the inner minimizations are convex optimization problems. Since $\mathcal{X}$ is non-empty and compact, strong duality holds (Boyd and Vandenberghe 2004, Dorn 1960). Therefore, (15) is equivalent to

$$\begin{aligned} \max_{u,z,Z,\tau} \quad & \tau \\ \text{s.t.} \quad & \max_{\alpha,\beta}\left\{b^\top\beta - \tfrac{1}{2}\alpha^\top Q^+\alpha \;\middle|\; A^\top\beta - Q^+\alpha \leq (b^\top Z)^\top + \tfrac{1}{2}c\right\} + b^\top z - \tfrac{1}{2}u^\top Q^+ u \geq \tau, \\ & \max_{\theta^i}\left\{b^\top\theta^i \;\middle|\; A^\top\theta^i \leq \left((-A^\top Z + Q^-)_i\right)^\top\right\} + (\tfrac{1}{2}c = Q^+ u - A^\top z)_i \geq 0, \quad i = 1,\dots,n_x. \end{aligned} \tag{16}$$

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

$$\begin{aligned} \max_{u,z,Z,\alpha,\beta,\theta} \quad & b^\top\beta - \tfrac{1}{2}\alpha^\top Q^+\alpha + b^\top z - \tfrac{1}{2}u^\top Q^+ u \\ \text{s.t.} \quad & A^\top\beta - Q^+\alpha \leq (b^\top Z)^\top + \tfrac{1}{2}c, \\ & b^\top\theta^i + (\tfrac{1}{2}c + Q^+ u - A^\top z)_i \geq 0, \qquad i = 1,\dots,n_x, \\ & A^\top\theta^i \leq \left((-A^\top Z + Q^-)_i\right)^\top, \qquad i = 1,\dots,n_x. \end{aligned} \tag{17}$$

Now, we show that (17) is the dual problem of (14). To do this, we first write (14) in the matrix form:

$$\begin{aligned} \min_{vec(\gamma),x,y} \quad & \frac{1}{2}\begin{pmatrix} x \\ y \\ vec(\gamma) \end{pmatrix}^\top \begin{pmatrix} Q^+ & 0 & 0 \\ 0 & Q^+ & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ vec(\gamma) \end{pmatrix} + \begin{pmatrix} \frac{c}{2} \\ \frac{c}{2} \\ vec(Q^-) \end{pmatrix}^\top \begin{pmatrix} x \\ y \\ vec(\gamma) \end{pmatrix} \\ \text{s.t.} \quad & \begin{pmatrix} A & 0 & 0 \\ 0 & A & 0 \\ 0 & B & C \\ B & 0 & D \end{pmatrix} \begin{pmatrix} x \\ y \\ vec(\gamma) \end{pmatrix} = \begin{pmatrix} b \\ b \\ 0 \\ 0 \end{pmatrix}, \\ & \begin{pmatrix} x \\ y \\ vec(\gamma) \end{pmatrix} \geq 0, \end{aligned} \tag{18}$$

where $B := -\begin{pmatrix} b_1 I_{n_x} \\ b_2 I_{n_x} \\ \vdots \\ b_{m_x} I_{n_x} \end{pmatrix}$, $C := \begin{pmatrix} A_{11} I_{n_x} & A_{12} I_{n_x} & \dots & A_{1n_n} I_{n_x} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m_x 1} I_{n_x} & A_{m_x 2} I_{n_x} & \dots & A_{m_x n_x} I_{n_x} \end{pmatrix}$, 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 (18) is

$$\begin{aligned} \max_{Y,W} \quad & -\tfrac{1}{2} Y^\top \begin{pmatrix} Q^+ & 0 & 0 \\ 0 & Q^+ & 0 \\ 0 & 0 & 0 \end{pmatrix} Y + \begin{pmatrix} b \\ b \\ 0 \\ 0 \end{pmatrix}^\top W \\ \text{s.t.} \quad & -\begin{pmatrix} Q^+ & 0 & 0 \\ 0 & Q^+ & 0 \\ 0 & 0 & 0 \end{pmatrix} Y + \begin{pmatrix} A & 0 & 0 \\ 0 & A & 0 \\ 0 & B & C \\ B & 0 & D \end{pmatrix}^\top W \leq \begin{pmatrix} \frac{c}{2} \\ \frac{c}{2} \\ vec(Q^-) \end{pmatrix}. \end{aligned} \tag{19}$$

Setting

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

(19) is the matrix form of (17). Hence, (14) is the dual of the deterministic reformulation of (13). □

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, the ADR approximation is tight for a linear ARO problem with a right-hand-side uncertainty when the uncertainty set is simplex (Bertsimas and Bidkhori 2015). 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 (14).

As mentioned in Remark 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.

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 effectively handles problems involving a limited number of adjustable variables (Zhen et al. 2018).

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 $k \in \{1, \ldots, m_x\}$. To eliminate w_{x_k}, the k-th component of the vector w_x, we first isolate it in the constraints:

$$\begin{cases} b_k w_{x_k} \geq \tau - \frac{1}{2}x^\top Q^+ x - \frac{1}{2}c^\top x + \frac{1}{2}u_x^\top Q^+ u_x - \sum\limits_{\substack{j=1\\ j\neq k}}^{m_x} b_j w_{x_j}, \\ A_{ki} w_{x_k} \leq (Q^- x + \frac{1}{2}c + Q^+ u_x)_i - \sum\limits_{\substack{j=1\\ j\neq k}}^{m_x} A_{ji} w_{x_j}, \qquad i = 1, \ldots, m_x. \end{cases} \tag{20}$$

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

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:

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

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

$$\begin{aligned}
& \tfrac{1}{A_{kr}}((Q^- x + \tfrac{1}{2}c + Q^+ u_x)_i - \sum_{\substack{j=1\\ j\neq k}}^{m_x} A_{ji} w_{x_j}) \geq \\
& \qquad \tfrac{1}{b_k}(\tau - \tfrac{1}{2}x^\top Q^+ x - \tfrac{1}{2}c^\top x + \tfrac{1}{2}u_x^\top Q^+ u_x - \sum_{\substack{j=1\\ j\neq k}}^{m_x} b_j w_{x_j}) && \begin{array}{c} r = 1, \ldots, m_x, \\ \text{where } b_k > 0 \text{ and } A_{kr} > 0, \end{array}\\
& 0 \geq \tau - \tfrac{1}{2}x^\top Q^+ x - \tfrac{1}{2}c^\top x + \tfrac{1}{2}u_x^\top Q^+ u_x - \sum_{\substack{j=1\\ j\neq k}}^{m_x} b_j w_{x_j} && \text{where } b_k = 0,\\
& \tfrac{1}{A_{kr}}((Q^- x + \tfrac{1}{2}c + Q^+ u_x)_r - \sum_{\substack{j=1\\ j\neq k}}^{m_x} A_{jr} w_{x_j}) \geq \\
& \qquad \tfrac{1}{A_{ki}}((Q^- x + \tfrac{1}{2}c + Q^+ u_x)_i - \sum_{\substack{j=1\\ j\neq k}}^{m_x} A_{ji} w_{x_j}) && \begin{array}{c} i, r = 1, \ldots, m_x, \\ \text{where } A_{ki} < 0 \text{ and } \ A_{kr} > 0, \end{array}
\end{aligned}$$

$$(Q^{-}x+\tfrac{1}{2}c+Q^{+}u_x)_s-\sum_{\substack{j=1\\ j\neq k}}^{m_x}A_{js}w_{x_j}\geq 0 \qquad\qquad s=1,\dots,m_x,\ \text{where } A_{sk}^{\top}=0.$$

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. Solution Method

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

After solving the approximated problem (13), we use the obtained solution to extract worst-case scenarios from each constraint of the robust counterpart problem (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 (QO) problem. Given the selected scenario, we employ the mounting claiming algorithm (Algorithm 1) for (Bi-QO) to improve the quality of the solution. This process ultimately leads us to an upper bound for (QO) problem.

Algorithm 1 Mountain Climbing Procedure

Input: Matrix Q and starting point x^0.

Initialization: Decompose $Q=Q^{+}+Q^{-}$ such that $Q^{+},-Q^{-}\succeq 0$.

Repeat: Execute the following steps:

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

Until: No further improvement is possible.

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

By employing the ARO reformulation, bi-convex reformulation, and 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. Algorithm 2 presents the pseudo-code of the approach discussed above.

5. Numerical Experiments

In this section, we conduct a comprehensive numerical experiment to evaluate the efficacy of Algorithm 2, which we call 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.

Algorithm 2 ARO-Based Algorithm to Obtain an Upper Bound for QO

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

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

(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 3).

(Step 2) Generation of Worst-Case Scenarios: Generate a finite set of worst-case scenarios by substituting the optimal decision rule into (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 $x^{(0)}$:

$$x^{(k+1)} \longleftarrow \arg\min \left\{ \tfrac{1}{2} x^\top Q^+ x + x^\top Q^- x^{(k)} + \tfrac{1}{2} c^\top x : \, x \in \mathcal{X} \right\}.$$

(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 $UB := (x^*)^\top Q x^* + c^\top x^*$.

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 (Löfberg 2004).

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 (Gurobi Optimization 2023, version 10.0) and CPLEX (IBM ILOG CPLEX 2019, version 12.9). Given that our bounds require solving multiple linear optimization problems, we specifically employ Gurobi for this purpose. Moreover, MOSEK (MOSEK ApS 2023, version 10.1.15) is used to solve second-order cone optimization problems.

5.1. Concave Quadratic Minimization

Let us consider a concave quadratic minimization over a polyhedron

$$\begin{aligned} \min_{x \geq 0} \quad & x^\top Q x + c^\top x \\ \text{s.t.} \quad & Ax \geq b, \end{aligned} \tag{21}$$

where $Q \in \mathbb{R}^{n_x \times n_x}$, and $-Q \succeq 0$, $c \in \mathbb{R}^{n_x}$, $A \in \mathbb{R}^{m_x \times n_x}$, and $b \in \mathbb{R}^{m_x}$ are given. From Corollary 2 and Table 6 in Appendix B, we have the following linear ARO reformulation of (21):

$$\max_{\tau \in \mathbb{R}} \ \tau$$
$$\text{s.t. } \forall x \in \mathcal{X}, \ \exists w_x : \begin{cases} \frac{1}{2} c^\top x + b^\top w_x \geq \tau, \\ A^\top w_x \leq Qx + \frac{1}{2} c, \\ w_x \geq 0, \end{cases} \tag{22}$$

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 $r \in \{1, 2, \ldots, m_x\}$, $z \in \mathbb{R}^r$, $Z \in \mathbb{R}^{r \times n_x}$, 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 decision rule type has its own advantages and disadvantages, which we will address later in this section.

In Selvi et al. (2022), 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.

In this section, we compare the quality of the solution obtained by ARO-QO Algorithm with Gurobi and CPLEX, and Selvi et al. (2022) 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. Adding the small constant 10^{-4} in the denominator ensures the prevention of division by zero.

Problem Instances First, we consider the seven test instances from Section 4.3 of Selvi et al. (2022). We undertake a detailed comparison of three versions of ARO-QO Algorithm (static, partial, and fully affine), Selvi et al. (2022) 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 $r = [\frac{m_x}{7}] + 1$ 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 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 Selvi et al. (2022) 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 1 Optimality gaps and solver times of concave minimization instances from Selvi et al. (2022).

	Static ARO-QO Algorithm		Partial ARO-QO Algorithm		Affine ARO-QO Algorithm		Selvi et al. (2022) method		Gurobi		CPLEX	
Problem	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. The first column in this table presents the problem numbers and their corresponding dimensions. The symbol "-" indicates that it was not possible to determine the 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. As, the upper bound is calculated independently of the solution of the lower bound. This independence is based on the structure of problem (22). In Step 2 of the ARO-QO Algorithm, where worst-case scenarios are generated from problem (22), 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 8 in Appendix C. When compared with alternative approaches, such as applying partial or full affine decision rules or using the Selvi et al. (2022) 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 the mentioned method necessitates 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 Selvi et al. (2022), 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 (21) using 15 groups of random instances, with the dimension n_x taking value in $\{50, 100, \ldots, 600, 700\}$ and the number of constraints m_x spanning a range in $\{100, 150, \ldots, 750, 800\}$. Each group contains five instances of the same size, which are generated similarly to those created in Selvi et al. (2022).

For each group, ranging from #1 to #15, Table 2 lists the mean optimality gap and solver time, with standard deviations included in brackets (details can be found in Table 9). 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 Selvi et al. (2022) displays more consistent optimality gaps across all problem groups despite being significantly higher than those of Gurobi and CPLEX in the initial groups. In particular, Gurobi achieved very low optimality gaps in Groups 1-3, 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, could not find 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 2 Statistic of optimality gaps and solver times for randomly generated concave minimization instances.

	ARO-QO Algorithm		Selvi et al. (2022) method		Gurobi		CPLEX	
Group	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]	**0.01** [0.00]	66.53 [44.43]	**0.01** [0.00]	16.93 [2.14]
#2	1.49 [0.35]	2.77 [0.17]	13.81 [0.88]	9.00 [0.40]	**0.01** [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]	**0.02** [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*	**0.22** [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*	**0.43** [6.26]	3000*
#6	**1.62** [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	**1.53** [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	**1.70** [0.25]	95.42 [21.84]	16.21 [1.83]	295.57 [12.37]	118.90 [88.99]	3000*	-	3000*
#9	**1.89** [0.20]	164.98 [7.30]	19.38 [0.77]	556.61 [19.42]	171.95 [103.12]	3000*	-	3000*
#10	**1.79** [0.08]	143.81 [11.21]	16.92 [0.56]	256.74 [15.37]	2521.58 [5213.52]	3000*	-	3000*
#11	**1.64** [0.19]	362.73 [23.07]	19.24 [0.66]	504.47 [15.83]	5420.98 [4697.11]	3000*	-	3000*
#12	**1.57** [0.12]	263.61 [35.30]	17.17 [0.89]	424.63 [43.45]	8616.82 [5403.16]	3000*	-	3000*
#13	**1.48** [0.29]	617.78 [48.84]	18.89 [1.50]	821.51 [75.41]	10233.75 [1598.93]	3000*	-	3000*
#14	**1.79** [0.39]	343.31 [12.94]	17.33 [1.08]	503.12 [36.98]	-	3000*	-	3000*
#15	**1.48** [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 each subgroup's optimality gaps and solver time. The symbol "-" indicates that it was not possible to determine upper bounds for all instances of the corresponding group within the maximum time limit.

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 (Bomze and De Klerk 2002). 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: $x^\top Q x$, 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:

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

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

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

where τ 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

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

which is equivalent to the following deterministic convex quadratic optimization problem

$$\max_{z,u,z_0,\tau,\alpha,\beta,\theta} \left\{ \tau \;\middle|\; \begin{array}{l} -\frac{1}{2}\alpha^\top Q^+ \alpha + \beta + z_0 - \frac{1}{2}u^\top Q^+ u \geq \tau, \\ e\beta - Q^+ \alpha \leq z, \\ Q^+ u + \theta - e z_0 \geq 0, \\ e\theta^\top \leq \left(-e z^\top + Q^-\right)^\top, \end{array} \right\}. \tag{23}$$

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

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

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

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

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 (23) which is the deterministic reformulation of (L1-StQO). We select scenarios using the following optimization problems

$$\begin{cases} \bar{x}^0 \in \arg\min\limits_{x \in \Delta} \left\{ \frac{1}{2}x^\top Q^+ x + {z^*}^\top x \right\}, \\ \bar{x}^i \in \arg\min\limits_{x \in \Delta} \left\{ (-e{z^*}^\top + Q^-)_i x \right\}, \quad i = 1, \ldots, n_x. \end{cases} \tag{24}$$

Note that, we do not need to solve linear optimization problems in (24) to find $\{\bar{x}^i\}_{i=1}^{n_x}$, as we only 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 (Gondzio and Yıldırım 2021, Lemma 2.1 part (iv)). We choose the best scenario, and denote it by x^{*1}, as the one with the lowest objective value, i.e.,

$$x^{*1} \in \arg\min_x \left\{ x^\top Q x \middle| \, x \in \{\bar{x}^i\}_{i=0}^{n_x} \right\}.$$

Scenario Based on L2-StQO. We can find scenarios from the uncertainty set Δ according to (L2-StQO) as follows

$$\hat{x}^i \in \arg\min_{x \in \Delta} \left\{ \tfrac{1}{2} x^\top Q^+ x + Q_i^- x \right\}, \quad i = 1, \ldots, n_x. \tag{25}$$

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

$$x^{*2} \in \arg\min_x \left\{ x^\top Q x \middle| \, x \in \{\hat{x}^i\}_{i=1}^{n_x} \right\}.$$

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 (Wächter and Biegler 2006) is renowned for its time computational efficiency but functions exclusively as a local solver.

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-obtained solution 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 randomly generated StQO instances are located either at vertices or edges of the standard simplex (Bomze et al. 2018). In order to make a fair comparison, 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 Bonami et al. (2019), Liuzzi et al. (2019), and Scozzari and Tardella (2008).

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 UB is the upper bound for a given instance.

Detailed Results We consider the upper bounds obtained from global solvers when setting their time limit to the time taken by the ARO-QO Algorithm. We use SGap1 and Time1 to refer to this method's gap and solution time, respectively. We also set the time limit for the solvers to 3,000 seconds and refer to the gap by SGap 2 and the solution time by Time2.

The statistic of solution gaps for two classes of test problems is presented in Table 3 and 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 3 Statistic of solution gaps and solution times of (StQO) instances in Class One.

	ARO-QO Algorithm		Gurobi				CPLEX				IPOPT	
Class One	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 Bonami et al. (2019).

Table 4 Statistic of solution gaps and solution times of (StQO) instances in Class Two.

	ARO-QO Algorithm		Gurobi				CPLEX				IPOPT	
Class Two	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 Bonami et al. (2019).

In Tables 3 and 4, we observe that within the time taken by the ARO-QO Algorithm, CPLEX exhibits the best performance, and our approach outperforms 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. It is apparent that global solvers are generally effective in handling small 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 Liuzzi et al. (2019), while others are generated using the pattern described in Liuzzi et al. (2019) and Scozzari and Tardella (2008).

Table 5 provides the average solution gaps for each instance group (the details can be found in Table 10).

Table 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.

Table 5 Statistic of solution gaps and solution times of generated large-sized (StQO) instances.

	ARO-QO Algorithm		Gurobi		CPLEX		IPOPT	
Group	SGap(%)	Time	SGap1	SGap2	SGap1	SGap2	SGap	Time
#1	2.84 [4.36]	2.53 [0.94]	4.50 [5.17]	**0.49** [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]	**0.00** [0.00]	3.46 [2.77]	0.21 [0.04]
#3	**0.64** [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]	**1.81** [3.50]	7.57 [4.71]	2.91 [0.64]
#5	**1.56** [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]	**0.35** [0.55]	12.28 [3.92]
#7	**0.06** [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]	-	**0.86** [1.91]	652.02 [71.03]	652.02 [71.03]	3.22 [2.79]	49.59 [17.47]
#9	**1.06** [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]	-	**1.40** [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]	-	**1.21** [1.69]	657.54 [157.00]	657.54 [157.00]	3.87 [2.21]	412.14 [149.47]
#12	**0.38** [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	**1.64** [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 where the time limit is set to that needed by the ARO-QO Algorithm, while SGap2 denotes the solution gap within the 3000-second time limit. 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.

The ARO-QO Algorithm consistently demonstrates 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 bounds were loose for small instances based on Table 10, 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.

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.

Acknowledgement

The authors would like to express their appreciation to Dick Den Hertog for the insightful discussions and feedback. Additionally, the first author is grateful for the guidance and support provided by Majid Soleimani-damaneh throughout this research

Appendices

This appendix is divided into three sections. In the first section, we illustrate the relationship between the finite scenario approach of ARO reformulation and the optimality in (QO). The second section summarizes all Adjustable Robust Optimization (ARO) reformulations of QO problems based on their feasible regions. The final section contains elaborated tables related to the numerical experiments.

A. 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 in the uncertainty set, resulting in an upper bound to the optimal objective value of the ARO problem. 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, which results in the following deterministic convex optimization problem:

$$\max_{\{u^k\}_k,\{w^k\}_k,\tau} \left\{ \tau \;\middle|\; \begin{array}{ll} \frac{1}{2}(x^k)^\top Q^+ x^k + \frac{1}{2}c^\top x^k + b^\top w^k - \frac{1}{2}(u^k)^\top Q^+ u^k \geq \tau, & k=1,\ldots,|\mathcal{W}| \\ -Q^+ u^k + A^\top w^k \leq Q^- x^k + \frac{1}{2}c, & k=1,\ldots,|\mathcal{W}| \end{array} \right\}, \tag{FSA-QO}$$

where $\mathcal{W} = \{x^1, \ldots, x^r\}$ 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 2. *Let $Q = Q^+ + Q^-$ where $Q \in \mathbb{R}^{n_x \times n_x}$, and $Q^+, -Q^- \succ 0$. 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). Based on the definition of the optimality, there exists $x^s \in \mathcal{W} \subsetneq \mathcal{X}$ for which the following constraint of (FSA-QO) is binding:

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

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

$$\begin{aligned}\bar{\tau} &= \tfrac{1}{2}(x^s)^\top Q^+ x^s + \tfrac{1}{2}c^\top x^s + \max_{w,u}\left\{b^\top w - \tfrac{1}{2}u^\top Q^+ u : -Q^+ u + A^\top w \leq Q^- x^s + \tfrac{1}{2}c\right\} \\ &= \tfrac{1}{2}(x^s)^\top Q^+ x^s + \tfrac{1}{2}c^\top x^s + \min_{z}\left\{\tfrac{1}{2}z^\top Q^+ z + \tfrac{1}{2}c^\top z + z^\top Q^- x^s \,|\, z \in \mathcal{X}\right\},\end{aligned} \tag{26}$$

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 1, we have $x^s = z^s$, and therefore x^s is optimal for the original (QO) problem. □

B. Alternative Forms

A quadratic optimization problem can be expressed in different formulations, depending on how the feasible region is formulated, leading to variations in its ARO reformulation. Table 6 summarizes some of these formulations.

Table 6 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. $\forall x \in \mathcal{X},\ \exists(u_x, w_x): \begin{cases} \frac{1}{2}x^\top Q^+ x + \frac{1}{2}c^\top x - \frac{1}{2}u_x^\top Q^+ u_x + b^\top w_x \geq \tau, \\ A^\top w_x - Q^+ u_x = Q^- x + \frac{1}{2}c, \\ w_x \geq 0. \end{cases}$

C. Detailed Results of Numerical Experiments

This appendix presents the outcomes derived from the numerical experiments conducted in the respective sections. The results showcased herein provide detailed insights into the findings obtained through various computational analyses and experiments discussed throughout this paper.

Table 7 Comparison result on concave quadratic minimization test instances from Selvi et al. (2022).

	Static ARO-QO Algorithm			Partial affine ARO-QO Algorithm			Full affine ARO-QO Algorithm			Selvi et al. (2022) method			Gurobi			CPLEX		
Problem	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. The first column in this table presents the problem numbers and 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, and the 'UB' column displays the upper bounds.

Table 8 Time Results for concave quadratic minimization test instances from Selvi et al. (2022).

	Static ARO-QO			Partial affine ARO-QO			Full affine ARO-QO			Selvi et al. (2022) method			Global Solver	
Problem	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 9 Detailed result on concave quadratic minimization.

Problem (size)	ARO-QO Algorithm			Selvi et al. (2022) method			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 ($m_x = 200, n_x = 100$)	-36273.69	-35835.66	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	-161463.49	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*	-1.02268×10^9	NA	3,000*
# 37 ($m_x = 450, n_x = 400$)	-219201.30	-215007.32	97.31	-252589.43	-215005.17	308.21	-401192.82	-213758.53	3,000*	-1.02521×10^9	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*	-1.02639×10^9	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*	-1.02626×10^9	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*	-1.02815×10^9	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*	-1.02268×10^9	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*	-1.02521×10^9	NA	3,000*
# 43 ($m_x = 500, n_x = 400$)	-161267.75	-158235.56	165.39	-187056.39	-158223.56	555.60	-425931.05	-110899.09	3,000*	-1.02626×10^9	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*	-1.02626×10^9	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*	-1.02815×10^9	NA	3,000*
# 46 ($m_x = 550, n_x = 500$)	-254620.02	-250228.59	140.01	-292466.45	-250227.15	269.22	-386062.82	-165730.74	3,000*	-2.00381×10^9	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*	-1.99773×10^9	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*	-2.00129×10^9	NA	3,000*
# 49 ($m_x = 550, n_x = 500$)	-268766.57	-264084.34	160.95	-309881.77	-264016.01	244.89	-449419.26	-197858.04	3,000*	-2.00369×10^9	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*	-2.00595×10^9	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*	-2.00381×10^9	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*	-1.99773×10^9	NA	3,000*
# 53 ($m_x = 600, n_x = 500$)	-211637.44	-207961.86	389.37	-247423.31	-207918.89	484.56	-11832043.00	-111613.71	3,000*	-2.00129×10^9	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*	-2.00369×10^9	NA	3,000*
# 55 ($m_x = 600, n_x = 500$)	-220691.08	-217845.10	341.90	-257822.62	-217845.30	490.03	-8732517.01	-108571.64	3,000*	-2.00595×10^9	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*	-3.45702×10^9	NA	3,000*
# 57 ($m_x = 650, n_x = 600$)	-403904.96	-398018.32	239.91	-463747.58	-398018.32	445.72	-14155023.89	-146383.96	3,000*	-3.45533×10^9	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*	-3.45744×10^9	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*	-3.46041×10^9	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*	-3.46327×10^9	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*	-3.45702×10^9	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*	-3.45533×10^9	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*	-3.45744×10^9	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*	-3.46041×10^9	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*	-3.46327×10^9	NA	3,000*
# 66 ($m_x = 750, n_x = 700$)	-453938.25	-445745.52	334.48	-522337.64	-445739.49	500.72	-30556778.77	NA	3,000*	-5.49163×10^9	NA	3,000*
# 67 ($m_x = 750, n_x = 700$)	-392258.28	-383657.82	362.86	-449380.91	-385644.74	455.94	-29261626.30	-161772.44	3,000*	-548510×10^9	NA	3,000*
# 68 ($m_x = 750, n_x = 700$)	-446958.57	-441833.49	350.21	-512918.35	-441818.13	559.76	-29099763.10	NA	3,000*	-5.48474×10^9	NA	3,000*
# 69 ($m_x = 750, n_x = 700$)	-437897.81	-430075.17	332.64	-507811.80	-429904.39	501.20	-28450230.87	NA	3,000*	-5.48958×10^9	NA	3,000*
# 70 ($m_x = 750, n_x = 700$)	-396137.17	-388867.26	336.34	-461346.54	-388657.60	497.99	-28394163.71	NA	3,000*	-5.50234×10^9	NA	3,000*
# 71 ($m_x = 800, n_x = 700$)	-311760.73	-306985.03	987.30	-366078.98	-306837.21	938.07	-28336162.67	NA	3,000*	-5.49163×10^9	NA	3,000*
# 72 ($m_x = 800, n_x = 700$)	-331384.65	-326264.31	972.63	-392327.23	-326243.53	950.99	-29520962.09	NA	3,000*	-548510×10^9	NA	3,000*
# 73 ($m_x = 800, n_x = 700$)	-302347.42	-297341.90	925.81	-358956.95	-297338.12	939.47	-28577654.37	NA	3,000*	-5.48474×10^9	NA	3,000*
# 74 ($m_x = 800, n_x = 700$)	-317298.48	-312825.99	802.18	-371677.77	-312807.92	2086.76	-29767062.12	NA	3,000*	-5.48958×10^9	NA	3,000*
# 75 ($m_x = 800, n_x = 700$)	-325810.25	-322123.47	845.32	-376710.09	-322132.66	906.84	-28979308.79	NA	3,000*	-5.50234×10^9	NA	3,000*

Notes. In the table, we applied the ARO-QO Algorithm with a static decision rule in the lower bound approximation step. The 'NA' indicates that the solver could not find any feasible solution within the time limit, which was set at 3,000 seconds.

Table 10 Detailed numerical results on standard quadratic optimization.

	ARO-QO Algorithm				Gurobi				CPLEX				IPOPT	
Problem	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	7.83	-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	-7.2736	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	-1784629.0506	0.4787	-1784629.0506	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.

References

Ahmadi, A. A. and Zhang, J. (2022). On the complexity of finding a local minimizer of a quadratic function over a polytope. *Mathematical Programming*, 195(1-2):783–792.

Andrianova, A., Korepanova, A., and Halilova, I. (2016). One algorithm for branch and bound method for solving concave optimization problem. In *IOP Conference Series: Materials Science and Engineering*, page 012005. IOP Publishing.

Anstreicher, K. M. (2009). Semidefinite programming versus the reformulation-linearization technique for nonconvex quadratically constrained quadratic programming. *Journal of Global Optimization*, 43:471–484.

Ardestani-Jaafari, A. and Delage, E. (2021). Linearized robust counterparts of two-stage robust optimization problems with applications in operations management. *INFORMS Journal on Computing*, 33(3):1138–1161.

Audet, C., Hansen, P., and Savard, G. (2005). *Essays and Surveys in Global Optimization*, volume 7. Springer Science & Business Media, New York.

Bao, X., Sahinidis, N. V., and Tawarmalani, M. (2011). Semidefinite relaxations for quadratically constrained quadratic programming: A review and comparisons. *Mathematical Programming*, 129:129–157.

Ben-Tal, A., Goryashko, A., Guslitzer, E., and Nemirovski, A. (2004). Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99(2):351–376.

Ben-Tal, A. and Roos, E. (2022). An algorithm for maximizing a convex function based on its minimum. *INFORMS Journal on Computing*, 34(6):3200–3214.

Bentobache, M., Telli, M., and Mokhtari, A. (2022). New LP-based local and global algorithms for continuous and mixed-integer nonconvex quadratic programming. *Journal of Global Optimization*, 82(4):659–689.

Bertsimas, D. and Bidkhori, H. (2015). On the performance of affine policies for two-stage adaptive optimization: a geometric perspective. *Mathematical Programming*, 153(2):577–594.

Bertsimas, D. and Dunning, I. (2016). Multistage robust mixed-integer optimization with adaptive partitions. *Operations Research*, 64(4):980–998.

Bertsimas, D. and Goyal, V. (2012). On the power and limitations of affine policies in two-stage adaptive optimization. *Mathematical Programming*, 134(2):491–531.

Bertsimas, D., Goyal, V., and Lu, B. Y. (2015). A tight characterization of the performance of static solutions in two-stage adjustable robust linear optimization. *Mathematical Programming*, 150(2):281–319.

Bertsimas, D., Iancu, D. A., and Parrilo, P. A. (2010). Optimality of affine policies in multistage robust optimization. *Mathematics of Operations Research*, 35(2):363–394.

Bhanja, S., Karunaratne, D., Panchumarthy, R., Rajaram, S., and Sarkar, S. (2016). Non-boolean computing with nanomagnets for computer vision applications. *Nature Nanotechnology*, 11(2):177–183.

Bomze, I. M. (2002). Regularity versus degeneracy in dynamics, games, and optimization: a unified approach to different aspects. *SIAM review*, 44(3):394–414.

Bomze, I. M. (2015). Copositive relaxation beats lagrangian dual bounds in quadratically and linearly constrained quadratic optimization problems. *SIAM Journal on Optimization*, 25(3):1249–1275.

Bomze, I. M. and De Klerk, E. (2002). Solving standard quadratic optimization problems via linear, semidefinite and copositive programming. *Journal of Global Optimization*, 24(2):163–185.

Bomze, I. M., Locatelli, M., and Tardella, F. (2008). New and old bounds for standard quadratic optimization: dominance, equivalence and incomparability. *Mathematical Programming*, 115(1):31–64.

Bomze, I. M., Schachinger, W., and Ullrich, R. (2018). The complexity of simple models—a study of worst and typical hard cases for the standard quadratic optimization problem. *Mathematics of Operations Research*, 43(2):651–674.

Bonami, P., Lodi, A., Schweiger, J., and Tramontani, A. (2019). Solving quadratic programming by cutting planes. *SIAM Journal on Optimization*, 29(2):1076–1105.

Boyd, S. P. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Bulo, S. R., Pelillo, M., and Bomze, I. M. (2011). Graph-based quadratic optimization: A fast evolutionary approach. *Computer Vision and Image Understanding*, 115(7):984–995.

Bundfuss, S. and Dur, M. (2009). An adaptive linear approximation algorithm for copositive programs. *SIAM Journal on Optimization*, 20(1):30–53.

Burer, S. (2009). On the copositive representation of binary and continuous nonconvex quadratic programs. *Mathematical Programming*, 120(2):479–495.

Burer, S. and Vandenbussche, D. (2008). A finite branch-and-bound algorithm for nonconvex quadratic programming via semidefinite relaxations. *Mathematical Programming*, 113(2):259–282.

Burer, S. and Vandenbussche, D. (2009). Globally solving box-constrained nonconvex quadratic programs with semidefinite-based finite branch-and-bound. *Computational Optimization and Applications*, 43(2):181–195.

Cen, X. and Xia, Y. (2021). A new global optimization scheme for quadratic programs with low-rank nonconvexity. *INFORMS Journal on Computing*, 33(4):1368–1383.

Cevikalp, H. and Polikar, R. (2008). Local classifier weighting by quadratic programming. *IEEE Transactions on Neural Networks*, 19(10):1832–1838.

Chen, J. and Burer, S. (2012). Globally solving nonconvex quadratic programming problems via completely positive programming. *Mathematical Programming Computation*, 4(1):33–52.

Chinchuluun, A., Pardalos, P. M., and Enkhbat, R. (2005). Global minimization algorithms for concave quadratic programming problems. *Optimization*, 54(6):627–639.

Cuong, T. H., Lim, Y., and Yen, N. D. (2022). On a solution method in indefinite quadratic programming under linear constraints. *Optimization*, pages 1–26.

De Ruiter, F. J., Zhen, J., and Den Hertog, D. (2023). Dual approach for two-stage robust nonlinear optimization. *Operations Research*, 71(5):1794–1799.

Delage, E. and Iancu, D. A. (2015). Robust multistage decision making. In *The operations research revolution*, pages 20–46. INFORMS.

Dorn, W. S. (1960). Duality in quadratic programming. *Quarterly of Applied Mathematics*, 18(2):155–162.

El Housni, O. and Goyal, V. (2021). On the optimality of affine policies for budgeted uncertainty sets. *Mathematics of Operations Research*, 46(2):674–711.

Fampa, M., Lee, J., and Melo, W. (2017). On global optimization with indefinite quadratics. *EURO Journal on Computational Optimization*, 5(3):309–337.

Gibbons, L. E., Hearn, D. W., Pardalos, P. M., and Ramana, M. V. (1997). Continuous characterizations of the maximum clique problem. *Mathematics of Operations Research*, 22(3):754–768.

Gökmen, Y. G. and Yıldırım, E. A. (2022). On standard quadratic programs with exact and inexact doubly nonnegative relaxations. *Mathematical Programming*, 193(1):365–403.

Gondzio, J. and Yıldırım, E. A. (2021). Global solutions of nonconvex standard quadratic programs via mixed integer linear programming reformulations. *Journal of Global Optimization*, 81(2):293–321.

Gouveia, J., Pong, T. K., and Saee, M. (2020). Inner approximating the completely positive cone via the cone of scaled diagonally dominant matrices. *Journal of Global Optimization*, 76:383–405.

Gurobi Optimization (2023). LLC: Gurobi optimizer reference manual. version 10.0.0.

Hadjiyiannis, M. J., Goulart, P. J., and Kuhn, D. (2011). 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*, pages 7386–7391. IEEE.

Hu, J., Mitchell, J. E., and Pang, J.-S. (2012). An LPCC approach to nonconvex quadratic programs. *Mathematical Programming*, 133(1-2):243–277.

Ibaraki, T. and Katoh, N. (1988). *Resource allocation problems: algorithmic approaches*. MIT press, Cambridge.

IBM ILOG CPLEX (2019). optimizer 12.9.0.0. *Webpage at IBM: https://www. ibm. com/analytics/cplex-optimizer*.

Khademi, A., Marandi, A., and Soleimani-damaneh, M. (2024). A new dual-based cutting plane algorithm for nonlinear adjustable robust optimization. *Journal of Global Optimization*.

Khadivar, F., Chatzilygeroudis, K., and Billard, A. (2023). Self-correcting quadratic programming-based robot control. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.

Kim, S., Kojima, M., and Toh, K.-C. (2020). A geometrical analysis on convex conic reformulations of quadratic and polynomial optimization problems. *SIAM Journal on Optimization*, 30(2):1251–1273.

Kozlov, M. K., Tarasov, S. P., and Khachiyan, L. G. (1980). The polynomial solvability of convex quadratic programming. *USSR Computational Mathematics and Mathematical Physics*, 20(5):223–228.

Liuzzi, G., Locatelli, M., and Piccialli, V. (2019). A new branch-and-bound algorithm for standard quadratic programming problems. *Optimization Methods and Software*, 34(1):79–97.

Löfberg, J. (2004). YALMIP: A toolbox for modeling and optimization in MATLAB. In *In Proceedings of the CACSD Conference*, Taipei, Taiwan.

Luo, H., Bai, X., Lim, G., and Peng, J. (2019). New global algorithms for quadratic programming with a few negative eigenvalues based on alternative direction method and convex relaxation. *Mathematical Programming Computation*, 11(1):119–171.

Markowitz, H. M. (1952). Protfilio selection. *J. Financ.*, 7:77–91.

Mitchell, J. E., Pang, J.-S., and Yu, B. (2014). Convex quadratic relaxations of nonconvex quadratically constrained quadratic programs. *Optimization Methods and Software*, 29(1):120–136.

MOSEK ApS (2023). The MOSEK optimization toolbox for MATLAB manual. version 10.1.15.

O'Nan, M. (1971). *Linear Algebra*. Eagle mathematics series. Harcourt Brace Jovanovich, New York.

Pardalos, P. M. and Schnitger, G. (1988). Checking local optimality in constrained quadratic programming is np-hard. *Operations Research Letters*, 7(1):33–35.

Pardalos, P. M. and Vavasis, S. A. (1991). Quadratic programming with one negative eigenvalue is NP-hard. *Journal of Global Optimization*, 1(1):15–22.

Park, J. (2016). Sparsity-preserving difference of positive semidefinite matrix representation of indefinite matrices. *arXiv:1609.06762*.

Phillips, A. T. and Rosen, J. B. (1988). A parallel algorithm for constrained concave quadratic global minimization. *Mathematical Programming*, 42:421–448.

Postek, K. and Hertog, D. d. (2016). Multistage adjustable robust mixed-integer optimization via iterative splitting of the uncertainty set. *INFORMS Journal on Computing*, 28(3):553–574.

Renegar, J. (2001). *A mathematical view of interior-point methods in convex optimization*. SIAM, Philadelphia.

Rostami, B., Errico, F., and Lodi, A. (2023). A convex reformulation and an outer approximation for a large class of binary quadratic programs. *Operations Research*, 71(2):471–486.

Scozzari, A. and Tardella, F. (2008). A clique algorithm for standard quadratic programming. *Discrete Applied Mathematics*, 156(13):2439–2448.

Selvi, A., Ben-Tal, A., Brekelmans, R., and den Hertog, D. (2022). Convex maximization via adjustable robust optimization. *INFORMS Journal on Computing*, 34(4):2091–2105.

Selvi, A., den Hertog, D., and Wiesemann, W. (2023). A reformulation-linearization technique for optimization over simplices. *Mathematical Programming*, 197(1):427–447.

Sherali, H. and Liberti, L. (2009). *Reformulation-linearization methods for global optimization. In Encyclopedia of Optimization*, pages 3263–3268. Springer, Boston, MA.

Sherali, H. D. and Alameddine, A. (1992). A new reformulation-linearization technique for bilinear programming problems. *Journal of Global Optimization*, 2:379–410.

Sherali, H. D. and Tuncbilek, C. H. (1995). A reformulation-convexification approach for solving nonconvex quadratic programming problems. *Journal of Global Optimization*, 7:1–31.

Wächter, A. and Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106:25–57.

Wang, A. L. and Kılınç-Karzan, F. (2022). On the tightness of SDP relaxations of QCQPs. *Mathematical Programming*, 193(1):33–73.

Wolkowicz, H., Saigal, R., and Vandenberghe, L. (2012). *Handbook of semidefinite programming: theory, algorithms, and applications*, volume 27. Springer Science & Business Media, Kluwer Academic Publishers, Boston.

Xia, W., Vera, J. C., and Zuluaga, L. F. (2020). Globally solving nonconvex quadratic programs via linear integer programming techniques. *INFORMS Journal on Computing*, 32(1):40–56.

Yanıkoğlu, İ., Gorissen, B. L., and den Hertog, D. (2019). A survey of adjustable robust optimization. *European Journal of Operational Research*, 277(3):799–813.

Zamani, M. (2023). New bounds for nonconvex quadratically constrained quadratic programming. *Journal of Global Optimization*, 85(3):595–613.

Zhen, J., Den Hertog, D., and Sim, M. (2018). Adjustable robust optimization via fourier–motzkin elimination. *Operations Research*, 66(4):1086–1100.

Zhen, J., Marandi, A., de Moor, D., den Hertog, D., and Vandenberghe, L. (2022). Disjoint bilinear optimization: A two-stage robust optimization perspective. *INFORMS Journal on Computing*, 34(5):2410–2427.

Zheng, X. J., Sun, X. L., and Li, D. (2011). Convex relaxations for nonconvex quadratically constrained quadratic programming: matrix cone decomposition and polyhedral approximation. *Mathematical Programming*, 129(2):301–329.