MATHEMATICAL ENGINEERING
TECHNICAL REPORTS

Successive Lagrangian Relaxation Algorithm for Nonconvex Quadratic Optimization

Shinji YAMADA and Akiko TAKEDA

METR 2017–08 March 2017

DEPARTMENT OF MATHEMATICAL INFORMATICS
GRADUATE SCHOOL OF INFORMATION SCIENCE AND TECHNOLOGY
THE UNIVERSITY OF TOKYO
BUNKYO-KU, TOKYO 113-8656, JAPAN

WWW page: http://www.keisu.t.u-tokyo.ac.jp/research/techrep/index.html

Successive Lagrangian Relaxation Algorithm for Nonconvex Quadratic Optimization

Shinji Yamada * Akiko Takeda †

March 31th, 2017

Abstract

Optimization problems whose objective function and constraints are quadratic polynomials are called quadratically constrained quadratic programs (QCQPs). QCQPs are NP-hard in general and are important in optimization theory and practice. There have been many studies on solving QCQPs approximately. Among them, semi-definite program (SDP) relaxation is a well-known convex relaxation method. In recent years, many researchers have tried to find better relaxed solutions by adding linear constraints as valid inequalities. On the other hand, SDP relaxation requires a long computation time, and it has high space complexity for large-scale problems in practice; therefore, SDP relaxation may not be useful for such problems.

In this paper, we propose a new convex relaxation method that is weaker but faster than SDP relaxation methods. The proposed method transforms a QCQP into a Lagrangian dual optimization problem and successively solves subproblems while updating the Lagrange multipliers. The subproblem in our method is a QCQP with only one constraint for which we propose an efficient algorithm. Numerical experiments confirm that our method can quickly find a relaxed solution with an appropriate termination condition.

1 Introduction

We consider the following quadratically constrained quadratic program (QCQP):

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \quad & \boldsymbol{x}^\top Q_0 \boldsymbol{x} + 2\boldsymbol{q}_0^\top \boldsymbol{x} + \gamma_0 \\ \text{subject to} \quad & \boldsymbol{x}^\top Q_i \boldsymbol{x} + 2\boldsymbol{q}_i^\top \boldsymbol{x} + \gamma_i \leq 0, \qquad i = 1, \cdots, m, \end{aligned} \tag{1}$$

*S. Yamada is with the Department of Mathematical Informatics, Graduate School of Information Science and Technology, The University of Tokyo, Tokyo, Japan (email: shinji_yamada@mist.i.u-tokyo.ac.jp)

†A. Takeda is with the Department of Mathematical Analysis and Statistical Inference, The Institute of Statistical Mathematics, Tokyo, Japan (email: atakeda@ism.ac.jp)

where each Q_i is an $n \times n$ symmetric matrix. $Q_i = O$ means a linear function. We call a QCQP with m constraints m-QCQP. In the case $Q_i \succeq O$ for every $i = 0, \cdots, m$, (1) is a convex program. However, in general, positive semidefiniteness is not assumed, and (1) is NP-hard [23]. QCQPs are important in optimization theory and in practice.

QCQPs are fundamental nonlinear programming problems that appear in many applications such as max-cut problems [24] and binary quadratic optimizations. Some relaxation methods exist for finding a global solution of (1). A standard approach is the branch-and-bound (or cut) method, where a simple relaxation, e.g., a linear programming (LP) relaxation problem [2], is solved in each iteration. Audet, Hansen, Jaumard and Savard [5] proposed to introduce additional constraints, constructed using the reformulation linearization technique (RLT), to the relaxation problem. Lagrangian bounds, i.e., bounds computed by Lagrangian relaxation, have also been used in branch-and-bound methods in order to reduce the duality gap [22, 27, 29]. The branch-and-bound (or cut) algorithm yields a global solution by solving many relaxation subproblems, which restricts the size of QCQPs.

Another avenue of research has investigated tight relaxation problems for QCQPs. Among the many convex relaxation methods, semidefinite program (SDP) relaxation is well known and have been extensively studied [11, 19, 28]. It is known that an SDP relaxation can be viewed as a Lagrangian dual problem of the original QCQP. SDP relaxation has been applied to various QCQPs that appear in combinatorial optimization problems [12, 13] as well as in signal processing and communications [19]. SDP relaxation is powerful, and it gives the exact optimal value particularly when there is only one constraint on the trust region subproblem (TRS). Furthermore, recent studies such as [3] have proposed to add new valid inequalities (e.g., linear constraints for matrix variables using the original upper and lower bound constraints) to SDP relaxation problems. In particular, Zheng, Sun, and Li [30, 31] proposed a decomposition-approximation scheme that generates an SDP relaxation at least as tight as the ordinary one. Jiang and Li [16] proposed second order cone constraints as valid inequalities for the ordinary SDP relaxation. Such methods aim at obtaining a better relaxed solution even if they take more time to solve than the original SDP relaxation.

However, SDP relaxation including additional valid inequalities increases the problem size, which leads to a longer computation time and often memory shortage errors for large-scale problems. Here, Kim and Kojima [17] proposed a second-order cone programming relaxation (SOCP relaxation), where valid second-order cone constraints derived from the positive semidefinite inequality are added to the LP relaxation. Burer, Kim, and Kojima [9] proposed a "weaker but faster" method than SDP relaxation that uses a block matrix decomposition. Such faster relaxation methods are useful for large-scale problems, and they can be repeatedly solved in, e.g., a branch-and-bound method.

In this paper, we propose a faster convex relaxation method that is not stronger than SDP relaxation if valid constraints are not considered. Our method solves the Lagrangian dual problem of the original QCQP by using a subgradient method, though the dual problem can be reformulated as an SDP and is solvable with the interior point method. Indeed, there are various studies that propose to solve the Lagrangian dual problems for nonconvex problems, but most of them transform the dual problem into an SDP problem [6] or a more general cone problem [18]. Here, to resort to more easily solved problems, we divide the minimization of the objective function in the Lagrangian dual problem into two stages and iteratively solve the inner problem as a 1-QCQP, which can be solved exactly and quickly. There are mainly two approaches to solving a 1-QCQP: one is based on eigenvalue computation, the other on SDP relaxation. In particular, Moré and Sorensen [20] proposed to iteratively solve a symmetric positive-definite linear system for TRS, while Adachi, Iwata, Nakatsukasa, and Takeda [1] proposed an accurate and efficient method that solves only one generalized eigenvalue problem. In this paper, we propose a new relaxation method that can solve a 1-QCQP exactly and quickly as a convex quadratic optimization problem. Furthermore, we prove that the convex quadratic problem constructs the convex hull of the feasible region of a 1-QCQP. Numerical experiments confirm that our convex quadratic relaxation method for solving 1-QCQPs is faster than SDP relaxation and eigenvalue methods. They also show that our method can quickly find a relaxed solution of an m-QCQP by iteratively solving a 1-QCQP with updated Lagrange multipliers. By adding valid constraints to our formulation, our method can sometimes find a better relaxed solution in a shorter computation time compared with the ordinary SDP relaxation. The relaxation technique can be embedded within a branch-and-bound framework to determine a global optimum to the original m-QCQP.

The remainder of this paper is organized as follows. We introduce SDP relaxation and other related studies in Section 2. We describe our method and its some properties in Section 3 and 4. We present computational results in Section 5. We conclude the paper in Section 6. The Appendix contains our proofs of the presented theorems.

Throughout the paper, we denote matrices by using uppercase letters such as "Q", vectors by using bold lowercase such as "$\boldsymbol{q}$" and scalars by using normal lower case such as "γ". The notation $A \succ B$ or $A \succeq B$ implies that the matrix $A-B$ is positive definite or semidefinite. $\boldsymbol{e}$ means the all-one vector.

2 Existing SDP relaxation methods for m-QCQP

2.1 SDP relaxation

An SDP relaxation can be expressed as a Lagrangian dual problem of the original problem (1) as follows:

$$\max_{\boldsymbol{\xi}\geq\mathbf{0}} \quad \phi(\boldsymbol{\xi}). \tag{2}$$

Here, $\phi(\boldsymbol{\xi})$ is an optimal value function defined by

$$\phi(\boldsymbol{\xi}) := \min_{\boldsymbol{x}} \left[\boldsymbol{x}^\top \left(Q_0 + \sum_{i=1}^m \xi_i Q_i \right) \boldsymbol{x} + 2 \left(\boldsymbol{q}_0 + \sum_{i=1}^m \xi_i \boldsymbol{q}_i \right)^\top \boldsymbol{x} + \gamma_0 + \sum_{i=1}^m \xi_i \gamma_i \right], \tag{3}$$

$$= \begin{cases} -\boldsymbol{q}(\boldsymbol{\xi})^\top Q(\boldsymbol{\xi})^\dagger \boldsymbol{q}(\boldsymbol{\xi}) + \gamma(\boldsymbol{\xi}), & (\text{if } Q(\boldsymbol{\xi}) \succeq O), \\ -\infty, & (\text{otherwise}), \end{cases} \tag{4}$$

where $Q(\boldsymbol{\xi}) := Q_0 + \sum_{i=1}^m \xi_i Q_i$, $\boldsymbol{q}(\boldsymbol{\xi}) := \boldsymbol{q}_0 + \sum_{i=1}^m \xi_i \boldsymbol{q}_i$, $\gamma(\boldsymbol{\xi}) := \gamma_0 + \sum_{i=1}^m \xi_i \gamma_i$ and "†" means the pseudo-inverse. Note that from (4), (2) is equivalent to

$$\begin{aligned} \max_{\boldsymbol{\xi}\geq\mathbf{0}} \quad & \phi(\boldsymbol{\xi}) \\ \text{s.t.} \quad & Q_0 + \sum_{i=1}^m \xi_i Q_i \succeq O. \end{aligned} \tag{5}$$

By considering $-\boldsymbol{q}(\boldsymbol{\xi})^\top Q(\boldsymbol{\xi})^\dagger \boldsymbol{q}(\boldsymbol{\xi}) + \gamma(\boldsymbol{\xi})$ as a Schur complement of $\begin{pmatrix} Q(\boldsymbol{\xi}) & \boldsymbol{q}(\boldsymbol{\xi}) \\ \boldsymbol{q}(\boldsymbol{\xi})^\top & \gamma(\boldsymbol{\xi}) \end{pmatrix}$, we can equivalently rewrite the dual problem (5) as a semidefinite program (SDP)

$$\begin{aligned} \max_{\boldsymbol{\xi}\geq\mathbf{0},\tau} \quad & \tau \\ \text{s.t.} \quad & \begin{pmatrix} Q(\boldsymbol{\xi}) & \boldsymbol{q}(\boldsymbol{\xi}) \\ \boldsymbol{q}(\boldsymbol{\xi})^\top & \gamma(\boldsymbol{\xi}) - \tau \end{pmatrix} \succeq O, \end{aligned} \tag{6}$$

which can be solved by using an interior point method. It should be noted that the dual of (6) is

$$\begin{aligned} \min_{\boldsymbol{x},X} \quad & Q_0 \cdot X + 2\boldsymbol{q}_0^\top \boldsymbol{x} + \gamma_0 \\ \text{s.t.} \quad & Q_i \cdot X + 2\boldsymbol{q}_i^\top \boldsymbol{x} + \gamma_i \leq 0, \qquad i = 1, \cdots, m, \\ & X \succeq \boldsymbol{x}\boldsymbol{x}^\top \end{aligned} \tag{7}$$

and (6) and (7) are equivalent under the Primal/Dual Slater condition.

SDP relaxation is a popular approach to dealing with (1). Sturm and Zhang [26] proved that when there is one constraint (i.e. a 1-QCQP), SDP relaxation can always obtain the exact optimal value. Goemans and Williamson showed an approximation bound of SDP relaxation for max-cut problems [13], and Goemans [12] applied SDP relaxation to various combinatorial problems. Their numerical experiments show that SDP relaxation can find a very tight relaxed solution for many kinds of problems. However, SDP relaxation has disadvantages in both computation time and space complexity because of the matrix variable; it cannot deal with large-scale problems because of shortage of memory. Although polynomial time algorithms, such as an interior point method, have been established, they often take a long time to solve an SDP relaxation problem in practice.

2.2 Stronger SDP relaxation using RLT

For further strengthening the SDP relaxation, Anstreicher [3] proposed the reformulation linearization technique (RLT). Moreover, [3] added new constraints and restricted the range of the new variables X_{ij}, $\forall i, j$. Here, one assumes the original problem (1) has box constraints, i.e., lower and upper bounds on each variable x_j (l_j and u_j, respectively). Note as well that even if there are no box constraints, we may be able to compute l_j and u_j by using the original constraints if the feasible region is bounded. The inequality $l_j \leq x_j \leq u_j$ (as a vector expression, $\boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u}$) leads to

$$(x_i - u_i)(x_j - u_j) \geq 0 \iff x_i x_j - u_i x_j - u_j x_i + u_i u_j \geq 0, \tag{8}$$

$$(x_i - u_i)(x_j - l_j) \leq 0 \iff x_i x_j - u_i x_j - l_j x_i + u_i l_j \leq 0, \tag{9}$$

$$(x_i - l_i)(x_j - l_j) \geq 0 \iff x_i x_j - l_i x_j - l_j x_i + l_i l_j \geq 0, \tag{10}$$

for $i, j = 1, \cdots, n$. By replacing $x_i x_j$ with X_{ij}, we get

$$X_{ij} - u_i x_j - u_j x_i + u_i u_j \geq 0, \tag{11}$$

$$X_{ij} - u_i x_j - l_j x_i + u_i l_j \leq 0, \tag{12}$$

$$X_{ij} - l_i x_j - l_j x_i + l_i l_j \geq 0. \tag{13}$$

(11)$\sim$(13) are linear inequalities that include matrix variables X_{ij}. Therefore, by adding these constraints, we can get a stronger relaxation. The disadvantage of RLT is that it increases computation time because of the increased variables X_{ij} and additional constraints (11)$\sim$(13).

Many studies have aimed at strengthening the relaxation by adding valid inequalities other than (11)$\sim$(13) [16, 25, 30, 31]. Their methods give very tight bounds, but they entail large amounts of computation time.

2.3 Weaker SDP relaxation method by block decomposition

Burer, Kim, and Kojima [9] aims to solve a relaxed problem faster than SDP relaxation can, although it is a weaker relaxation; as such, it shares

a similar motivation as ours. First, [9] assumes that the original problem has [0,1] box constraints (i.e. $\forall i;\ 0 \leq x_i \leq 1$) in order to avoid a situation in which the optimal value diverges. Then, [9] proves that we can compute a block diagonal matrix D_i which satisfies $Q_i + D_i \succeq O$ for the matrix Q_i appearing in the objective function or the constraints. By using D_i, we can transform a quadratic polynomial,

$$\boldsymbol{x}^\top Q_i \boldsymbol{x} + 2\boldsymbol{q}_i^\top \boldsymbol{x} + \gamma_i = -\boldsymbol{x}^\top D_i \boldsymbol{x} + \boldsymbol{x}^\top (Q_i + D_i)\boldsymbol{x} + 2\boldsymbol{q}_i^\top \boldsymbol{x} + \gamma_i$$

and relax $\boldsymbol{x}^\top D_i \boldsymbol{x}$ to $D_i \cdot X$ and $X \succeq O$ as in SDP relaxation. As a whole, a relaxation problem is as follows.

$$\begin{aligned}
\min_{\boldsymbol{x},X} \quad & -D_0 \cdot X + \boldsymbol{x}^\top (Q_0 + D_0)\boldsymbol{x} + 2\boldsymbol{q}_0^\top \boldsymbol{x} + \gamma_0 \\
\text{s.t.} \quad & -D_i \cdot X + \boldsymbol{x}^\top (Q_i + D_i)\boldsymbol{x} + 2\boldsymbol{q}_i^\top \boldsymbol{x} + \gamma_i \leq 0, \qquad i = 1, \cdots, m, \\
& X_k \succeq \boldsymbol{x}_k \boldsymbol{x}_k^\top, \qquad\qquad\qquad\qquad\qquad\quad k = 1, \cdots, r,
\end{aligned}$$

where r denotes the number of blocks of D_i and X_k or $\boldsymbol{x}_k$ denotes a partial matrix or vector in X or $\boldsymbol{x}$ corresponding to each block of D_i. Note that in a similar way as (12), we get new constraints $X_{ii} \leq x_i,\ i = 1, \cdots, m$ for the matrix variable X from the box constraints. Since we relax only the quadratic form for each block part, the matrix X only has block part components. Therefore, we can consider the positive semidefinite constraint only for the block parts: $X_k \succeq \boldsymbol{x}_k \boldsymbol{x}_k^\top$. The number of variables related to the positive semidefinite constraint is reduced, and that is why we can obtain the optimal value so quickly. We call this method **Block-SDP** and use it in the numerical experiments in Section 5.

In [9], it is proposed to divide D_i as evenly as possible, that is, by making the difference between the largest block size and the smallest block size at most one for a given r.

3 Proposed Method

3.1 Assumptions

Before we explain our method, we will impose the following three assumptions.

Assumption 1. *(a) The feasible region of* (1) *has some interior points.*

(b) There exists at least one matrix Q_i $(i = 0, \cdots, m)$ *such that* $Q_i \succ O$.

(c) When $Q_0 \succeq O$, *any optimal solution* $\bar{\boldsymbol{x}} := -Q_0^\dagger \boldsymbol{q}_0$ *of the following unconstrained optimization problem*

$$\min_{\boldsymbol{x}} \qquad \boldsymbol{x}^\top Q_0 \boldsymbol{x} + 2\boldsymbol{q}_0^\top \boldsymbol{x} + \gamma_0. \tag{14}$$

is not feasible for the original QCQP (1).

Assumption 1 (a) is the primal Slater condition, and (b) is a sufficient condition of the Dual Slater condition of the original QCQP. Assumption 1 (c) is not a strong one because if $\bar{\boldsymbol{x}}$ is feasible for (1), it is a global optimal solution and we can check it easily.

3.2 The Whole Algorithm

We further transform the Lagrangian dual problem (2) into

$$\max_{\boldsymbol{\lambda}\in\Lambda_{\rm s}} \quad \max_{\mu\geq 0} \quad \phi(\mu\boldsymbol{\lambda}), \tag{15}$$

where $\Lambda_{\rm s} := \{\boldsymbol{\lambda} \geq \mathbf{0} \mid \ \boldsymbol{e}^\top\boldsymbol{\lambda} = 1\}$ is a simplex. Now we define $\psi(\boldsymbol{\lambda})$ as the optimal value of the inner optimization problem of (15) for a given $\boldsymbol{\lambda} \in \Lambda_{\rm s}$:

$$\psi(\boldsymbol{\lambda}) := \quad \max_{\mu\geq 0} \quad \phi(\mu\boldsymbol{\lambda}). \tag{16}$$

Note that (16) is the Lagrangian dual problem for the following 1-QCQP:

$$\begin{aligned} \psi(\boldsymbol{\lambda}) = \ & \min_{\boldsymbol{x}} \ \ \boldsymbol{x}^\top Q_0\boldsymbol{x} + 2\boldsymbol{q}_0^\top\boldsymbol{x} + \gamma_0 \\ & \text{s.t.} \ \ \boldsymbol{x}^\top\left(\sum_{i=1}^m \lambda_i Q_i\right)\boldsymbol{x} + 2\left(\sum_{i=1}^m \lambda_i\boldsymbol{q}_i\right)^\top\boldsymbol{x} + \sum_{i=1}^m \lambda_i\gamma_i \leq 0. \end{aligned} \tag{17}$$

There is no duality gap between (16) and its Lagrangian dual (17), since [26] proves that the SDP formulation of (16) has the same optimal value as the 1-QCQP (17). We will show how to solve the 1-QCQP (17) exactly and quickly in Section 4.1. The SDP relaxation problem (2) can be written as

$$\max_{\boldsymbol{\lambda}\in\Lambda_{\rm s}} \quad \psi(\boldsymbol{\lambda}). \tag{18}$$

Here, we propose an algorithm which iteratively solves (17) with updated $\boldsymbol{\lambda} \in \Lambda_{\rm s}$ for finding an optimal solution of the SDP relaxation problem. $\Lambda_{\rm s}$ is a convex set, and $\psi(\boldsymbol{\lambda})$ is a quasi-concave function, as shown in Section 3.3. Therefore, we will apply the standard gradient descent method to (18) for updating $\boldsymbol{\lambda}$. The speed of convergence of gradient methods is slow in general especially near optimal solutions, and therefore, we will obtain a relaxed solution by using an appropriate termination criterion. **Algorithm 1** summarizes the proposed method.

We divide the "max" for the Lagrange function into two parts and iteratively solve the 1-QCQPs. Note that our method solves the 1-QCQPs $\psi(\boldsymbol{\lambda})$ successively, so it is not stronger than SDP relaxation. We explain our method (especially, the relationship between (P_k) and (27) or (28)) in Section 4.

Algorithm 1 Successive Lagrangian Relaxation (SLR)

Given $Q_0, \cdots, Q_m$ $(\exists i;\ Q_i \succ O)$, $\boldsymbol{q}_0, \cdots, \boldsymbol{q}_m$, $\gamma_0, \cdots, \gamma_m$, tolerance ϵ and sufficiently small value $\psi(\boldsymbol{\lambda}^{-1})$,

Step 1: Set $k = 0$, and define an initial point $\boldsymbol{\lambda}^{(0)}$.

Step 2: Find an optimal solution $\boldsymbol{x}^{(k)}$ and the optimal value $\psi(\boldsymbol{\lambda}^{(k)})$ of (P_k):

$$\begin{aligned} \psi(\boldsymbol{\lambda}^{(k)}) \ = \ & \min_{\boldsymbol{x}} \ \boldsymbol{x}^\top Q_0 \boldsymbol{x} + 2\boldsymbol{q}_0^\top \boldsymbol{x} + \gamma_0 \\ & \text{s.t.} \ \ \boldsymbol{x}^\top \left(\sum_{i=1}^m \lambda_i^{(k)} Q_i \right) \boldsymbol{x} + 2 \left(\sum_{i=1}^m \lambda_i^{(k)} \boldsymbol{q}_i \right)^\top \boldsymbol{x} + \sum_{i=1}^m \lambda_i^{(k)} \gamma_i \leq 0 \end{aligned} \qquad (\mathrm{P}_k)$$

by solving the convex problem (27) or (28).

Step 3: If $\left| \frac{\psi(\boldsymbol{\lambda}^k) - \psi(\boldsymbol{\lambda}^{k-1})}{\psi(\boldsymbol{\lambda}^{k-1})} \right| < \epsilon$, then stop the algorithm. Otherwise, update $\boldsymbol{\lambda}^{(k)}$ by **Algorithm 2** shown in Section 4.2 and $k \leftarrow k + 1$. Go to **Step 2**.

3.3 Quasi-Concavity of $\psi(\boldsymbol{\lambda})$

Objective functions of Lagrangian dual problems are concave for Lagrange multipliers (e.g. [7]). The function $\phi(\mu\boldsymbol{\lambda})$ for fixed $\boldsymbol{\lambda}$ is hence concave for μ, but $\psi(\boldsymbol{\lambda})$ is not necessarily concave for $\boldsymbol{\lambda}$. However, we can prove that $\psi(\boldsymbol{\lambda})$ is a quasi-concave function and has some of the desirable properties that concave functions have.

Before we prove the quasi-concavity of $\psi(\boldsymbol{\lambda})$, we have to define the set Λ_+,

$$\Lambda_+ = \{\boldsymbol{\lambda} \in \Lambda_{\mathrm{s}} \mid \psi(\boldsymbol{\lambda}) > \phi(\boldsymbol{0})\}, \qquad (19)$$

in order to explain the properties of $\psi(\boldsymbol{\lambda})$. Note that $\phi(\boldsymbol{0})$ is the optimal value of the unconstrained problem (14) and $\psi(\boldsymbol{\lambda}) \geq \phi(\boldsymbol{0})$ holds for all $\boldsymbol{\lambda}$. We can also see that Λ_+ is nonempty if and only if the SDP relaxation value is larger than $\phi(\boldsymbol{0})$, i.e., $\mathrm{OPT}_{\mathrm{SDP}} > \phi(\boldsymbol{0})$ holds. The above statement is obvious from $\mathrm{OPT}_{\mathrm{SDP}} = \max\{\psi(\boldsymbol{\lambda}) \mid \boldsymbol{\lambda} \in \Lambda_{\mathrm{s}}\}$ (see (18)) and (19). In other words, $\mathrm{OPT}_{\mathrm{SDP}} = \phi(\boldsymbol{0})$ means that for all $\boldsymbol{\lambda} \in \Lambda_{\mathrm{s}}$, there exists an optimal solution of (14) which is feasible for (17).

From the definition of Λ_+, we can see that for $\boldsymbol{\lambda} \in \Lambda_+$, an optimal solution of (16), $\bar{\mu}_{\boldsymbol{\lambda}}$, is a positive value. When $\boldsymbol{\lambda} \notin \Lambda_+$ (i.e. $\psi(\boldsymbol{\lambda}) = \phi(\boldsymbol{0})$), we can set $\bar{\mu}_{\boldsymbol{\lambda}}$ to zero without changing the optimal value and solution. By

using such $\bar{\mu}_{\boldsymbol{\lambda}}$, we will identify (19) and

$$\Lambda_+ = \{\boldsymbol{\lambda} \in \Lambda_{\rm s} \mid \bar{\mu}_{\boldsymbol{\lambda}} > 0\}. \tag{20}$$

Now let us prove the quasi-concavity of $\psi(\boldsymbol{\lambda})$ and some other properties..

Theorem 1. *Let $\psi(\boldsymbol{\lambda})$ be the optimal value of* (16) *and $(\bar{\boldsymbol{x}}_{\boldsymbol{\lambda}}, \bar{\mu}_{\boldsymbol{\lambda}})$ be its optimal solution. Then, the following* (i)∼(iv) *hold.*

(i) The vector

$$(\tilde{\boldsymbol{g}}_{\boldsymbol{\lambda}})_i = \bar{\mu}_{\boldsymbol{\lambda}} \left(\bar{\boldsymbol{x}}_{\boldsymbol{\lambda}}^\top Q_i \bar{\boldsymbol{x}}_{\boldsymbol{\lambda}} + 2\boldsymbol{q}_i^\top \bar{\boldsymbol{x}}_{\boldsymbol{\lambda}} + \gamma_i \right) \tag{21}$$

is a subgradient, which is defined as a vector in the quasi-subdifferential (see e.g., [14, 15]):

$$\partial\psi(\boldsymbol{\lambda}) := \{\boldsymbol{s} \mid \boldsymbol{s}^\top(\boldsymbol{\nu} - \boldsymbol{\lambda}) \geq 0, \quad \forall \boldsymbol{\nu};\ \psi(\boldsymbol{\nu}) > \psi(\boldsymbol{\lambda})\} \tag{22}$$

of ψ at $\boldsymbol{\lambda}$.

(ii) $\psi(\boldsymbol{\lambda})$ is a quasi-concave function for $\boldsymbol{\lambda} \in \Lambda_{\rm s}$.

(iii) Λ_+ is a convex set.

(iv) If $\psi(\boldsymbol{\lambda})$ has stationary points in Λ_+, all of them are global optimal solutions in $\Lambda_{\rm s}$.

The proof is in the Appendix.

Note that the set of global solutions of ψ is convex because of the quasi-concavity of $\psi(\boldsymbol{\lambda})$. (iv) is similar to the property that concave functions have. Therefore, a simple subgradient method such as SLR, which searches for stationary points, works well. SLR is an algorithm for finding a stationary point in Λ_+, which Theorem 1 (iv) proves to be a global optimal solution in Λ_s.

Figures 1 and 2 are images of $\psi(\boldsymbol{\lambda})$ for $m = 2$, where $\boldsymbol{\lambda} \in \Lambda_{\rm s}$ is expressed by one variable $\alpha \in [0, 1]$ as $\boldsymbol{\lambda} = (\alpha, 1-\alpha)^\top$. The vertical axis shows $\psi(\boldsymbol{\lambda})$, and the horizontal one shows α for a randomly generated 2-QCQP. We can make sure that $\psi(\boldsymbol{\lambda})$ is a quasi-concave function from these figures.

There are subgradient methods for maximizing a quasi-concave function. If the objective function satisfies some assumptions, the convergence of the algorithms is guaranteed. However, this may not be the case for the problem setting of (18). For example, in [15], $\psi(\boldsymbol{\lambda})$ must satisfy the Hölder condition of order $p > 0$ with modulus $\mu > 0$, that is,

$$\psi(\boldsymbol{\lambda}) - \psi_* \leq \mu(\mathrm{dist}(\boldsymbol{\lambda}, \Lambda^*))^p, \quad \forall \boldsymbol{\lambda} \in \mathbb{R}^m,$$

where ψ_* is the optimal value, Λ^* is the set of optimal solutions and $\mathrm{dist}(\boldsymbol{y}, Y)$ denotes the Euclidean distance from a vector $\boldsymbol{y}$ to a set Y. It is hard to check whether $\psi(\boldsymbol{\lambda})$ satisfies the Hölder condition. Ensuring the convergence of SLR seems difficult, but numerical experiments imply that SLR works well and often obtains the optimal value same as SDP relaxation can.

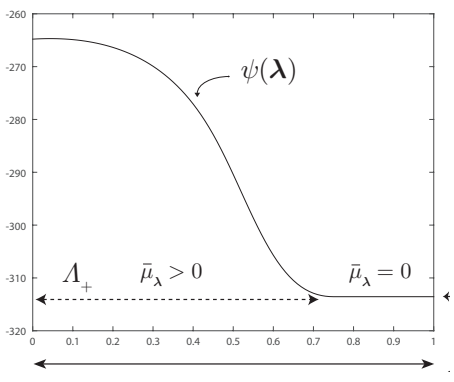

Figure 1: $\psi(\boldsymbol{\lambda})$ and Λ_+ when $Q_0 \succeq O$

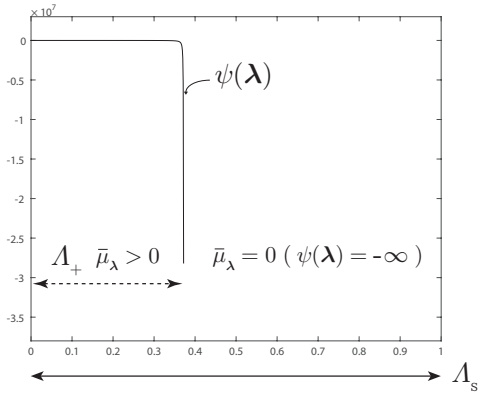

Figure 2: $\psi(\boldsymbol{\lambda})$ and Λ_+ when $Q_0 \not\succeq O$

4 Details of the Algorithm

4.1 1-QCQP as a Subproblem

SLR needs to solve 1-QCQP ($\mathrm{P_k}$). Here, we describe a fast and exact solution method for a general 1-QCQP. First, we transform the original 1-QCQP by using a new variable t into the form:

$$\begin{array}{ll} \min\limits_{\boldsymbol{x},t} & t \\ \text{s.t.} & \boldsymbol{x}^\top Q_0 \boldsymbol{x} + 2\boldsymbol{q}_0^\top \boldsymbol{x} + \gamma_0 \le t, \\ & \boldsymbol{x}^\top Q_{\boldsymbol{\lambda}} \boldsymbol{x} + 2\boldsymbol{q}_{\boldsymbol{\lambda}}^\top \boldsymbol{x} + \gamma_{\boldsymbol{\lambda}} \le 0, \end{array} \tag{23}$$

where

$$Q_{\boldsymbol{\lambda}} = \sum_{i=1}^m \lambda_i^{(k)} Q_i, \quad \boldsymbol{q}_{\boldsymbol{\lambda}} = \sum_{i=1}^m \lambda_i^{(k)} \boldsymbol{q}_i, \quad \gamma_{\boldsymbol{\lambda}} = \sum_{i=1}^m \lambda_i^{(k)} \gamma_i,$$

in the SLR algorithm. Here, we assume that (23) satisfies the following Primal and Dual Slater conditions:

$$\begin{array}{rlll} \text{(Primal Slater condition)} & \exists \boldsymbol{x} \in \mathbb{R}^n & \text{s.t.} & \boldsymbol{x}^\top Q_{\boldsymbol{\lambda}} \boldsymbol{x} + 2\boldsymbol{q}_{\boldsymbol{\lambda}}^\top \boldsymbol{x} + \gamma_{\boldsymbol{\lambda}} < 0 \\ \text{(Dual Slater condition)} & \exists \sigma \ge 0 & \text{s.t.} & Q_0 + \sigma Q_{\boldsymbol{\lambda}} \succ O \end{array}$$

Note that ($\mathrm{P_k}$) satisfies the Primal Slater condition because of Assumption 1 (a), and it also satisfies the Dual Slater condition because either Q_0 or $Q_{\boldsymbol{\lambda}}$ is positive definite by the updating rule of $\boldsymbol{\lambda}^{(k)}$ explained in the next subsection. Here, we define

$$S := \{\sigma \ge 0 \mid Q_0 + \sigma Q_{\boldsymbol{\lambda}} \succeq O\},$$

which is a convex set of one dimension, that is, an interval. The Dual Slater condition implies that S is not a point, and therefore, $\underline{\sigma} < \bar{\sigma}$ holds for

$$\bar{\sigma} := \sup_{\sigma \in S} \sigma, \tag{24}$$

$$\underline{\sigma} := \inf_{\sigma \in S} \sigma. \tag{25}$$

We set $\bar{\sigma} = +\infty$ when $Q_{\boldsymbol{\lambda}} \succ O$ and $\underline{\sigma} = 0$ when $Q_0 \succeq O$. For (23), we make the following relaxation problem using $\bar{\sigma}$ and $\underline{\sigma}$:

$$\begin{aligned} \min_{\boldsymbol{x},t} \quad & t \\ \text{s.t.} \quad & \boldsymbol{x}^\top (Q_0 + \underline{\sigma} Q_{\boldsymbol{\lambda}}) \boldsymbol{x} + 2(\boldsymbol{q}_0 + \underline{\sigma} \boldsymbol{q}_{\boldsymbol{\lambda}})^\top \boldsymbol{x} + \gamma_0 + \underline{\sigma} \gamma_{\boldsymbol{\lambda}} \leq t, \\ & \boldsymbol{x}^\top (Q_0 + \bar{\sigma} Q_{\boldsymbol{\lambda}}) \boldsymbol{x} + 2(\boldsymbol{q}_0 + \bar{\sigma} \boldsymbol{q}_{\boldsymbol{\lambda}})^\top \boldsymbol{x} + \gamma_0 + \bar{\sigma} \gamma_{\boldsymbol{\lambda}} \leq t. \end{aligned} \tag{26}$$

Note that in the SLR algorithm, we keep either Q_0 or $Q_{\boldsymbol{\lambda}}$ positive semidefinite. When $\underline{\sigma} = 0$, (26) is equivalent to the following relaxed problem:

$$\begin{aligned} \min_{\boldsymbol{x},t} \quad & t \\ \text{s.t.} \quad & \boldsymbol{x}^\top Q_0 \boldsymbol{x} + 2\boldsymbol{q}_0^\top \boldsymbol{x} + \gamma_0 \leq t, \\ & \boldsymbol{x}^\top (\hat{\sigma} Q_0 + Q_{\boldsymbol{\lambda}}) \boldsymbol{x} + 2(\hat{\sigma} \boldsymbol{q}_0 + \boldsymbol{q}_{\boldsymbol{\lambda}})^\top \boldsymbol{x} + (\hat{\sigma} \gamma_0 + \gamma_{\boldsymbol{\lambda}}) \leq \hat{\sigma} t, \end{aligned} \tag{27}$$

where $\hat{\sigma} = 1/\bar{\sigma}$, and $\hat{\sigma}$ can be easily calculated. On the other hand, when $\bar{\sigma} = +\infty$, (26) is equivalent to

$$\begin{aligned} \min_{\boldsymbol{x},t} \quad & t \\ \text{s.t.} \quad & \boldsymbol{x}^\top (Q_0 + \underline{\sigma} Q_{\boldsymbol{\lambda}}) \boldsymbol{x} + 2(\boldsymbol{q}_0 + \underline{\sigma} \boldsymbol{q}_{\boldsymbol{\lambda}})^\top \boldsymbol{x} + \gamma_0 + \underline{\sigma} \gamma_{\boldsymbol{\lambda}} \leq t, \\ & \boldsymbol{x}^\top Q_{\boldsymbol{\lambda}} \boldsymbol{x} + 2\boldsymbol{q}_{\boldsymbol{\lambda}}^\top \boldsymbol{x} + \gamma_{\boldsymbol{\lambda}} \leq 0. \end{aligned} \tag{28}$$

(28) can be viewed as dividing the second constraint of (26) by $\bar{\sigma}$ and $\bar{\sigma} \to \infty$.

The following theorem shows the equivalence of the proposed relaxation problem (26) and the original problem (23).

Theorem 2. *Under the Primal and Dual Slater conditions, the feasible region* Δ_{rel} *of the proposed relaxation problem* (26) *is the convex hull of the feasible region* Δ *of the original problem* (23)*, i.e.,* $\Delta_{\mathrm{rel}} = \mathrm{conv}(\Delta)$*.*

The proof is in the Appendix.

Theorem 2 implies that (26) gives an exact optimal solution of 1-QCQP (23) since the objective function is linear. The outline of the proof is as follows (see Figure 3). We choose an arbitrary point $(\boldsymbol{x}^*, t^*)$ in Δ_{rel} and show that there exists two points P and Q in Δ which express $(\boldsymbol{x}^*, t^*)$ as a convex combination of P and Q. We show in the Appendix how to obtain P and Q for an arbitrary point in Δ_{rel}. Using this technique, we can find an

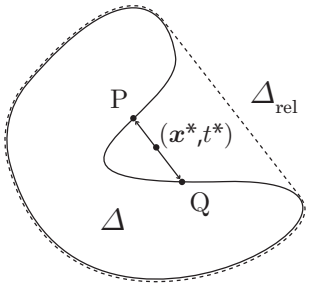

Figure 3: Image of Δ and $\Delta_{\rm rel}$

optimal solution of 1-QCQP (23). By comparison, SDP relaxation can not always find a feasible solution for the 1-QCQP (though it can obtain the optimal value).

(26) is a convex quadratic problem equivalent to 1-QCQP, which we will call **CQ1**. Note that CQ1 has only two constraints, and we can solve it very quickly. CQ1 can be constructed for general 1-QCQPs, including the Trust Region Subproblem (TRS). The numerical experiments in Section 5 imply that CQ1 can be solved by a convex quadratic optimization solver faster than by an efficient method for solving a TRS and hence that CQ1 can speed up SLR.

Now let us explain how to calculate $\hat{\sigma}$ or $\underline{\sigma}$ especially when either Q_0 or Q_λ is positive definite. We explain how to convexify a matrix which has some negative eigenvalues by using a positive definite matrix, i.e., for both cases when $Q_0 \succ O$ in (27) and $Q_\lambda \succ O$ in (28). First, we calculate $\hat{\sigma}$ in (27) when $Q_0 \succ O$. Let $Q_0^{\frac{1}{2}}$ be the square root of the matrix Q_0 and $Q_0^{-\frac{1}{2}}$ be its inverse. Then,

$$\begin{aligned} \sigma Q_0 + Q_\lambda \succeq O &\iff Q_0^{-\frac{1}{2}}(\sigma Q_0 + Q_\lambda)Q_0^{-\frac{1}{2}} \succeq O \\ &\iff \sigma I + Q_0^{-\frac{1}{2}} Q_\lambda Q_0^{-\frac{1}{2}} \succeq O \end{aligned}$$

holds. Therefore, $\hat{\sigma}$ can be calculated as

$$\hat{\sigma} = |\min\{\sigma_{\min}(Q_0^{-\frac{1}{2}} Q_\lambda Q_0^{-\frac{1}{2}}), 0\}|,$$

where $\sigma_{\min}(X)$ is the minimum eigenvalue of X. Similarly, we can calculate $\underline{\sigma}$ in (28) as

$$\underline{\sigma} = |\min\{\sigma_{\min}(Q_\lambda^{-\frac{1}{2}} Q_0 Q_\lambda^{-\frac{1}{2}}), 0\}|,$$

when $Q_\lambda \succ O$.

It is true that (26) gives us the exact optimal value of (23). When both Q_0 and Q_λ have zero (or even negative) eigenvalues, $\hat{\sigma}$ and $\underline{\sigma}$ can not be

computed, but such cases can be ignored because an 1-QCQP (23) with such Q_0 and $Q_{\boldsymbol{\lambda}}$ does not give an optimal solution of (18). Therefore, in the **Algorithm 1**, we keep either Q_0 or $\sum_{i=1}^m \sigma_i Q_i$ positive definite.

4.2 Update Rule of $\boldsymbol{\lambda}$

Now let us explain the update rule of $\boldsymbol{\lambda}$, which is shown in **Step 2** of **Algorithm 1**. The update rule is constructed in a similar way to the gradient projection method for solving (18). **Step 4** is needed only when $Q_0 \not\succ O$ and $\sum_{i=1}^m \lambda_i^{(k+1)} Q_i \not\succ O$ hold. We update the Lagrange multipliers corresponding to convex constraints, whose index set is defined as $C := \{i \mid 1 \leq i \leq m,\ Q_i \succ O\}$. When $Q_0 \not\succ O$, Assumption 1 (b) assures that C is non-empty.

Algorithm 2 Update rule of $\boldsymbol{\lambda}$

Given a sufficiently small positive scalar δ,

Step 1: Calculate the gradient vector $\boldsymbol{g}^{(k)}$ as $g_i^{(k)} = {\boldsymbol{x}^{(k)}}^\top Q_i \boldsymbol{x}^{(k)} + 2\boldsymbol{q}_i^\top \boldsymbol{x}^{(k)} + \gamma_i$.

Step 2: Normalize $\boldsymbol{g}^{(k)}$ as $\boldsymbol{g}^{(k)} \leftarrow \frac{\boldsymbol{g}^{(k)}}{|\boldsymbol{e}^\top \boldsymbol{\lambda}|}$ and set the step size h. Update $\boldsymbol{\lambda}^{(k+1)}$ as

$$\boldsymbol{\lambda}^{(k+1)} = \mathbf{proj}_{\Lambda_{\mathrm{s}}}(\boldsymbol{\lambda}^{(k)} + h\boldsymbol{g}^{(k)}), \tag{29}$$

where $\mathbf{proj}_{\Lambda_{\mathrm{s}}}(\boldsymbol{a}) := \underset{\boldsymbol{b} \in \Lambda_{\mathrm{s}}}{\arg\min} \|\boldsymbol{a} - \boldsymbol{b}\|_2$ is the projection onto Λ_{s}.

Step 3: If $Q_0 \succ O$ or $\sum_{i=1}^m \lambda_i^{(k+1)} Q_i \succ O$, terminate and return $\boldsymbol{\lambda}^{(k+1)}$.

Step 4: Otherwise, find a minimum positive scalar α such that $\alpha \sum_{i \in C} Q_i + \sum_{i=1}^m \lambda_i^{(k+1)} Q_i \succeq O$ and update $\lambda_i^{(k+1)} \leftarrow \lambda_i^{(k+1)} + \alpha + \delta$ for $i \in C$. After computing

$$\boldsymbol{\lambda}^{(k+1)} \leftarrow \frac{1}{\sum_{i=1}^m \lambda_i^{(k+1)}} \boldsymbol{\lambda}^{(k+1)},$$

terminate and return $\boldsymbol{\lambda}^{(k+1)}$.

Theorem 1 (i) shows that a subgradient vector of $\psi(\boldsymbol{\lambda})$ at $\boldsymbol{\lambda}^{(k)}$ is

$$\tilde{\boldsymbol{g}}_{\boldsymbol{\lambda}^{(k)}} = \bar{\mu}_{\boldsymbol{\lambda}^{(k)}} \boldsymbol{g}^{(k)},$$

where $g_i^{(k)} = {\boldsymbol{x}^{(k)}}^\top Q_i \boldsymbol{x}^{(k)} + 2\boldsymbol{q}_i^\top \boldsymbol{x}^{(k)} + \gamma_i,\ \forall i$. To find a larger function value of $\psi(\boldsymbol{\lambda})$ at the kth iteration, we use $\boldsymbol{g}^{(k)}$ as the subgradient vector

of $\psi(\boldsymbol{\lambda})$ rather than $\tilde{\boldsymbol{g}}_{\boldsymbol{\lambda}^{(k)}}$ for the following reasons. When $\bar{\mu}_{\boldsymbol{\lambda}^{(k)}} > 0$, we can use $\boldsymbol{g}^{(k)}$ as a subgradient vector of $\psi(\boldsymbol{\lambda})$ at $\boldsymbol{\lambda}^{(k)}$. When $\bar{\mu}_{\boldsymbol{\lambda}^{(k)}} = 0$ (i.e. $\boldsymbol{\lambda}^{(k)} \notin \Lambda_+$), Q_0 should be positive semidefinite because of the constraint $Q_0 + \mu \sum_{i=1}^m \lambda_i Q_i \succeq O$, and the optimal value of (17) equals that of (14) ($= \phi(\mathbf{0})$). In this case, $\phi(\mathbf{0})$ is the smallest possible value, but it is not the optimal one of the original problem (1) because an optimal solution of the unconstrained problem (14), $\bar{\boldsymbol{x}}$, is not in the feasible region of (1), from Assumption 1 (c). Therefore, when $\bar{\mu}_{\boldsymbol{\lambda}^{(k)}} = 0$, the algorithm needs to move $\boldsymbol{\lambda}^{(k)}$ toward Λ_+; precisely, $\boldsymbol{\lambda}^{(k)}$ is moved in the direction of $\boldsymbol{g}^{(k)}$, although $\tilde{\boldsymbol{g}}_{\boldsymbol{\lambda}^{(k)}}$ is the zero vector. It can be easily confirmed that by moving $\boldsymbol{\lambda}^{(k)}$ sufficiently far in this direction, the left side of the constraint of (P_k) becomes positive and $\bar{\boldsymbol{x}}$ moves out of the feasible region of ($\mathrm{P_k}$).

The whole algorithm updates $\boldsymbol{\lambda}^{(k)}$ and $\boldsymbol{x}^{(k)}$, $k = 1, 2, \ldots$. In order for ($\mathrm{P_k}$) to have an optimal solution $\boldsymbol{x}^{(k)}$, $\boldsymbol{\lambda}^{(k)}$ needs to be set appropriately so as to satisfy $Q_0 + \mu \sum_{i=1}^m \lambda_i^{(k)} Q_i \succeq O$ for some $\mu \geq 0$. If the input Q_0 of the given problem satisfies $Q_0 \succeq O$, then $Q_0 + \mu \sum_{i=1}^m \lambda_i^{(k)} Q_i \succeq O$ holds with $\mu = 0$, which makes ($\mathrm{P_k}$) bounded.

On the other hand, in the case of $Q_0 \not\succeq O$, the optimal value of ($\mathrm{P_k}$), $\psi(\boldsymbol{\lambda}^{(k)})$, possibly becomes $-\infty$, and we can not find an optimal solution. In such case, we can not calculate $\boldsymbol{g}^{(k)}$ and the algorithm stops. To prevent this from happening, we define a subset of Λ_s so that the optimal value does not become $-\infty$. When $Q_0 \not\succeq O$, Λ_+ can be rewritten as

$$\Lambda_+ = \{\boldsymbol{\lambda} \geq \mathbf{0} \mid \boldsymbol{e}^\top \boldsymbol{\lambda} = 1,\ \exists \mu \geq 0;\ Q_0 + \mu \sum_{i=1}^m \lambda_i Q_i \succeq O\}.$$

Λ_+ is the set of $\boldsymbol{\lambda}$ for which $\psi(\boldsymbol{\lambda}) > -\infty$. However, the above description of Λ_+ is complicated because of the positive semidefinite constraint. Furthermore, CQ1 requires that either Q_0 or $\sum_{i=1}^m \lambda_i Q_i$ be positive definite. Therefore, when $Q_0 \not\succ O$, we approximate Λ_+ as

$$\Lambda'_+ := \{\boldsymbol{\lambda} \geq \mathbf{0} \mid \boldsymbol{e}^\top \boldsymbol{\lambda} = 1,\ \sum_{i=1}^m \lambda_i Q_i \succ O\},$$

and keep $\boldsymbol{\lambda}^{(k)}$ in Λ'_+ by **Step 4**. It can be easily confirmed that Λ'_+ is a convex set. By replacing the feasible region Λ_s of (18) by Λ'_+ ($\subseteq \Lambda_+$), the relaxation can be weaker and the optimal value is not necessarily equal to the SDP relaxation value. Thus, when $Q_0 \not\succ O$, SLR may be worse than it is when $Q_0 \succ O$.

Now let us explain how to choose the step size h. Gradient methods have various rules to determine an appropriate step size. Simple ones include a constant step size $h = c$ or a diminishing step size (e.g. $h = c/\sqrt{k}$), where k is the number of iterations (e.g., see [8]). A more complicated one is the backtracking line search (e.g., see [4, 21]). Although the backtracking line

search has been shown to perform well in many cases, we use a **diminishing step size** $h = c/\sqrt{k}$ to save the computation time of SLR. The point of SLR is to obtain a relaxed solution quickly, so we should choose the simpler way.

In **Step 2**, we compute $\boldsymbol{\lambda}^{(k+1)}$ by using $\boldsymbol{g}^{(k)}$ and h by using (29). We can easily compute the projection onto $\Lambda_{\rm s}$ by using the method proposed by Chen and Ye [10]. Here, the condition: $\exists \mu \geq 0;\ Q_0 + \mu \sum_{i=1}^m \lambda_i Q_i \succ O$ in Λ_+ is ignored in the projection operation, but when $Q_0 \succ O$, the resulting projected point $\boldsymbol{\lambda}^{(k+1)}$ is in Λ_+. On the other hand, when $Q_0 \not\succ O$, the vector $\boldsymbol{\lambda}^{(k+1)}$ is not necessarily in Λ_+ or Λ'_+. In such case, $\boldsymbol{\lambda}^{(k+1)}$ is modified in **Step 4** so as to belong to Λ'_+. **Step 4** is a heuristic step; it is needed to keep $\boldsymbol{\lambda} \in \Lambda'_+$ when $Q_0 \not\succ O$.

4.3 Setting the Initial Point

The number of iterations of SLR depends on how we choose the initial point. In this section, we propose two strategies for choosing it. Note that at an optimal solution $\boldsymbol{\lambda}$, all elements λ_i corresponding to convex constraints with $Q_i \succ O$, $i \in C$, are expected to have positive weights. Hence, we will give positive weights only for λ_i, $i \in C$ (if it exists).

Here, we assume that $(Q_i, \boldsymbol{q}_i, \gamma_i)$ in each constraint is appropriately scaled by a positive scalar as follows. When the matrix Q_i has positive eigenvalues, $(Q_i, \boldsymbol{q}_i, \gamma_i)$ is scaled so that the minimum positive eigenvalue of Q_i is equal to one. If Q_i has no positive eigenvalues, it is scaled such that the maximum negative eigenvalue is equal to -1.

The first approach is "equal" weights. It gives equal weights to $\lambda_i^{(0)}$ s.t. $Q_i \succ O$ or if there are no $Q_i \succ O$ (which implies that $Q_0 \succ O$), it gives equal weights to all $\lambda_i^{(0)}$ as follows:

Equal weights rule If the index set of convex constraints C is nonempty, we define $\boldsymbol{\lambda}^{(0)}$ as

$$\lambda_i^{(0)} = \begin{cases} \frac{1}{|C|}, & \text{if } Q_i \succ O, \\ 0, & \text{otherwise.} \end{cases} \tag{30}$$

If $C = \emptyset$, we define $\boldsymbol{\lambda}^{(0)}$ as

$$\lambda_i^{(0)} = \frac{1}{m}, \quad i = 1, \ldots, m. \tag{31}$$

The second approach uses the idea of the Schur complement. Note that this rule only applies when there are some i (≥ 1) such that $Q_i \succ O$. For the constraint with $Q_i \succ O$, we have

$$\begin{aligned} & \boldsymbol{x}^\top Q_i \boldsymbol{x} + 2\boldsymbol{q}_i^\top \boldsymbol{x} + \gamma_i \leq 0 \\ \Longleftrightarrow\ & (\boldsymbol{x} + Q_i^{-1}\boldsymbol{q})^\top Q_i (\boldsymbol{x} + Q_i^{-1}\boldsymbol{q}_i) \leq \boldsymbol{q}_i^\top Q^{-1} \boldsymbol{q}_i - \gamma_i. \end{aligned}$$

The right-hand side $\eta_i := \boldsymbol{q}_i^\top Q^{-1}\boldsymbol{q}_i - \gamma_i$ can be considered the volume of the ellipsoid. From Assumption 1 (a), the ellipsoid has positive volume and we have $\eta_i > 0$. A numerical experiment shows that constraints having small positive η_i tend to become active in SDP relaxation. Therefore, it seems reasonable to give large weights to constraints whose η_i (> 0) is small. On the other hand, since we treat the constraint as

$$\begin{pmatrix} 1 \\ \boldsymbol{x} \end{pmatrix}^\top \begin{pmatrix} \gamma_i & \boldsymbol{q}_i^\top \\ \boldsymbol{q}_i & Q_i \end{pmatrix} \begin{pmatrix} 1 \\ \boldsymbol{x} \end{pmatrix} \leq 0,$$

the value $-\eta_i$ can be viewed as the Schur complement of $\begin{pmatrix} \gamma_i & \boldsymbol{q}_i^\top \\ \boldsymbol{q}_i & Q_i \end{pmatrix}$. It is known that when $Q_i \succ O$, $\begin{pmatrix} \gamma_i & \boldsymbol{q}_i^\top \\ \boldsymbol{q}_i & Q_i \end{pmatrix} \succeq O$ is equivalent to $-\eta_i \geq 0$. However, in this case, $\begin{pmatrix} \gamma_i & \boldsymbol{q}_i^\top \\ \boldsymbol{q}_i & Q_i \end{pmatrix} \succeq O$ does not hold since $\eta_i > 0$. But we consider this value to be an indicator of convexity. We give large weights for the constraints whose Schur complement $-\eta_i$ is large. Then, since $\eta_i > 0$, we give large weights for the constraints whose $-\eta_i$ (< 0) are close to zero; that is, $\frac{1}{|\eta_i|}$ are large. Here, we consider the following rule:

Schur complement rule For $i \in C$, calculate $s_i := 1/|\eta_i|$. We define $\boldsymbol{\lambda}^{(0)}$ as

$$\lambda_i^{(0)} = \begin{cases} \frac{s_i}{\sum_{i=1}^m s_i} & \text{if } Q_i \succ O, \\ 0, & \text{otherwise.} \end{cases} \tag{32}$$

Although the Schur complement rule also has no theoretical guarantee, numerical experiments show their usefulness especially when $Q_0 \not\succ O$.

4.4 RQT Constraints

We may be able to find a better optimal value of the SDP relaxation problem (2) by adding a redundant convex quadratic constraint constructed similarly to RLT (this is discussed in Section 2.2) to (1) when there are box constraints and by applying SLR to the resulting QCQP. Since (9) holds for $1 \leq i = j \leq n$, we have

$$x_i^2 - (u_i + l_i)x_i + u_i l_i \leq 0, \qquad i = 1, \cdots, n. \tag{33}$$

The summation of (33) for $i = 1, \cdots, n$ leads to

$$\boldsymbol{x}^\top \boldsymbol{x} - (\boldsymbol{u} + \boldsymbol{l})^\top \boldsymbol{x} + \boldsymbol{u}^\top \boldsymbol{l} \leq 0. \tag{34}$$

We call this method the reformulation quadraticization technique (RQT). Since (34) is a convex quadratic constraint, it may be effective for the SLR

relaxation tighter. The numerical experiments in Section 5 show that by adding (34), we could get a tighter optimal value in some cases than SDP relaxations.

There are other ways of making new convex quadratic constraints. Furthermore, even nonconvex constraints (like (8) or (10)) are possibly effective for tightening SLR. However, in this study, we only considered (34) to save computation time.

5 Numerical Experiments

We implemented SLR, SDP relaxation, and Block-SDP (in Section 2.3) and compared their results. In [9], there are no rules to decide the number of blocks r of Block-SDP. In our experiments, we tried several values of r and chose $r := 0.05 \times n$, which seemed to work well.

We used MATLAB Ver. 8.4.0 (R2014b) for all the numerical experiments. We solved the SDP relaxation and Block-SDP by using SeDuMi 1.3 [32]. To solve the convex quadratic optimization problems, 1-QCQP (27) and (28) in the SLR algorithm, we used CPLEX Ver. 12.5. We used a computer with a 2.4 GHz CPU and 16GB RAM.

5.1 Random m-QCQP

First, we checked the performance of SLR for random m-QCQP generated in the way that Zheng Sun and Li [31] did. In Sections 5.1.1∼5.1.4, we consider problems without box constraints; we compare SLR (or CQ1) and SDP relaxation. In Section 5.1.5, we consider problems including box constraints; we compare SLR, SDP relaxation, and Block-SDP.

5.1.1 Tolerance ϵ and Computation Time

We now investigate the computation times of SLR for given tolerance values ϵ. We randomly generated 30 instances of a 10-QCQP, whose problem sizes were $n = 30$ and $m = 10$. Among the $m = 10$ constraints, there were five convex ones. The objective functions of all instances were strictly convex, i.e., $Q_0 \succ O$.

The relationship between the tolerance ϵ and the computation time is shown in Figure 4. The smaller ϵ becomes, the longer the computation takes. In this setting, SLR can solve the 10-QCQP faster than SDP relaxation can when $\epsilon > 10^{-4}$. Hence, we set $\epsilon = 10^{-4}$ in what follows.

5.1.2 Effect of Initial Points

We compared the two strategies for choosing the initial points (30) (or (31)) and (32) and checked the results for $Q_0 \succ O$ and $Q_0 \not\succ O$. We only show

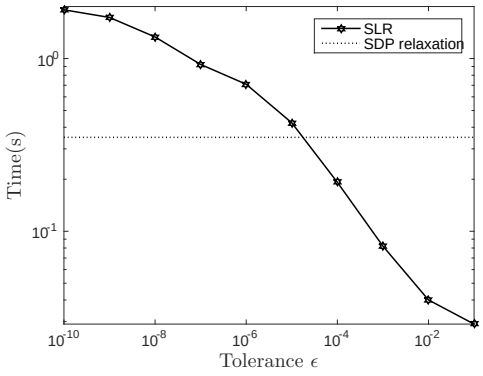

Figure 4: Average Computation Time versus Tolerance ϵ

results for $Q_0 \not\succ O$ because both initial point rules gave almost the same results when $Q_0 \succ O$. We randomly generated 30 instances for each setting, where $n = 100$ and $m = 10$, and varied the number of convex constraints from $|C| = 1$ to 9.

Note that SLR is not stronger than SDP relaxation and we do not know the exact optimal value of each random m-QCQP. Therefore, we checked the performance of SLR by comparing its value with the optimal value of SDP relaxation. Here, we used the error ratio defined as

$$\text{Ratio} := \left| \frac{\text{OPT}_{\text{SLR}}}{\text{OPT}_{\text{SDPrelax}}} \right|.$$

This indicator was used in all of the experiments described below. It is greater than or equal to one since SLR is not stronger than SDP relaxation. The performance of SLR is said to be good when the ratio is close to one.

Figures 5 and 6 plot the number of iterations and the error ratio versus the number of convex constraints. When there is only one convex constraint, an optimal solution $\boldsymbol{\lambda}$ for $\psi(\boldsymbol{\lambda})$ usually has only one positive element corresponding to the convex constraint and all the other elements are zero. In this case, SLR needs only few iterations. When $Q_0 \not\succ O$, the Schur complement rule works well in terms of computation time and the error ratio as the number of convex constraints increases. This may be because an optimal solution of SDP relaxation has many non-zero components and the equal weights rule can not represent each weight appropriately. On the basis of the above considerations, we decided to use the Schur complement rule in the remaining experiments.

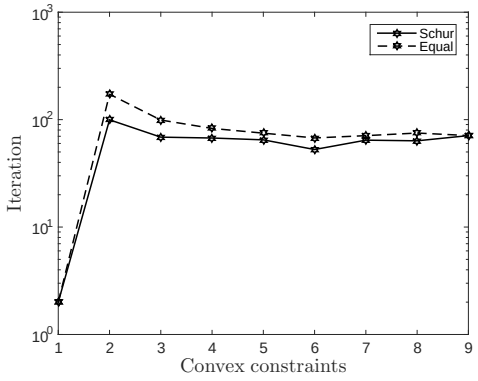

Figure 5: Average Number of Iterations ($Q_0 \not\succ O$)

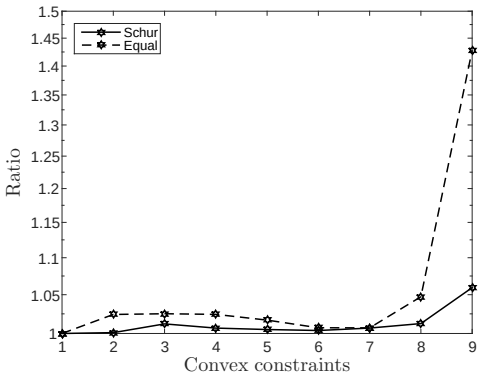

Figure 6: Average Error Ratio ($Q_0 \not\succ O$)

5.1.3 Effect of the Number of Variables n on the Computation Time and Error

We checked the computation time of SLR by changing the number of variables n. Note that SLR works better when $Q_0 \succ O$ than when $Q_0 \not\succ O$ because we have to approximate the feasible region Λ_+ when $Q_0 \not\succ O$. In this experiment, n was varied from 25 to 5000, and we set $m = 15$, of which 8 constraints were convex. We generated 30 instances when $n \leq 250$, ten instances when $250 < n \leq 1000$, and one instance when $n \geq 2500$. In this experiment, we set $\epsilon = 1.0^{-3}$ because large problems take a very long time to solve.

Case 1. $Q_0 \succ O$. SLR performed well when $Q_0 \succ O$ (Figures 7 and 8). The computation time was almost one order of magnitude smaller than that of SDP relaxation, and the error ratio was less than 1.06. There were many instances which SLR can obtain the optimal value same as SDP relaxation can. Furthermore, SLR was able to solve problems that SDP relaxation could not because it ran out of memory.

Case 2. $Q_0 \not\succ O$. We replaced the objective function of each instance used in Case 1 by a nonconvex quadratic function and conducted the same experiments in each case. Figures 9 and 10 show the results for $Q_0 \not\succ O$. The performance of SLR deteriorated, but it was still faster than SDP relaxation and the error ratio was about 1.1. Note that we conducted only one experiment on $n = 2500, 5000$ to shorten the time of the experiment.

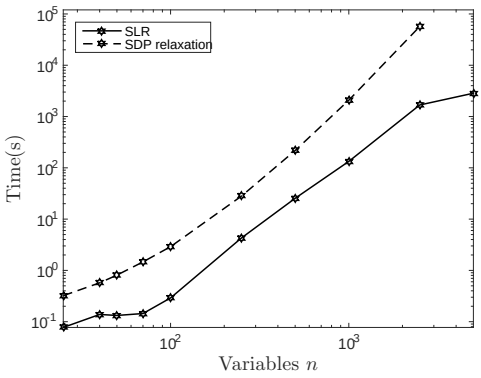

Figure 7: Average Computation Time for n ($Q_0 \succ O$)

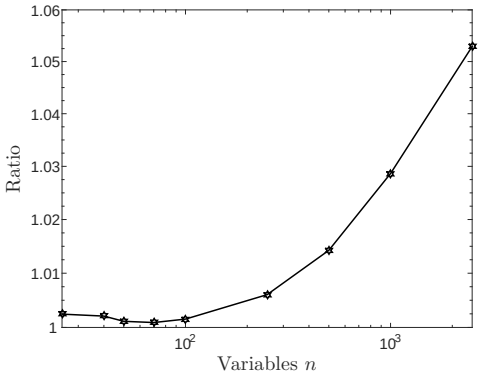

Figure 8: Average Error Ratio for n ($Q_0 \succ O$)

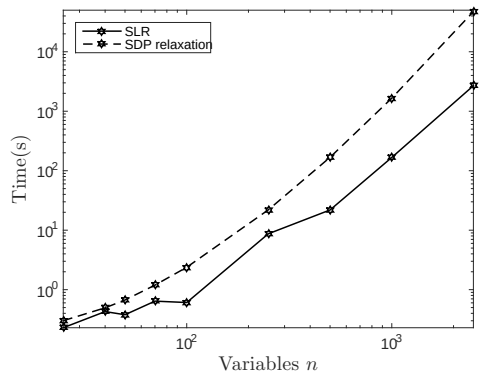

Figure 9: Average Computation Time for n ($Q_0 \not\succ O$)

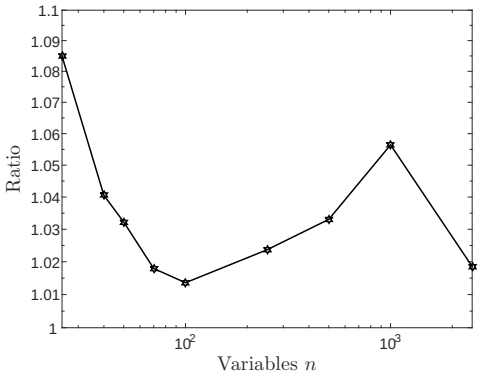

Figure 10: Average Error Ratio for n ($Q_0 \not\succ O$)

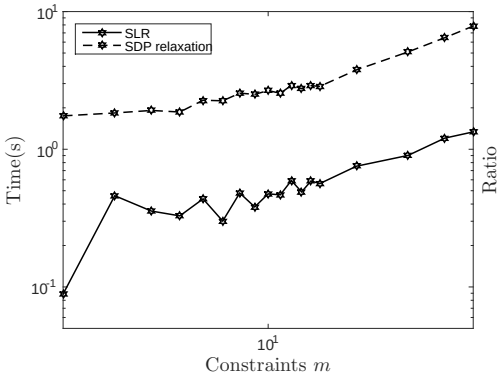

Figure 11: Average Computation Time for m ($Q_0 \succ O$)

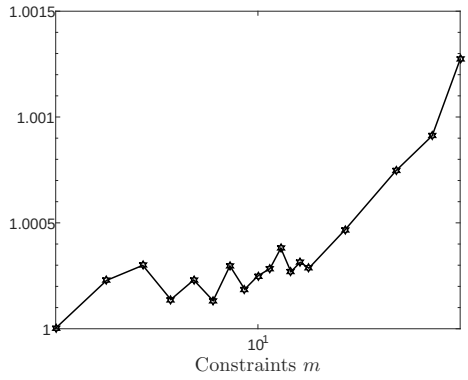

Figure 12: Average Error Ratio for m ($Q_0 \succ O$)

5.1.4 Effect of the Number of Constraints m on the Computation Time and Error

We checked the computation time of SLR by varying the number of constraints m. For $n = 100$, the number of constraints m was varied from 2 to 50. Half of the constraints (i.e. ceil$(m/2)$) were convex. We generated 30 instances for each setting.

Case 1. $Q_0 \succ O$. Figures 11 and 12 show the results. As a whole, the error ratios were less than 1.0015, and the computation time was about one order of magnitude smaller than that of SDP relaxation.

Case 2. $Q_0 \not\succ O$. Figures 13 and 14 show the results. SLR took longer than in Case 1, and the error ratio was about 1.06. SLR performed worse when $Q_0 \not\succ O$ because we approximated the feasible region.

5.1.5 RQT Constraints

We randomly generated problems with box constraints and added the RQT constraint proposed in Section 4.4 to the problems. n was varied from 30 to 500, and we set $m = 0.3n$, including ceil$(m/2)$ convex constraints. We generated 30 instances when $n \leq 100$ and 10 instances when $n > 100$. We added box constraints $\forall i;\ -1 \leq x_i \leq 1$ to all the instances. The following results are only for the case in which the objective function is nonconvex. When $Q_0 \succ O$, the RQT constraint did not affect the performance of our method by much.

The results are shown in Figures 15 and 16. In Figure 16, the ratio is less than one. This implies that SLR can get better optimal values than

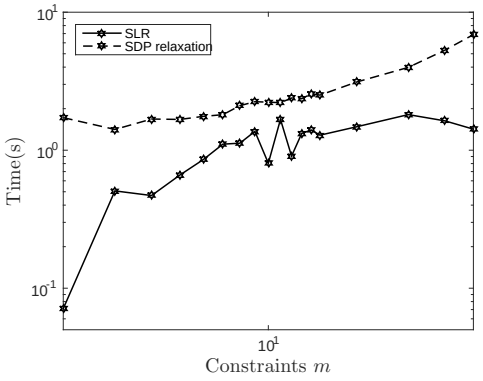

Figure 13: Average Computation Time for m ($Q_0 \not\succ O$)

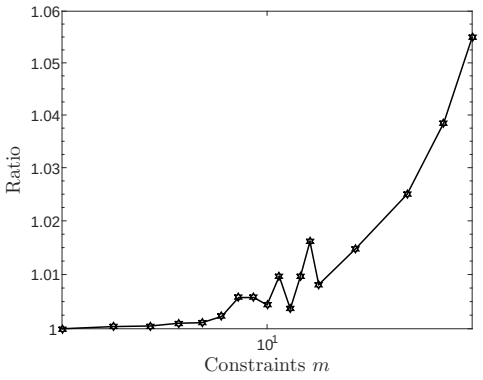

Figure 14: Average Error Ratio for m ($Q_0 \not\succ O$)

SDP relaxation can by adding RQT constraints. SLR is thus "stronger and faster" than SDP relaxation. In this sense, Block-SDP is similar to SLR. The performance of Block-SDP depends on the number of blocks r, but in this setting, SLR is faster than Block-SDP, although its error ratio is worse than that of Block-SDP.

5.2 1-QCQP

We checked the performance of CQ1 in solving the 1-QCQP of ($\mathrm{P_k}$). We compared CQ1, SDP relaxation, and an eigen-computation-based method for random 1-QCQPs. For a 1-QCQP with $Q_1 \succ O$, Adachi, Iwata, Nakatsukasa and Takeda [1] proposed an accurate and efficient method that solves a generalized eigenvalue problem only once. They called this method "GEP". We ran their MATLAB code for solving a 1-QCQP with $Q_1 \succ O$. Note that all of the methods obtained the exact optimal value of 1-QCQP. The computation time was plotted versus n. As described in Section 5.1, we generated the 1-QCQP in the way [31] did. Figures 17 and 18 are double logarithmic charts of n and the average computation time of 30 instances. Figure 17 shows that CQ1 is about one order of magnitude faster than SDP relaxation for all n. Figure 18 shows that CQ1 is faster than GEP when n is large. CQ1 or SLR is intended to be a "weaker, but faster" method than SDP relaxation, and such methods are useful for large-scale problems.

5.3 Max-Cut Problems

Max-cut problems [24] can be viewed as an application of 1-QCQP. A graph Laplacian matrix L can be obtained from a given undirected and weighted graph G. For the max-cut value for G, we solve the following nonconvex

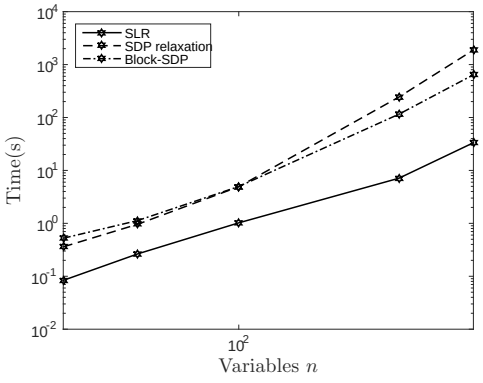

Figure 15: Average Computation Time for n (Additional RQT constraint)

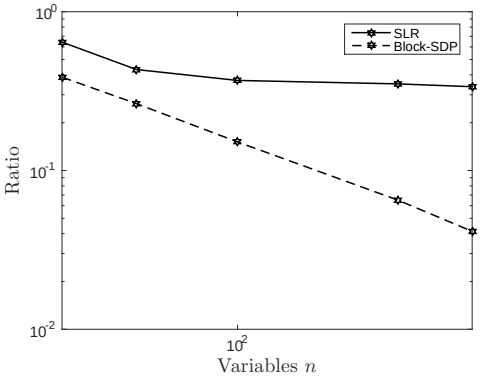

Figure 16: Average Error Ratio for n (Additional RQT constraint)

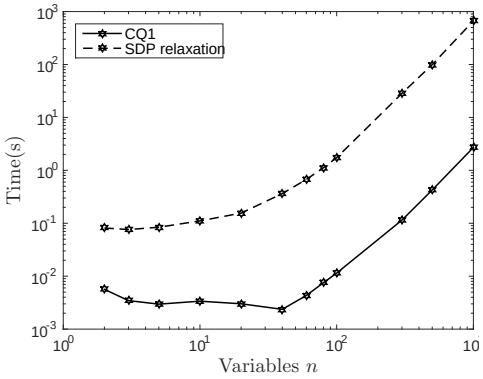

Figure 17: Average Computation Time for 1-QCQP in the case of $Q_0 \succ O$

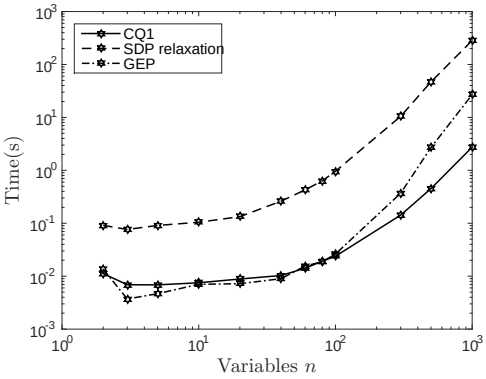

Figure 18: Average Computation Time for 1-QCQP in the case of $Q_1 \succ O$

$\{1, -1\}$ integer program:

$$\begin{array}{lll} \min\limits_{\boldsymbol{x}} & \boldsymbol{x}^\top L\boldsymbol{x} & \\ \text{s.t.} & x_i^2 = 1, & i = 1, \cdots, n. \end{array} \tag{35}$$

We relax (35) into

$$\begin{array}{lll} \min\limits_{\boldsymbol{x}} & \boldsymbol{x}^\top L\boldsymbol{x} & \\ \text{s.t.} & -1 \le x_i \le 1 & i = 1, \cdots, n, \end{array}$$

and then apply SDP relaxation and Block-SDP. For CQ1, we further relax the box constraints as follows:

$$\forall i; \;\; -1 \le x_i \le 1 \;\; \Longrightarrow \;\; \boldsymbol{x}^\top \boldsymbol{x} \le n,$$

because CQ1 needs at least one convex constraint. The resulting 1-QCQP is

$$\begin{array}{lll} \min\limits_{\boldsymbol{x}} & \boldsymbol{x}^\top L\boldsymbol{x} & \\ \text{s.t.} & \boldsymbol{x}^\top \boldsymbol{x} \le n. & \end{array} \tag{36}$$

Note that (36) can be regarded as a simple minimum eigenvalue problem. An optimal solution is an eigenvector corresponding to the minimum eigenvalue. However, our purpose is to check the computational result, and we use CQ1 for (36).

We solved max-cut instances from [24]. Many randomly generated instances are shown in [24], and the optimal values are known. The results are in Table 1. In this table, the "error" is defined as

$$\text{Error} := \left| \frac{\text{OPT}_{\text{method}} - \text{OPT}}{\text{OPT}} \right|,$$

where $\text{OPT}_{\text{method}}$ is the optimal value of each method and OPT is the exact optimal value. In [24], the names of the instances indicate how they were generated as well as the number of variables. For example, "g05_80", "80" means the number of variables, and "g05" means the density of edges and whether the weights of graph are all positive or include negative values. The details are given in [24] and there are ten instances for each kind of problem. In Table 1, "Time(s)" means the average time for ten instances, and the best methods among SDP relaxation, Block-SDP, and CQ1 in terms of either average computation time or average error are listed in bold.

Table 1 shows that CQ1 is "weaker" but "faster" than SDP relaxation. Block-SDP is weaker and even slower than SDP relaxation in these problem settings. CQ1 is much faster than SDP relaxation, so we can solve CQ1

Table 1: Time and Error for Max-cut

Method	SDP relaxation		Block-SDP		CQ1		Multiple CQ1	
Instance	Error	Time(s)	Error	Time(s)	Error	Time(s)	Error	Time(s)
g05_80	**0.02**	0.363	0.15	0.472	0.15	**0.022**	0.02	0.075
g05_100	**0.02**	0.504	0.13	0.545	0.14	**0.014**	0.01	0.093
pm1d_80	**0.17**	0.341	0.53	0.419	1.10	**0.012**	0.10	0.093
pm1d_100	**0.17**	0.487	0.52	0.504	1.01	**0.020**	0.09	0.114
pm1s_80	**0.15**	0.324	0.54	0.392	1.14	**0.013**	0.10	0.086
pm1s_100	**0.14**	0.490	0.50	0.491	1.08	**0.016**	0.12	0.107
pw01_100	**0.05**	0.478	0.19	0.550	0.62	**0.015**	0.04	0.099
pw05_100	**0.03**	0.509	0.13	0.579	0.17	**0.015**	0.02	0.096
pw09_100	**0.02**	0.498	0.14	0.611	0.09	**0.017**	0.01	0.103
w01_100	**0.13**	0.494	0.53	0.544	1.28	**0.013**	0.10	0.113
w05_100	**0.17**	0.482	0.51	0.556	0.92	**0.020**	0.10	0.114
w09_100	**0.17**	0.485	0.51	0.562	0.94	**0.015**	0.12	0.123

many times in the same period of time it takes to solve the SDP relaxation once. Accordingly, we tried to strengthen CQ1 by iterating it with a certain rounding rule as follows. An optimal solution of CQ1, $\bar{\boldsymbol{x}}$, satisfies $\bar{\boldsymbol{x}}^\top \bar{\boldsymbol{x}} = n$ because the objective function is nonconvex. Consequently, there exists a component of $\bar{\boldsymbol{x}}$ whose absolute value is more than one (otherwise, all the components are ± 1, and $\bar{\boldsymbol{x}}$ is an exact optimal solution for (35)). Then, we fix such a component as ± 1 and solve a small problem recursively. Note that if the objective function becomes positive (semi)definite by fixing some of the components and there exists no x_i whose absolute value is more than one, we set the component which has the maximum absolute value of all the components to 1 or -1. We perform this rounding until all the components are ± 1. Therefore, we have a feasible solution of the original problem (35) and obtain an upper bound of the original optimal value, while SDP relaxation, Block-SDP, and CQ1 find lower bounds. We call this rounding "Multiple CQ1" and show the results in the right-most column of Table 1. The results indicate that Multiple CQ1 is still "faster" than SDP relaxation. Such a faster method is useful when we want to solve a problem repeatedly.

6 Conclusions

In this paper, we proposed SLR, a new, fast convex relaxation for QCQP. SLR is a method for solving the Lagrangian dual problem of a given QCQP. There have been many studies on constructing Lagrangian dual problems

for nonconvex problems and reformulating them as semidefinite problems (SDPs). Instead of solving an SDP, our method divides the objective function of the Lagrangian dual problem into two parts and iteratively solves a 1-QCQP. We furthermore transform the 1-QCQP into a convex quadratic 1-QCQP called CQ1 whose feasible region forms the convex hull of the feasible region of the original 1-QCQP. Hence, we can obtain the exact optimal value of the 1-QCQP by solving CQ1. SDP relaxation can also solve the 1-QCQP exactly, but CQ1 is much faster. Numerical experiments confirmed this advantage of CQ1. CQ1 performed well for randomly generated 1-QCQP and max-cut problems.

In SLR, we successively solve a 1-QCQP with the Lagrange multiplier $\boldsymbol{\lambda}$ updated using a gradient method. We proved that the objective function $\psi(\boldsymbol{\lambda})$ is quasi-concave and has the good property that all the stationary points in Λ_+ are global optimal solutions, and thus, simple gradient methods work well. SLR is a faster relaxation compared with the interior point method for SDP relaxation for large n and m. Furthermore, by adding a new valid RQT constraint, we could obtain even a better optimal value than SDP relaxation could for some m-QCQP instances.

Our method can be regarded as a subgradient method that is applied to a quasi-concave problem induced from the Lagrangian dual of an m-QCQP. To ensure convergence, the quasi-concave problem must satisfy certain conditions, (e.g., in [15]) but unfortunately, it is not easy to check whether our quasi-concave problem satisfies the Hölder condition. In the future, we would like to investigate the global convergence of our algorithm.

When the objective function is nonconvex, we need to approximate the feasible region Λ_+ of the Lagrangian dual problem, and as a result, the SLR become worse in performance than that of solving m-QCQP with the convex objective function. We would like to improve the performance of SLR for instances having nonconvex objective functions.

References

[1] S. Adachi, S. Iwata, Y. Nakatsukasa and A. Takeda: Solving the trust region subproblem by a generalized eigenvalue problem. *SIAM Journal on Optimization*, 27:269–291, 2017.

[2] F. A. Al-Khayyal, C. Larsen and T. V. Voorhis: A relaxation method for nonconvex quadratically constrained quadratic programs. *Journal of Global Optimization*, 6: 215–230, 1995.

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

[4] L. Armijo: Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 16: 1–3, 1966.

[5] C. Audet, P. Hansen, B. Jaumard and G. Savard: A branch and cut algorithm for nonconvex quadratically constrained quadratic programming. *Mathematical Programming*, 87: 131–152, 2000.

[6] L. Bai, J. E. Mitchell and J. S. Pang: Using quadratic convex reformulation to tighten the convex relaxation of a quadratic program with complementarity constraints. *Optimization Letters*, 8: 811–822, 2014.

[7] S. Boyd and L. Vandenberghe: *Convex Optimization*. Cambridge University Press, UK, 2010.

[8] S. Boyd, L. Xiao, A. Mutapcic: Subgradient Methods. lecture notes of EE392o, Stanford University, Autumn Quarter, 2003.

[9] S. Burer, S. Kim and M. Kojima: Faster, but weaker, relaxations for quadratically constrained quadratic programs. *Computational Optimization and Applications*, 59: 27–45, 2014.

[10] Y. Chen and X. Ye: Projection onto s simplex. *ArXiv:1101.6081v2*, 2011.

[11] T. Fujie and M. Kojima: Semidefinite programming relaxation for nonconvex quadratic programs. *Journal of Global Optimization*, 10: 367–380, 1997.

[12] M. X. Goemans: Semidefinite programming in combinatorial optimization. *Mathematical Programming*, 79: 143–161, 1997.

[13] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the Association for Computing Machinery*, 42: 1115–1145, 1995.

[14] Y. Hu, C. Yu and C. Li: Stochastic Subgradient method for quasiconvex optimization problems. *Journal of Nonlinear and Convex Analysis*, 17: 711-724, 2016.

[15] Y. Hu, X. Yang and C. Sim: Inexact subgradient methods for quasiconvex optimization problems. *European Journal of Operational Research*, 240: 315–327, 2015.

[16] R. Jiang and D. Li: Convex relaxations with second order cone constraints for nonconvex quadratically constrained quadratic programming. *ArXiv:1608.02096v1*, 2016.

[17] S. Kim and M. Kojima: Second order cone programming relaxation of nonconvex quadratic optimization problems. *Optimization Methods and Software*, 15: 201–224, 2001.

[18] C. Lu, S. Fang, Q. Jin, Z. Wang and W. Xing: KKT solution and conic relaxation for solving quadratically constrained quadratic programming problems. *SIAM Journal on Optimization*, 21:1475–1490, 2011.

[19] Z. Q. Luo, W. K. Ma, A. M. C. So, Y. Ye and S. Zhang: Semidefinite relaxation of quadratic optimization problems. *IEEE Signal Processing Magazine*, 27: 20–34, 2010.

[20] J. J. Moré and D. C. Sorensen: Computing a trust region step. *SIAM Journal on Scientific and Statistical Computing*, 4: 553–572, 1983.

[21] J. Nocedal and S. J. Wright: *Numerical Optimization*. Springer-Verlag, Berlin, 2006.

[22] I. Novak: Dual bounds and optimality cuts for all-quadratic programs with convex constraints. *Journal of Global Optimization*, 18: 337–356, 2000.

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

[24] F. Rendl, G. Rinaldi and A. Wiegele: Solving max-cut to optimality by intersecting semidefinite and polyhedral relaxations. *Mathematical Programming Series A*, 121: 307–335, 2010.

[25] H. D. Sherali and B. M. P. Fraticelli: Enhancing RLT relaxation via a new class of semidefinite cuts. *Journal of Global Optimization*, 22: 233–261, 2002.

[26] J. F. Sturm and S. Zhang: On cones of nonnegative quadratic functions. *Mathematics of Operations Research*, 28: 246–267, 2003.

[27] H. Tuy: On solving nonconvex optimization problems by reducing the duality gap. *Journal of Global Optimization*, 32: 349–365, 2005.

[28] L. Vandenberghe and S. Boyd: Semidefinite programming. *SIAM Review*, 38: 49–95, 1996.

[29] T. V. Voorhis: A global optimization algorithm using Lagrangian underestimates and the interval newton method. *Journal of Global Optimization*, 24: 349–370, 2002.

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

[31] X. J. Zheng, X. L. Sun, D. Li: Nonconvex quadratically constrained quadratic programming: best D.C.decompositions and their SDP representations. *Journal of Global Optimization*, 50: 695–712, 2011.

[32] SeDuMi optimization over symmetric cones. `http://sedumi.ie.lehigh.edu/`.

A Proofs of Theorems

Proof of Theorem 1

Proof. The vector $\tilde{\boldsymbol{g}}_{\boldsymbol{\lambda}}$ of (21) can be found from

$$\begin{aligned}\psi(\boldsymbol{\lambda}) &= \phi(\bar{\mu}_{\boldsymbol{\lambda}}\boldsymbol{\lambda}) \\ &= \bar{\boldsymbol{x}}_{\boldsymbol{\lambda}}^{\top}\left(Q_0 + \bar{\mu}_{\boldsymbol{\lambda}}\sum_{i=1}^{m}\lambda_i Q_i\right)\bar{\boldsymbol{x}}_{\boldsymbol{\lambda}} + 2\left(\boldsymbol{q}_0 + \bar{\mu}_{\boldsymbol{\lambda}}\sum_{i=1}^{m}\lambda_i\boldsymbol{q}_i\right)^{\top}\bar{\boldsymbol{x}}_{\boldsymbol{\lambda}} + \gamma_0 + \bar{\mu}_{\boldsymbol{\lambda}}\sum_{i=1}^{m}\lambda_i\gamma_i.\end{aligned}$$

We prove that the vector $\tilde{\boldsymbol{g}}_{\boldsymbol{\lambda}}$ is in the quasi-subdifferential $\partial\psi$ defined by (22). Note that in [14, 15], the quasi-subdifferential is defined for a quasi-convex function, but ψ is quasi-concave. Therefore (22) is modified from the original definition of $\partial\psi$ for a quasi-convex function. We further consider (22) as

$$\begin{aligned}\partial\psi(\boldsymbol{\lambda}) &= \{\boldsymbol{s} \mid \psi(\boldsymbol{\nu}) \leq \psi(\boldsymbol{\lambda}), \quad \forall\boldsymbol{\nu};\ \boldsymbol{s}^{\top}(\boldsymbol{\nu}-\boldsymbol{\lambda}) < 0\} \\ &= \{\boldsymbol{s} \mid \psi(\boldsymbol{\nu}) \leq \psi(\boldsymbol{\lambda}), \quad \forall\boldsymbol{\nu};\ \boldsymbol{s}^{\top}\boldsymbol{\nu} < \boldsymbol{s}^{\top}\boldsymbol{\lambda}\} \qquad (37)\end{aligned}$$

Now we show that $\tilde{\boldsymbol{g}}_{\boldsymbol{\lambda}}$ is in (37). When $\bar{\mu}_{\boldsymbol{\lambda}} = 0$, $\tilde{\boldsymbol{g}}_{\boldsymbol{\lambda}} = \boldsymbol{0}$ satisfies (22) and $\tilde{\boldsymbol{g}}_{\boldsymbol{\lambda}} \in \partial\psi(\boldsymbol{\lambda})$ holds. When $\bar{\mu}_{\boldsymbol{\lambda}} > 0$, it is sufficient to consider the vector

$$\boldsymbol{g}_{\boldsymbol{\lambda}} := \begin{pmatrix} \bar{\boldsymbol{x}}_{\boldsymbol{\lambda}}^{\top}Q_1\bar{\boldsymbol{x}}_{\boldsymbol{\lambda}} + 2\boldsymbol{q}_1^{\top}\bar{\boldsymbol{x}}_{\boldsymbol{\lambda}} + \gamma_1 \\ \vdots \\ \bar{\boldsymbol{x}}_{\boldsymbol{\lambda}}^{\top}Q_m\bar{\boldsymbol{x}}_{\boldsymbol{\lambda}} + 2\boldsymbol{q}_m^{\top}\bar{\boldsymbol{x}}_{\boldsymbol{\lambda}} + \gamma_m \end{pmatrix}$$

instead of $\tilde{\boldsymbol{g}}_{\boldsymbol{\lambda}}$ because $\partial\psi(\boldsymbol{\lambda})$ forms a cone. Then, since $\bar{\boldsymbol{x}}_{\boldsymbol{\lambda}}$ is feasible for $\boldsymbol{\lambda}$, we have

$$\bar{\boldsymbol{x}}_{\boldsymbol{\lambda}}^{\top}\left(\sum_{i=1}^{m}\lambda_i Q_i\right)\bar{\boldsymbol{x}}_{\boldsymbol{\lambda}} + 2\left(\sum_{i=1}^{m}\lambda_i\boldsymbol{q}_i\right)^{\top}\bar{\boldsymbol{x}}_{\boldsymbol{\lambda}} + \sum_{i=1}^{m}\lambda_i\gamma_i \leq 0. \qquad (38)$$

From the definition of $\boldsymbol{g}_{\boldsymbol{\lambda}}$, we can rewrite (38) as

$$\boldsymbol{g}_{\boldsymbol{\lambda}}{}^{\top}\boldsymbol{\lambda} \leq 0. \tag{39}$$

Then, we have to prove that for an arbitrary vector $\boldsymbol{\nu}$ which satisfies $\boldsymbol{g}_{\boldsymbol{\lambda}}{}^{\top}\boldsymbol{\nu} < \boldsymbol{g}_{\boldsymbol{\lambda}}{}^{\top}\boldsymbol{\lambda}$,

$$\psi(\boldsymbol{\nu}) \leq \psi(\boldsymbol{\lambda})$$

holds. From (39), we get $\boldsymbol{g}_{\boldsymbol{\lambda}}{}^{\top}\boldsymbol{\nu} \leq 0$ and it means that $\bar{\boldsymbol{x}}_{\boldsymbol{\lambda}}$ is feasible for $\boldsymbol{\nu}$. This implies that at an optimal solution $\bar{\boldsymbol{x}}_{\boldsymbol{\nu}}$ for $\boldsymbol{\nu}$, the optimal value is less than or equal to the one at $\bar{\boldsymbol{x}}_{\boldsymbol{\lambda}}$. Therefore, we have $\psi(\boldsymbol{\nu}) \leq \psi(\boldsymbol{\lambda})$.

Next, we prove (ii)$\sim$(iv). First, $\phi(\boldsymbol{\lambda})$ defined by (3) is concave for $\boldsymbol{\lambda}$. It is a general property of the objective function of the Lagrangian dual problem (e.g. [7]). Furthermore, note that $\psi(\boldsymbol{\lambda})$ is the maximum value of $\phi(\mu\boldsymbol{\lambda})$ with respect to $\mu \geq 0$. Therefore, $\psi(\boldsymbol{\lambda}) \geq \phi(\boldsymbol{0})$ holds for all $\boldsymbol{\lambda} \in \Lambda_{\mathrm{s}}$.

We show (ii) first. Let $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2$ ($\boldsymbol{\lambda}_1 \neq \boldsymbol{\lambda}_2$) be arbitrary points in Λ_{s}, and $\bar{\mu}_1$ and $\bar{\mu}_2$ be optimal solutions of (16) with fixed $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$, respectively. Without loss of generality, we can assume that $\psi(\boldsymbol{\lambda}_1) \geq \psi(\boldsymbol{\lambda}_2)$. Now, it is sufficient to prove that for any fixed $\alpha \in [0, 1]$,

$$\psi(\boldsymbol{\lambda}_\alpha) \geq \psi(\boldsymbol{\lambda}_2), \tag{40}$$

where $\boldsymbol{\lambda}_\alpha := \alpha\boldsymbol{\lambda}_1 + (1-\alpha)\boldsymbol{\lambda}_2$. If $\boldsymbol{\lambda}_1 \notin \Lambda_+$ or $\boldsymbol{\lambda}_2 \notin \Lambda_+$ holds, we get $\psi(\boldsymbol{\lambda}_2) = \phi(\boldsymbol{0})$ and (40) holds. Therefore, we only have to consider the case when both $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$ are in Λ_+, implying that $\bar{\mu}_1$ and $\bar{\mu}_2$ are positive by (20). Since $\phi(\boldsymbol{\lambda})$ is concave for $\boldsymbol{\lambda}$, we can see that for any $\beta \in [0, 1]$,

$$\begin{aligned}\phi(\boldsymbol{\xi}(\beta)) &\geq \beta\phi(\bar{\mu}_1\boldsymbol{\lambda}_1) + (1-\beta)\phi(\bar{\mu}_2\boldsymbol{\lambda}_2)\\ &= \beta\psi(\boldsymbol{\lambda}_1) + (1-\beta)\psi(\boldsymbol{\lambda}_2),\end{aligned}$$

where $\boldsymbol{\xi}(\beta) := \beta\bar{\mu}_1\boldsymbol{\lambda}_1 + (1-\beta)\bar{\mu}_2\boldsymbol{\lambda}_2$. Accordingly, we can confirm that there exists

$$\bar{\beta} := \frac{\bar{\mu}_2\alpha}{\bar{\mu}_1(1-\alpha) + \bar{\mu}_2\alpha} \in [0, 1]$$

which satisfies

$$\boldsymbol{\xi}(\bar{\beta}) = \frac{\bar{\mu}_1\bar{\mu}_2}{\bar{\mu}_1(1-\alpha) + \bar{\mu}_2\alpha}\boldsymbol{\lambda}_\alpha$$

For this $\bar{\beta}$, we get

$$\begin{aligned}\psi(\boldsymbol{\lambda}_\alpha) &= \phi(\bar{\mu}_{\boldsymbol{\lambda}_\alpha}\boldsymbol{\lambda}_\alpha)\\ &\geq \phi\left(\frac{\bar{\mu}_1\bar{\mu}_2}{\bar{\mu}_1(1-\alpha) + \bar{\mu}_2\alpha}\boldsymbol{\lambda}_\alpha\right)\\ &= \phi(\boldsymbol{\xi}(\bar{\beta}))\\ &\geq \bar{\beta}\psi(\boldsymbol{\lambda}_1) + (1-\bar{\beta})\psi(\boldsymbol{\lambda}_2)\\ &\geq \psi(\boldsymbol{\lambda}_2),\end{aligned} \tag{41}$$

where $\bar{\mu}_{\lambda_\alpha}$ is an optimal solution for $\boldsymbol{\lambda}_\alpha$. Therefore, (ii) holds.

We can easily prove (iii). In the above proof of (ii), we assume $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2$ is in Λ_+. Then, (41) means that $\psi(\boldsymbol{\lambda}_\alpha) \geq \psi(\boldsymbol{\lambda}_2) > \phi(\mathbf{0})$, and we get $\boldsymbol{\lambda}_\alpha \in \Lambda_+$ for any $\alpha \in [0, 1]$.

Lastly, we prove (iv). Let $\boldsymbol{\lambda}^\dagger$ be an arbitrary stationary point in Λ_+, and let $\bar{\mu}_{\lambda^\dagger}$ be an optimal solution of (16) for $\boldsymbol{\lambda}^\dagger$. Moreover, $\bar{\boldsymbol{x}}_{\lambda^\dagger}$ denotes an optimal solution of $\phi(\bar{\mu}_{\lambda^\dagger}\boldsymbol{\lambda}^\dagger)$. From (21) and (20), we have

$$\bar{\boldsymbol{x}}_{\lambda^\dagger}^\top Q_i \bar{\boldsymbol{x}}_{\lambda^\dagger} + 2\boldsymbol{q}_i^\top \bar{\boldsymbol{x}}_{\lambda^\dagger} + \gamma_i = 0, \qquad i = 1, \cdots, m. \tag{42}$$

On the other hand, it can be confirmed that

$$\begin{pmatrix} \bar{\boldsymbol{x}}_{\lambda^\dagger}^\top Q_1 \bar{\boldsymbol{x}}_{\lambda^\dagger} + 2\boldsymbol{q}_1^\top \bar{\boldsymbol{x}}_{\lambda^\dagger} + \gamma_1 \\ \vdots \\ \bar{\boldsymbol{x}}_{\lambda^\dagger}^\top Q_m \bar{\boldsymbol{x}}_{\lambda^\dagger} + 2\boldsymbol{q}_m^\top \bar{\boldsymbol{x}}_{\lambda^\dagger} + \gamma_m \end{pmatrix}$$

is a subgradient vector for $\phi(\boldsymbol{\xi}^\dagger)$, where $\boldsymbol{\xi}^\dagger = \bar{\mu}_{\lambda^\dagger}\boldsymbol{\lambda}^\dagger$. Hence, (42) implies that $\bar{\mu}_{\lambda^\dagger}\boldsymbol{\lambda}^\dagger$ is a stationary point of ϕ. From the properties of concave functions, all stationary points are global optimal solutions. Therefore, $\bar{\mu}_{\lambda^\dagger}\boldsymbol{\lambda}^\dagger$ is a global optimal solution of ϕ and $\boldsymbol{\lambda}^\dagger$ is also a global optimal solution of ψ. □

Proof of Theorem 2

Proof. Let Δ be the feasible region for $(\boldsymbol{x}, t)$ of (23) and Δ_{rel} be the feasible region of the relaxed problem (26). Let $\text{conv}(\Delta)$ be the convex hull of Δ. We will prove that $\Delta_{\text{rel}} = \text{conv}(\Delta)$. In this proof, we write "$\boldsymbol{a} \xleftarrow{conv} \{\boldsymbol{b}, \boldsymbol{c}\}$" if

$$\exists s \in [0, 1] \text{ s.t. } \boldsymbol{a} = s\boldsymbol{b} + (1 - s)\boldsymbol{c}$$

holds. From the definition, it is obvious that Δ_{rel} is convex and $\Delta \subseteq \Delta_{\text{rel}}$ holds. Meanwhile, the definition of the convex hull is the minimum convex set that includes Δ, which implies $\text{conv}(\Delta) \subseteq \Delta_{\text{rel}}$ is obvious. The convex hull consists of all the points obtained as convex combinations of any points in Δ. Therefore, if the proposition,

$$\forall \boldsymbol{z} \in \Delta_{\text{rel}}, \ \exists \boldsymbol{z}_1, \boldsymbol{z}_2 \in \Delta; \ \boldsymbol{z} \xleftarrow{conv} \{\boldsymbol{z}_1, \boldsymbol{z}_2\} \tag{43}$$

holds, it leads to $\Delta_{\text{rel}} \subseteq \text{conv}(\Delta)$ and we get $\Delta_{\text{rel}} = \text{conv}(\Delta)$. To show (43), let us choose an arbitrary point $(\boldsymbol{x}^*, t) \in \Delta_{\text{rel}}$ and let t^* be the lower bound of t for $\boldsymbol{x}^*$ in Δ_{rel}. Since $(\boldsymbol{x}^*, t) \in \Delta_{\text{rel}}$ holds for all $t \geq t^*$, if

$$\forall (\boldsymbol{x}^*, t^*) \in \Delta_{\text{rel}}, \ \exists (\boldsymbol{x}_1, t_1), (\boldsymbol{x}_2, t_2) \in \Delta; \quad (\boldsymbol{x}^*, t^*) \xleftarrow{conv} \{(\boldsymbol{x}_1, t_1), (\boldsymbol{x}_2, t_2)\} \tag{44}$$

holds, then for any $\delta_t \geq 0$, $(\boldsymbol{x}^*, t^*+\delta_t) \in \Delta_{\rm rel} \xleftarrow{conv} \{(\boldsymbol{x}_1, t_1+\delta_t), (\boldsymbol{x}_2, t_2+\delta_t)\}$. These points are in Δ, and therefore, it is sufficient to focus on the case of $t = t^*$.

To prove (44), we claim that if a point $(\boldsymbol{x}, t) \in \Delta_{\rm rel}$ satisfies both inequalities of (26) with equality, then the point is also in Δ. Since $\underline{\sigma} < \bar{\sigma}$, we can see that $\boldsymbol{x}^\top Q_\lambda \boldsymbol{x} + 2\boldsymbol{q}_\lambda^\top \boldsymbol{x} + \gamma_\lambda = 0$ by setting the inequalities of (26) to equality and taking their difference. Then, we can easily get $\boldsymbol{x}^\top Q_0 \boldsymbol{x} + 2\boldsymbol{q}_0^\top \boldsymbol{x} + \gamma_0 = t$. Therefore, $(\boldsymbol{x}, t)$ is feasible for (23) and in Δ. In what follows, we focus on when only one of the two inequalities is active.

Then, we have to prove (44) for when $Q_\lambda \succeq O$ and $Q_\lambda \not\succeq O$. However, due to space limitations, we will only show the harder case, i.e., when $Q_\lambda \not\succeq O$, implying $0 < \bar{\sigma} < \infty$. The proof of the other case is almost the same. In the following explanation, we want to find two points in Δ (i.e., points which satisfy both inequalities of (26) with equality). Figure 3 illustrates Δ and $\Delta_{\rm rel}$. In the figure, we want to find P and Q.

The optimal solution $(\boldsymbol{x}^*, t^*)$ of the relaxation problem (26) satisfies at least one of the two inequalities with equality. Here, we claim that the matrix $Q_0 + \sigma Q_\lambda$ ($\sigma \in \{\bar{\sigma}, \underline{\sigma}\}$) in the active inequality has at least one zero eigenvalue and the kernel is not empty if $(\boldsymbol{x}^*, t^*) \notin \Delta$ (the claim is proved at the end of this proof). We denote the matrix in the inactive inequality as $Q_0 + \sigma' Q_\lambda$ ($\sigma' \in \{\bar{\sigma}, \underline{\sigma}\}$). By using σ and σ', $(\boldsymbol{x}^*, t^*)$ satisfies (26) as follows:

$$\begin{cases} {\boldsymbol{x}^*}^\top (Q_0 + \sigma Q_\lambda) \boldsymbol{x}^* + 2(\boldsymbol{q}_0 + \sigma \boldsymbol{q}_\lambda)^\top \boldsymbol{x}^* + \gamma_0 + \sigma \gamma_\lambda = t^*, \\ {\boldsymbol{x}^*}^\top (Q_0 + \sigma' Q_\lambda) \boldsymbol{x}^* + 2(\boldsymbol{q}_0 + \sigma' \boldsymbol{q}_\lambda)^\top \boldsymbol{x}^* + \gamma_0 + \sigma' \gamma_\lambda \leq t^*. \end{cases} \tag{45}$$

Since $Q_0 + \sigma Q_\lambda$ ($\succeq O$) has a zero eigenvalue, we can decompose $\boldsymbol{x}^*$ into

$$\begin{aligned} \boldsymbol{x}^* = \boldsymbol{u}^* + \tau^* \boldsymbol{v}^* \quad \text{s.t.} \quad & \boldsymbol{u}^* \in \mathrm{Ker}(Q_0 + \sigma Q_\lambda)^\perp, \\ & \boldsymbol{v}^* \in \mathrm{Ker}(Q_0 + \sigma Q_\lambda),\ \|\boldsymbol{v}^*\|_2 = 1, \\ & \tau^* \in \mathbb{R}. \end{aligned} \tag{46}$$

Substituting these expressions into the constraints of (45), we get

$$\begin{cases} (\boldsymbol{u}^* + \tau^* \boldsymbol{v}^*)^\top (Q_0 + \sigma Q_\lambda)(\boldsymbol{u}^* + \tau^* \boldsymbol{v}^*) + 2(\boldsymbol{q}_0 + \sigma \boldsymbol{q}_\lambda)^\top (\boldsymbol{u}^* + \tau^* \boldsymbol{v}^*) + \gamma_0 + \sigma \gamma_\lambda = t^*, \\ (\boldsymbol{u}^* + \tau^* \boldsymbol{v}^*)^\top (Q_0 + \sigma' Q_\lambda)(\boldsymbol{u}^* + \tau^* \boldsymbol{v}^*) + 2(\boldsymbol{q}_0 + \sigma' \boldsymbol{q}_\lambda)^\top (\boldsymbol{u}^* + \tau^* \boldsymbol{v}^*) + \gamma_0 + \sigma' \gamma_\lambda \leq t^*. \end{cases} \tag{47}$$

By fixing $\boldsymbol{u}^*$ and $\boldsymbol{v}^*$, we can see that (47) is of the form,

$$\begin{cases} A + \alpha \tau^* = t^*, \\ B + \beta \tau^* + \gamma (\tau^*)^2 \leq t^*, \end{cases} \tag{48}$$

where $(A, B, \alpha, \beta, \gamma)$ are appropriate constants. Here, we regard τ^* and t^* in (48) as variables (τ, t) and illustrate (48) in Figure 19. The feasible region

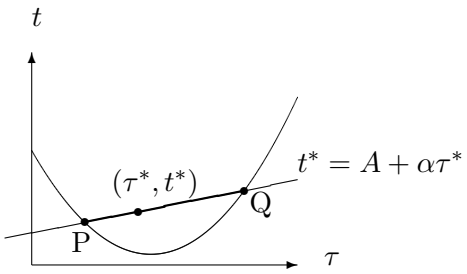

Figure 19: The feasible region of (τ, t)

of (26) for fixed $\boldsymbol{u}^*$ and $\boldsymbol{v}^*$ is shown by the bold line. Note that the line and the parabola have at least one intersection point $(\boldsymbol{x}^*, t^*)$. Here, both points $\mathrm{P}(\tau_1, t_1)$ and $\mathrm{Q}(\tau_2, t_2)$ in Figure 19 satisfy both formulas of (48) with equality, so these points are in Δ. Furthermore, it is obvious from Figure 19 that $(\tau^*, t^*) \overset{conv}{\longleftarrow} \{(\tau_1, t_1), (\tau_2, t_2)\}$. Therefore, (44) holds for any $(\boldsymbol{x}^*, t^*)$.

Now let us check that $\gamma > 0$ and thereby show that the second formula of (48) actually forms a parabola. In (48), $\gamma = {\boldsymbol{v}^*}^\top (Q_0 + \sigma' Q_\lambda)\boldsymbol{v}^* \geq 0$. However, if $\gamma = 0$, then $\boldsymbol{v}^* \in \mathrm{Ker}(Q_0 + \sigma' Q_\lambda)$, so $(Q_0 + \sigma' Q_\lambda)\boldsymbol{v}^* = 0$ holds. Meanwhile, from the definition of $\boldsymbol{v}^*$, we have $(Q_0 + \sigma Q_\lambda)\boldsymbol{v}^* = 0$. Since $\sigma \neq \sigma'$, we get $\boldsymbol{v}^* \in \mathrm{Ker}(Q_0) \cap \mathrm{Ker}(Q_\lambda)$, and this contradicts the Dual Slater condition.

Finally, we prove that $Q_0 + \sigma Q_\lambda$ $(\succeq O)$ has a zero eigenvalue if $(\boldsymbol{x}^*, t^*) \notin \Delta$. From the definition of $\bar{\sigma}$ and $\underline{\sigma}$ (see (24) and (25)), $Q_0 + \bar{\sigma} Q_\lambda$ or $Q_0 + \underline{\sigma} Q_\lambda$ has a zero eigenvalue if $\bar{\sigma}$ or $\underline{\sigma}$ is positive. Moreover, from the Dual Slater condition, $\bar{\sigma} > 0$ holds, so $Q_0 + \bar{\sigma} Q_\lambda$ always has a zero eigenvalue. Therefore, we only have to consider the case when $\underline{\sigma} = 0$, i.e., $Q_0 + \underline{\sigma} Q_\lambda = Q_0$, implying $Q_0 \succeq O$. If Q_0 does not have a zero eigenvalue (i.e. $Q_0 \succ O$), the claim does not hold. However, we can confirm that in this case $(\boldsymbol{x}^*, t^*)$ is already feasible for (23) (i.e. $(\boldsymbol{x}^*, t^*) \in \Delta$) and we do not need to consider this case. We can check its feasibility for (23) by subtracting an equality with $\underline{\sigma}$ $(= 0)$ from an inequality with $\bar{\sigma}$ and dividing the resulting inequality by $\bar{\sigma}$ (> 0). □

Remark 1. This proof suggests that if an optimal solution $\boldsymbol{x}^*$ of (26) is found and $\boldsymbol{x}^*$ is not feasible for (23), we can find an optimal feasible solution of (23). If $(\boldsymbol{x}^*, t^*)$ is an optimal solution of (26), then in Figure 19, the slope α of the line $A + \alpha\tau^* = t^*$ is zero or $(\boldsymbol{x}^*, t^*)$ is equal to the end point P or Q and is already feasible for (23) because t^* is the minimum value such that we can not obtain a t any smaller than t^* in the bold line part in Figure 19. In the former case, we can find an optimal feasible solution of (23) by moving $\boldsymbol{x}^*$ in the direction of $\pm\boldsymbol{v}^*$ defined in (46). We find an optimal feasible solution of (P_k) in this way in SLR.