J Glob Optim (2018) 71:313–339
https://doi.org/10.1007/s10898-018-0617-2

Successive Lagrangian relaxation algorithm for nonconvex quadratic optimization

Shinji Yamada[1,4] · **Akiko Takeda**[2,3]

Received: 31 March 2017 / Accepted: 1 February 2018 / Published online: 16 February 2018

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, a semidefinite 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, the SDP relaxation requires a long computation time, and it has high space complexity for large-scale problems in practice; therefore, the 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.

This research was done at The University of Tokyo before joining the company, not a research conducted by the company.

✉ Akiko Takeda
atakeda@ism.ac.jp

Shinji Yamada
monta0410@hotmail.co.jp

1 Department of Mathematical Informatics, Graduate School of Information Science and Technology, The University of Tokyo, Tokyo, Japan

2 Department of Mathematical Analysis and Statistical Inference, The Institute of Statistical Mathematics, 10-3 Midori-cho, Tachikawa, Tokyo 190-8562, Japan

3 Center for Advanced Intelligence Project (AIP), RIKEN, Tokyo, Japan

4 Present Address: Tokio Marine & Nichido Fire Insurance Co., Ltd., Tokyo, Japan

Keywords Nonconvex quadratically constrained quadratic program · Lagrangian dual optimization · Semidefinite program relaxation · Subgradient method · Quasi-convex function

1 Introduction

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

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

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 $i = 0, \ldots, m$, (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. A standard approach for finding a global solution of (1) 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 et al. [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 [22,28,30]. 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, a semidefinite program (SDP) relaxation is well known and have been extensively studied [11,19,29]. It is known that the an SDP relaxation can be viewed as the Lagrangian dual problem of the original QCQP. The 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]. The SDP relaxation is powerful, and it gives the exact optimal value for 1-QCQP having only one constraint such as 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 et al. [31,32] 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 ordinary SDP relaxation.

However, an 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 et al. [9] proposed a "weaker but faster" method that uses a block matrix decomposition, compared to SDP relaxations. 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 relaxations if valid constraints are not considered. Our method solves the Lagrangian

dual problem of the original QCQP by using a subgradient method, though the Lagrangian 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 levels and iteratively solve the inner problem as a 1-QCQP, which can be solved exactly and efficiently. There are mainly two approaches to solving a 1-QCQP: one is based on eigenvalue computation, the other is on the SDP relaxation. In particular, Moré and Sorensen [20] proposed to iteratively solve a symmetric positive-definite linear system for TRS, while Adachi et al. [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 efficiently 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 relaxations 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 find a global optimum to the original m-QCQP.

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

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

The Lagrangian dual problem of the original problem (1) is described as

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

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

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

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)

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

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

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

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

The SDP relaxation is a popular approach to find a relaxed solution of (1). Sturm and Zhang [27] proved that the SDP relaxation can always obtain the exact optimal value for a 1-QCQP. Goemans and Williamson showed an approximation bound given by the SDP relaxation for max-cut problems [13], and Goemans [12] applied the SDP relaxation to various combinatorial problems. Their numerical experiments show that the SDP relaxation can find a very tight relaxed solution for many kinds of problems. However, the 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.

Remark 1 We propose a subgradient-based approach for (5) instead of applying the interior-point method to the equivalent SDP (6) or (7) in this paper. Efficiently applying the ordinary subgradient method to (5) is not straightforward. Indeed, after obtaining a subgradient vector of $\phi(\boldsymbol{\xi})$, we need to project the vector onto the the set $\Lambda := \{\boldsymbol{\xi} : \boldsymbol{\xi} \geq \mathbf{0}, Q_0 + \sum_{i=1}^m \xi_i Q_i \succeq O\}$. For the projection computation, we need to solve another SDP with similar matrix size to (7). This direct application of a subgradient method to (5) is not a good idea.

We will construct a subset of Λ such that the projection of a vector onto the set is easy and use the subset instead of Λ. The proposed approach may give loser relaxation than the SDP relaxation to QCQPs, but our numerical experiments show that the proposed approach combined with the reformulation linearization technique (RLT) can be better than the SDP relaxation in terms of the tightness of the relaxation and computation time.

2.2 Stronger SDP relaxation using RLT

Anstreicher [3] used the reformulation linearization technique (RLT) to further strengthen the SDP relaxation; new constraints are added to the ordinary SDP relaxation and the range of the new variables X_{ij}, $\forall i, j$ is restricted. 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 $i, j = 1, \ldots, n$. By replacing $x_i x_j$ with X_{ij}, we have

$$X_{ij} - u_i x_j - u_j x_i + u_i u_j \geq 0, \tag{11}$$
$$X_{ij} - u_i x_j - l_j x_i + u_i l_j \leq 0, \tag{12}$$
$$X_{ij} - l_i x_j - l_j x_i + l_i l_j \geq 0. \tag{13}$$

(11)–(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)–(13).

Many studies such as [16,26,31,32] have aimed at strengthening the relaxation by adding valid inequalities other than (11)–(13). These methods give very tight bounds, but they tend to entail large amounts of computation time.

2.3 Weaker SDP relaxation method by block decomposition

Burer et al. [9] aims to solve a relaxed problem faster than SDP relaxations, 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 of a relaxation problem diverges. Then, by following [9], 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 transform a quadratic polynomial as follows,

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

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

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

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 $D_i, \forall i$. Note that in a similar way as (12), we have new constraints $X_{ii} \leq x_i,\ i = 1, \ldots, m$ 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, which will shorten the computation time. We call this method **Block-SDP** and use it in the numerical experiments in Sect. 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 $(i = 0, \ldots, m)$ such that $Q_i \succ O$.
(c) An optimal solution of QCQP (1) can not be obtained by solving the following unconstrained optimization problem

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

Assumption 1 (a) is the primal Slater condition, and (b) is a sufficient condition of the Dual Slater condition of the original QCQP.

3.2 The whole algorithm

We further transform the Lagrangian dual problem (2) into

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

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

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

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

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

There is no duality gap between (16) and its Lagrangian dual (17), since [27] proves that the SDP formulation of (16) has the same optimal value as the 1-QCQP (17). We will show how to exactly and efficiently solve the 1-QCQP (17) in Sect. 4.1. The Lagrangian dual problem (2) can be written as

$$\max_{\boldsymbol{\lambda} \in \Lambda_s} \quad \psi(\boldsymbol{\lambda}). \tag{18}$$

Here, we propose an algorithm which iteratively solves (17) with updated $\boldsymbol{\lambda} \in \Lambda_s$ for finding an optimal solution of the Lagrangian dual problem. Λ_s is a convex set, and $\psi(\boldsymbol{\lambda})$ is a quasi-concave function, as shown in Sect. 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 levels and iteratively solve the inner problem, 1-QCQP, which computes $\psi(\boldsymbol{\lambda})$ with a given $\boldsymbol{\lambda}$. Our method is not stronger than the ordinary SDP relaxation as discussed in Sect. 4. We also explain the details of our method (especially, the relationship between (P_k) and (27) or (28)) later in Sect. 4.

Algorithm 1 Successive Lagrangian Relaxation (SLR)

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

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

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

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

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 Sect. 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 with respect to Lagrange multipliers (e.g. [7]). The function $\phi(\mu\boldsymbol{\lambda})$ with fixed $\boldsymbol{\lambda}$ is hence concave with respect to 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_+,

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

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 $\psi(\boldsymbol{\lambda}) \ge \phi(\boldsymbol{0})$ 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. This 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 the SDP relaxation gives us an optimal solution of (14).

From the definition of Λ_+, we see that an optimal solution of (16) for $\boldsymbol{\lambda} \in \Lambda_+$, which we denote by $\bar{\mu}_{\boldsymbol{\lambda}}$, 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

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

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 of ψ at $\boldsymbol{\lambda}$, which is defined as a vector in the quasi-subdifferential (see e.g., [14,15]):

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

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

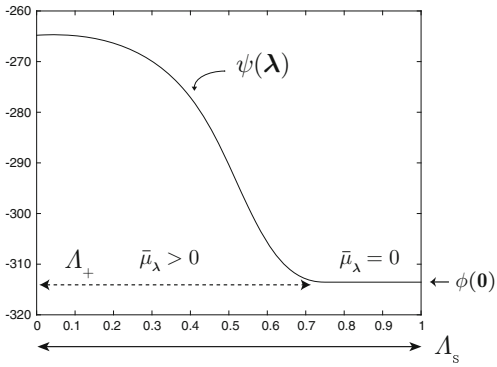

Fig. 1 $\psi(\boldsymbol{\lambda})$ and Λ_+ when $Q_0 \succeq O$

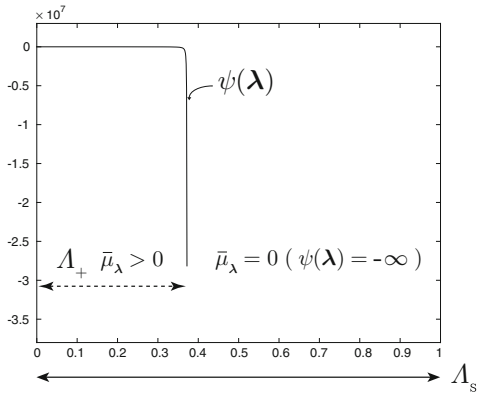

Fig. 2 $\psi(\boldsymbol{\lambda})$ and Λ_+ when $Q_0 \not\succeq O$

(iii) Λ_+ *is a convex set.*
(iv) *If* $\psi(\boldsymbol{\lambda})$ *has stationary points in* Λ_+, *all of them are global optimal solutions of* (18).

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 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 $\boldsymbol{\lambda} \in \Lambda_s$ 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 and also that ψ is not always a concave function even over Λ_+.

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 [14], $\psi(\boldsymbol{\lambda})$ 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 achieves the same optimal value as the SDP relaxation value.

4 Details of SLR

4.1 Subproblem of SLR: 1-QCQP

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:

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

Indeed, ($\mathrm{P_k}$) 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 $Q_0 \succeq O$. We construct the following relaxation problem of (23) using $\bar{\sigma}$ and $\underline{\sigma}$:

$$\begin{array}{ll} \min\limits_{\boldsymbol{x},t} & t \\ \text{s.t.} & \boldsymbol{x}^\top (Q_0 + \underline{\sigma} Q_{\boldsymbol{\lambda}}) \boldsymbol{x} + 2(\boldsymbol{q}_0 + \underline{\sigma} \boldsymbol{q}_{\boldsymbol{\lambda}})^\top \boldsymbol{x} + \gamma_0 + \underline{\sigma} \gamma_{\boldsymbol{\lambda}} \le t, \\ & \boldsymbol{x}^\top (Q_0 + \bar{\sigma} Q_{\boldsymbol{\lambda}}) \boldsymbol{x} + 2(\boldsymbol{q}_0 + \bar{\sigma} \boldsymbol{q}_{\boldsymbol{\lambda}})^\top \boldsymbol{x} + \gamma_0 + \bar{\sigma} \gamma_{\boldsymbol{\lambda}} \le t, \end{array} \tag{26}$$

which will be shown to be equivalent to (23) later in Theorem 2. Note that in the SLR algorithm, we keep either Q_0 or $Q_{\boldsymbol{\lambda}}$ positive semidefinite. When $\underline{\sigma} = 0$ (i.e., $Q_0 \succeq O$), (26)

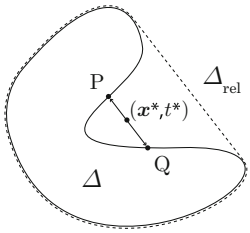

Fig. 3 Image of Δ and Δ_{rel}

is equivalent to the following relaxed problem:

$$\begin{array}{ll} \min\limits_{\boldsymbol{x},t} & t \\ \text{s.t.} & \boldsymbol{x}^\top Q_0 \boldsymbol{x} + 2\boldsymbol{q}_0^\top \boldsymbol{x} + \gamma_0 \le t, \\ & \boldsymbol{x}^\top (\hat{\sigma} Q_0 + Q_{\boldsymbol{\lambda}})\boldsymbol{x} + 2(\hat{\sigma}\boldsymbol{q}_0 + \boldsymbol{q}_{\boldsymbol{\lambda}})^\top \boldsymbol{x} + (\hat{\sigma}\gamma_0 + \gamma_{\boldsymbol{\lambda}}) \le \hat{\sigma} t, \end{array} \tag{27}$$

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

$$\begin{array}{ll} \min\limits_{\boldsymbol{x},t} & t \\ \text{s.t.} & \boldsymbol{x}^\top (Q_0 + \underline{\sigma} Q_{\boldsymbol{\lambda}})\boldsymbol{x} + 2(\boldsymbol{q}_0 + \underline{\sigma}\boldsymbol{q}_{\boldsymbol{\lambda}})^\top \boldsymbol{x} + \gamma_0 + \underline{\sigma}\gamma_{\boldsymbol{\lambda}} \le t, \\ & \boldsymbol{x}^\top Q_{\boldsymbol{\lambda}} \boldsymbol{x} + 2\boldsymbol{q}_{\boldsymbol{\lambda}}^\top \boldsymbol{x} + \gamma_{\boldsymbol{\lambda}} \le 0. \end{array} \tag{28}$$

(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.,* $\Delta_{\text{rel}} = \text{conv}(\Delta)$.

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 Fig. 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 of Δ for an arbitrary point $(\boldsymbol{x}^*, t^*)$ in Δ_{rel}. Using this technique, we can find an optimal solution of 1-QCQP (23). By comparison, SDP relaxations 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**. CQ1 can be constructed from a general 1-QCQP including the Trust Region Subproblem (TRS). The numerical experiments in Sect. 5 show that CQ1 can be solved by a convex quadratic optimization solver faster than by an efficient method for solving a TRS and the interior-point method, and hence, solving CQ1 as subproblems can speed up SLR.

Now let us explain how to calculate $\hat{\sigma}$ or $\underline{\sigma}$ using a positive definite matrix, i.e., for both cases when $Q_0 \succ O$ in (27) and $Q_{\boldsymbol{\lambda}} \succ O$ in (28). We calculate $\hat{\sigma}$ in (27) when $Q_0 \succ O$ as follows. 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, $\hat{\sigma}$ can be calculated as

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

where $\sigma_{\min}(X)$ is the minimum eigenvalue of X, because

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

Similarly, we can calculate $\underline{\sigma}$ in (28) as

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

when $Q_{\boldsymbol{\lambda}} \succ O$.

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

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

Given $\boldsymbol{\lambda}^{(k)}$ and 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 $\boldsymbol{g}^{(k)}$ 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 $\mathbf{proj}_{\Lambda_{\mathrm{s}}}(\boldsymbol{a}) := \arg\min_{\boldsymbol{b} \in \Lambda_{\mathrm{s}}} \|\boldsymbol{a} - \boldsymbol{b}\|_2$ 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 $\boldsymbol{\lambda}^{(k+1)}$.

Now let us explain the update rule of $\boldsymbol{\lambda}$, which is used in **Step 3** of Algorithm 1. Algorithm 2 summarizes the procedure of updating $\boldsymbol{\lambda}$. The whole algorithm, SLR, updates $\boldsymbol{\lambda}^{(k)}$ and $\boldsymbol{x}^{(k)}$ in each iteration $k = 1, 2, \ldots$. 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 $Q_0 \succ O$, 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 $-\infty$ with inappropriately chosen $\boldsymbol{\lambda}^{(k)}$, and we can not find an optimal solution. In such case, we can not calculate $\boldsymbol{g}^{(k)}$ in **Step 1** of Algorithm 2 and the algorithm stops. To

prevent this from happening, we use a subset Λ_+ of Λ_s so that $\psi(\boldsymbol{\lambda}^{(k)})$ does not become $-\infty$ and Λ_+ does not change the optimum value of the Lagrangian dual problem (18). When $Q_0 \not\succ O$, Λ_+ of (20) can be rewritten as

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

under Assumption 1 (b). Λ_+ is the set of $\boldsymbol{\lambda}$ for which $\psi(\boldsymbol{\lambda}) > -\infty$. To exactly solve the problem (18), the projection computation in (29) should be done for the set Λ_+ when $Q_0 \not\succ O$. However, the projection computation can not be easily done because of the complicated condition on μ. Therefore, when $Q_0 \not\succ O$, we approximate Λ_+ as

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

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 we give details on some steps in Algorithm 2.

Step 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 $g_i^{(k)} = \boldsymbol{x}^{(k)\top} Q_i \boldsymbol{x}^{(k)} + 2\boldsymbol{q}_i^\top \boldsymbol{x}^{(k)} + \gamma_i, \forall i$. At the kth iteration of the algorithm, 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 ($\mathrm{P_k}$).

Step 2: We compute $\boldsymbol{\lambda}^{(k+1)}$ with $\boldsymbol{g}^{(k)}$ and h as in (29). We can easily compute the projection onto Λ_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 a 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 $Q_0 \not\succ O$.

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 advantage of SLR is to efficiently find a relaxed solution, so we will choose a simple policy for the step size h.

Step 4: It is needed only when both conditions, $Q_0 \not\succ O$ 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 $C := \{i \mid 1 \le i \le m,\ Q_i \succ O\}$. When $Q_0 \not\succ O$, Assumption 1 (b) assures that C is non-empty.

Remark 2 Remark 1 explains that a direct application of the subgradient method to the SDP (5) is not efficient because the projection computation onto the set $\Lambda := \{\boldsymbol{\xi} : \boldsymbol{\xi} \ge \mathbf{0}, Q_0 + \sum_{i=1}^m \xi_i Q_i \succeq O\}$ is not easy. As shown above, we can project a subgradient vector onto a simple set $\Lambda' := \{\boldsymbol{\xi} : \boldsymbol{\xi} \ge \mathbf{0}\}$ and modify the projected point so as to satisfy $Q_0 + \sum_{i=1}^m \xi_i Q_i \succeq O$. We tried some heuristic modifications in numerical experiments, but the resulting subgradient method with any of modifications does not converge properly.

Now we will discuss why the above subgradient method does not converge unlike our method.

When the concerned problem has $Q_0 \succ O$, the modification is not necessary for our method, while it is necessary for the subgradient method. Even for the case of $Q_0 \not\succ O$, the modification shown in Step 4 of Algorithm 2 is simple and μ in Λ_+ is appropriately determined by solving (P_k), which might give us a good approximation to the projection onto Λ_+. On the other hand, the modification in the subgradient method for satisfying $Q_0 + \sum_{i=1}^m \xi_i Q_i \succeq O$ often drastically changes the projected vector because of Q_0, which may be a reason of unstable performance.

4.3 Setting an initial solution

The number of iterations of SLR depends on how we choose an initial solution. 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.

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

More specifically, 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

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

Schur complement rule 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

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

The right-hand side $\eta_i := \boldsymbol{q}_i^\top Q_i^{-1}\boldsymbol{q}_i - \gamma_i$ can be considered the volume of the ellipsoid and the value $-\eta_i$ can be viewed as the Schur complement of the block Q_i of $\begin{pmatrix} \gamma_i & \boldsymbol{q}_i^\top \\ \boldsymbol{q}_i & Q_i \end{pmatrix}$. From Assumption 1 (a), the ellipsoid has positive volume and we have $\eta_i > 0$. Numerical experiments show that constraints having small positive η_i tend to become active in the SDP relaxation. Therefore, it seems reasonable to give large weights to the constraints whose $-\eta_i$ (< 0) are close to zero; that is, $\frac{1}{|\eta_i|}$ are large. Here, we consider the following 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 can find a better optimal value of the Lagrangian dual problem (2) by adding a redundant convex quadratic constraint constructed similarly to RLT (this is discussed in Sect. 2.2) to (1) when there are box constraints and by applying SLR to the resulting QCQP. Since (9) holds for $1 \le i = j \le n$, we have

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

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

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

We call this method the reformulation quadraticization technique (RQT). Since (34) is a convex quadratic constraint, it may be effective to make the SLR relaxation tighter. The numerical experiments in Sect. 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 compared the performance of SLR, SDP relaxation (6), and Block-SDP shown in Sect. 2.3. We used MATLAB Ver. 8.4.0 (R2014b) for all the numerical experiments. We solved SDP relaxation and Block-SDP by using the interior-point method (SeDuMi 1.3 [25] with the tolerance 10^{-8} (initial setting)). We solved 1-QCQP (27) and (28) in the SLR algorithm by using a convex quadratic optimization solver (CPLEX Ver. 12.5). We used a laptop computer with a 2.4 GHz CPU and 16GB RAM. 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.

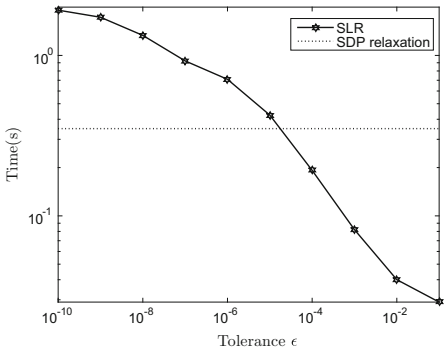

Fig. 4 Average computation time versus tolerance ϵ

5.1 Performance of SLR for random m-QCQP

First, we checked the performance of SLR for random m-QCQP generated in the way of Zheng et al. [32]. In this subsection, we consider problems without box constraints; we compare SLR (or the subroutine of solving CQ1) and SDP relaxation. In Sect. 5.3, 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 computation time of SLR for given tolerance values ϵ. We randomly generated 30 instances of 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 Fig. 4. The smaller ϵ becomes, the longer the computation takes. In this setting, SLR could solve the 10-QCQP faster than SDP relaxation when $\epsilon > 10^{-4}$. Hence, we set $\epsilon = 10^{-4}$ in what follows.

5.1.2 Comparison of two initial-solution strategies

We compared the two strategies for choosing the initial solutions (30) (or (31)) and (32) and checked the results for $Q_0 \succ O$ and $Q_0 \not\succ O$. We only show results for $Q_0 \not\succ O$ because both initial-solution 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 the ordinary SDP relaxation (6). The performance of SLR is said to be good when the ratio is close to one.

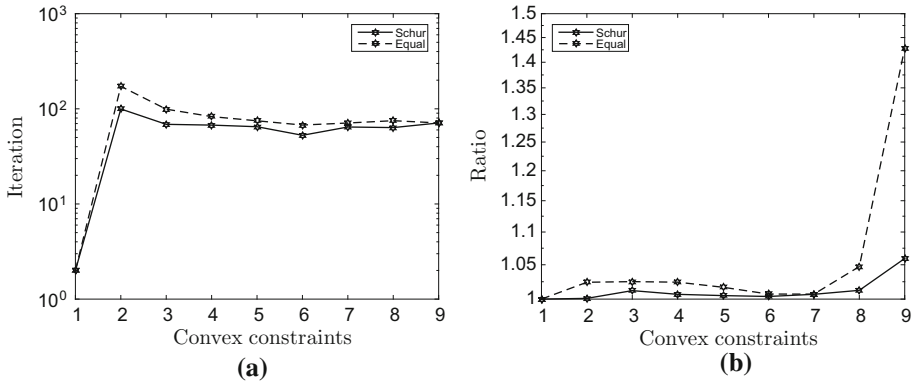

Fig. 5 Comparison of two initial-solution strategies. **a** Average number of iterations ($Q_0 \not\succ O$). **b** Average error ratio ($Q_0 \not\succ O$)

Figures 5a, b 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 a 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. On the basis of the above considerations, we decided to use the Schur complement rule in the remaining experiments.

5.1.3 Computation time and error for the number of variables n and the number of constraints m

We checked the computation time of SLR by changing the number of variables n and the number of constraints m, respectively. In this experiment of Fig. 6a–d, 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.

In Fig. 6e, f, we checked the computation time of SLR by varying the number of constraints m. The number of constraints m was varied from 2 to 50, while the number of variables was fixed as $n = 100$. Half of the constraints (i.e. ceil$(m/2)$) were convex. We generated 30 instances for each setting.

Case 1 $Q_0 \succ O$. SLR performed well when $Q_0 \succ O$ (Fig. 6a, b). The computation time was almost one order of magnitude smaller than that of SDP relaxation, and the error ratio was less than 1.06. The relaxation precision suddenly deteriorated as the number of variables n increased, but there were many instances with smaller n which SLR could obtain the optimal value same as the SDP relaxation value. Furthermore, SLR was able to solve problems with $n = 5000$, while SDP relaxation could not because of out of memory issue. Regarding varied m, the error ratios were less than 1.0015, and the computation time was about one order of magnitude smaller than that of SDP relaxation.

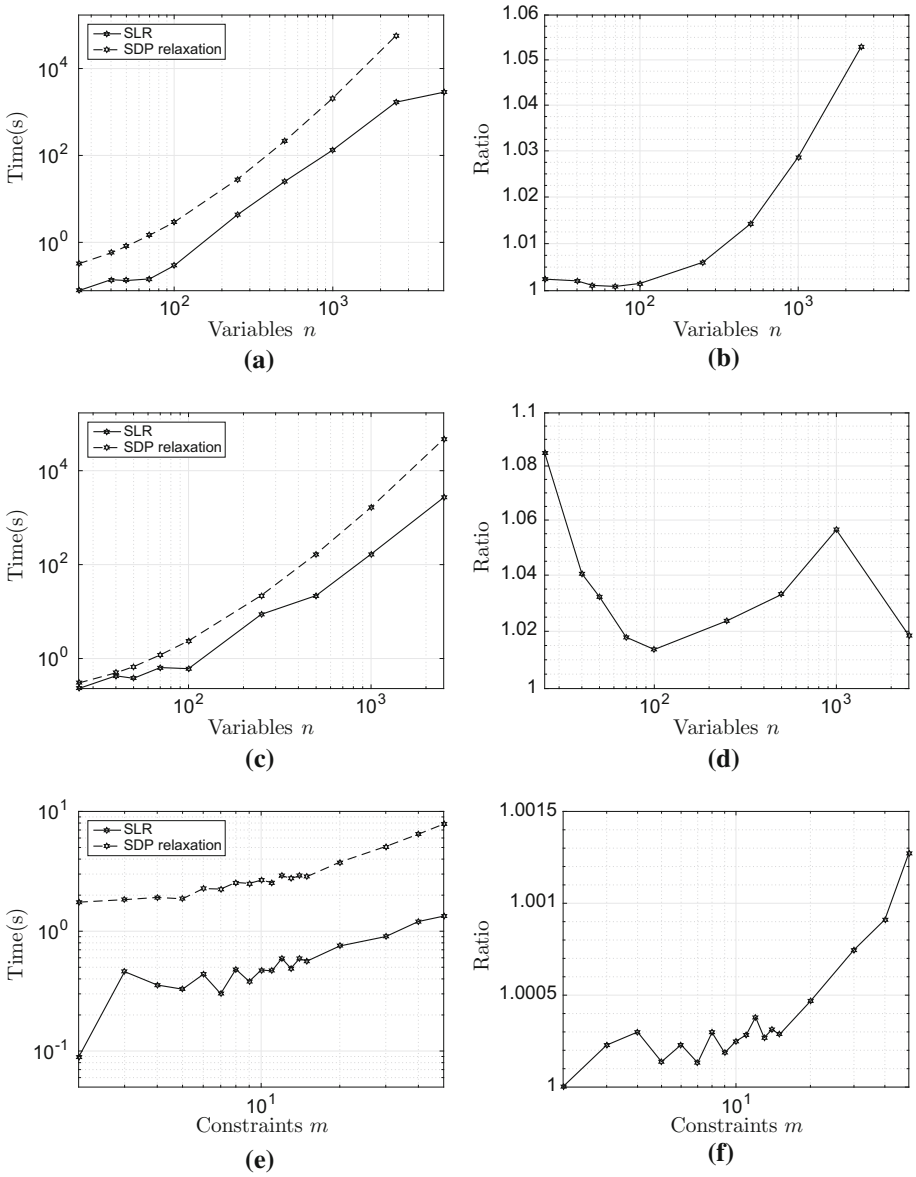

Fig. 6 Computation time and error with varied number of variables n and Varied number of constraints m. **a** Average computation time for n ($Q_0 \succ O$). **b** Average error ratio for n ($Q_0 \succ O$). **c** Average computation time for n ($Q_0 \not\succ O$). **d** Average error ratio for n ($Q_0 \not\succ O$). **e** Average computation time for m ($Q_0 \succ O$). **f** Average error ratio for m ($Q_0 \succ O$)

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. Figure 6c, d show the results for $Q_0 \not\succ O$. The error rate of SLR deteriorated compared to the case of $Q_0 \succ O$, 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.

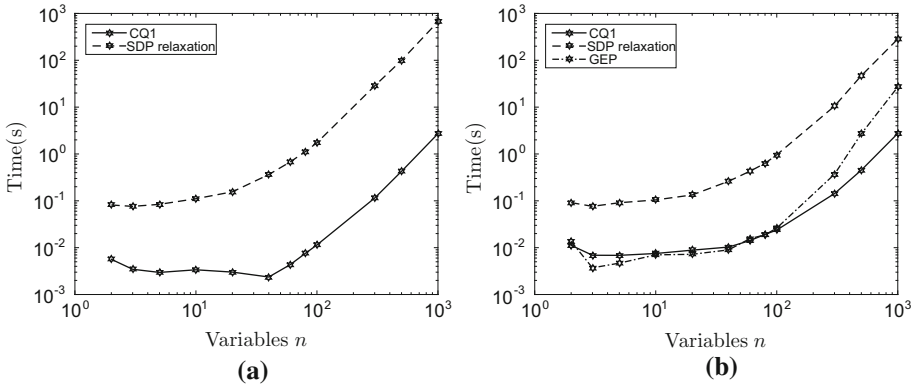

Fig. 7 Comparison of three exact solution methods for 1-QCQP . **a** Average computation time ($Q_0 \succ O$). **b** Average computation time ($Q_1 \succ O$)

Although the numerical results of varied m are omitted to save space in the paper, the trends of time and error with respect to increased m were almost same to Fig. 6e, f except that the error ratio was about 1.06.

SLR tends to give smaller error rates for QCQPs with $Q_0 \succ O$ than for QCQPs with $Q_0 \not\succ O$. This may be because we do no need to approximate the feasible region Λ_+ when $Q_0 \succ O$.

5.2 Performance of CQ1 for solving 1-QCQP

We checked the performance of CQ1; transforming 1-QCQP of ($\mathrm{P_k}$) to the convex quadratic formulation (26) and solving it by the CPLEX solver. 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 et al. [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 $Q_1 \succ O$. It should be noted that all of the methods can obtain the exact optimal value of 1-QCQP. The computation time was plotted versus n. As described in Sect. 5.1, we generated the 1-QCQP in the way of [32]. Figure 7a, b are double logarithmic charts of n and the average computation time of 30 instances. Figure 7a shows that CQ1 is about one order of magnitude faster than SDP relaxation for all n. Figure 7b shows that CQ1 is faster than GEP when n is large. Therefore, we see that solving a quadratic optimization problem with two convex constraints (26) by the CPLEX is a good strategy for solving 1-QCQP of ($\mathrm{P_k}$) in Algorithm 1.

5.3 Strengthened SLR by additional constraints

From the performance of SLR shown in Sect. 5.1, we confirmed that SLR is a "weaker but faster" method than SDP relaxation. The numerical experiment in this subsection implies that SLR including the RQT constraint (34) can be "stronger and faster" than ordinary SDP relaxation: (6) and (7).

We randomly generated problems with box constraints as follows: We varied n from 30 to 2000 and set $m = \mathrm{floor}(0.08n)$ or $m = \mathrm{floor}(0.3n)$ including $\mathrm{ceil}(m/2)$ convex constraints. We generated 30 instances when $n \leq 100$, ten instances when $100 < n \leq 1000$ and

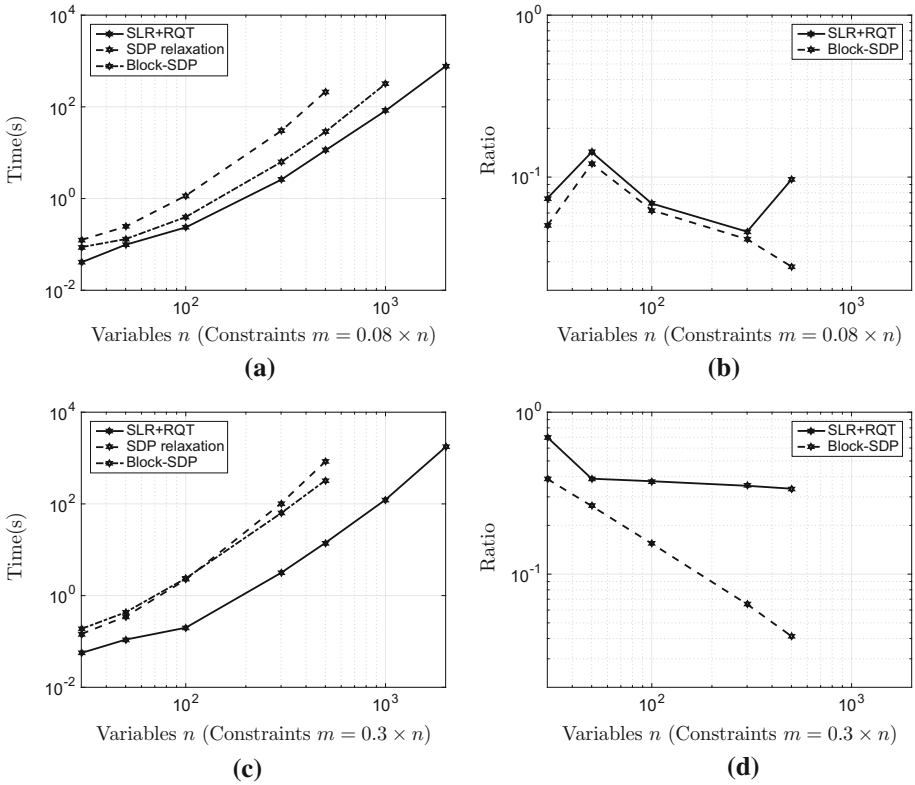

Fig. 8 Effects of additional RQT constraint. **a** Average computation time for n and m. **b** Average error ratio for n and m. **c** Average computation time for n and m. **d** Average error ratio for n and m.

one instance for the case of $n = 2000$. We added box constraints $-1 \leq x_i \leq 1, \forall i$, to all the instances, which make it possible to generate additional constraints such as the RLT constraints (12) and the RQT constraint (34). The following results are only for the case in which the objective function is nonconvex because the RQT constraint affected the performance of our method by much.

The results for instances with smaller number of constraints $m = \text{floor}(0.08n)$ are shown in Fig. 8a, b, while the results for instances with larger number of constraints $m = \text{floor}(0.3n)$ are shown in Fig. 8c, d. "SLR + RQT" indicates that SLR includes the RQT constraint (34). Block-SDP includes some RLT constraints (12) generated from the box constraints (see Sect. 2.3). On the other hand, SDP relaxation includes none of additional constraints.

In Fig. 8b, d, the ratio is less than one. This implies that SLR can get better optimal values than SDP relaxation by adding RQT constraints, i.e, SLR can be "stronger and faster" than SDP relaxation. If all RLT constraints are added to SDPs, the interior-point method for such SDPs will give the tighter relaxation than those three methods. However, the resulting SDPs grow larger than those of ordinary SDP relaxation, which will be a serious problem when solving them with the interior-point method. Indeed, the interior-point method could not solve ordinary SDP instances with $n = 10^3$ because of out of memory issue. Therefore, if we use the interior-point method for SDP relaxation of QCQPs, problem instances of QCQPs might be restricted to less than $n = 10^3$.

The performance of Block-SDP is highly influenced by the number of constraints m. When m is small (i.e., $m = \text{floor}(0.08n)$), Block-SDP performs closely to SLR+RQT in terms of computation time and error ratio. When $m = \text{floor}(0.3n)$, Block-SDP finds very good relaxed solutions with almost the same computation time to the ordinary SDP relaxation. Block-SDP could solve $n = 10^3$ when $m = \text{floor}(0.08n)$[1] but could not solve instances of $n = 2000$ with both settings of m because of out of memory status.

5.4 Iterative application of SLR to 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 $\{-1, 1\}$ integer program:

$$\min_{\boldsymbol{x}} \ \boldsymbol{x}^\top L \boldsymbol{x} \quad \text{s.t.} \quad x_i \in \{-1, 1\}, \quad i = 1, \ldots, n. \tag{35}$$

We relax (35) into

$$\min_{\boldsymbol{x}} \ \boldsymbol{x}^\top L \boldsymbol{x} \quad \text{s.t.} \quad -1 \leq x_i \leq 1, \quad i = 1, \ldots, n,$$

and then apply SDP relaxation and Block-SDP.

In SLR, the box constraints are replaced by the RQT constraint $\boldsymbol{x}^\top \boldsymbol{x} \leq n$ because SLR needs at least one strictly convex constraint. Then the resulting problem is a 1-QCQP;

$$\min_{\boldsymbol{x}} \ \boldsymbol{x}^\top L \boldsymbol{x} \quad \text{s.t.} \ \boldsymbol{x}^\top \boldsymbol{x} \leq n. \tag{36}$$

Therefore, the SLR for (36) is the same approach to CQ1 which solves 1-QCQP ($\mathrm{P_k}$). Here we identify CQ1 and SLR in this problem setting. 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 SLR 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 SLR in terms of either average computation time or average error are listed in bold.

Table 1 shows that SLR is "weaker" but "faster" than SDP relaxation. Block-SDP is weaker and even slower than SDP relaxation in these problem settings. SLR is much faster than SDP relaxation, so we could solve SLR many times with the time to solve an SDP

[1] Block-SDP could not solve one of ten instances with $n = 10^3$ when $m = \text{floor}(0.3n)$ due to the numerical errors that would have been caused by large scale calculations.

Table 1 Time and error for max-cut

Method	SDP relaxation		Block-SDP		SLR		Multiple SLR	
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

by the interior-point method. Accordingly, we tried to strengthen SLR by iterating it with a certain rounding rule as follows. An optimal solution of SLR, $\bar{\boldsymbol{x}}$, 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 SLR find lower bounds. We call this rounding "Multiple SLR" and show the results in the right-most column of Table 1. The results indicate that Multiple SLR 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 various 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 levels 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 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., the Hölder condition in [14]) 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 becomes 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.

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

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

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

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

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 of (17) 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)–(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 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 $\bar{\mu}_{\boldsymbol{\lambda}_\alpha}$ 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 $\psi(\boldsymbol{\lambda}_\alpha) \geq \psi(\boldsymbol{\lambda}_2) > \phi(\boldsymbol{0})$, 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 $\bar{\mu}_{\boldsymbol{\lambda}^\dagger}$ be an optimal solution of (16) for $\boldsymbol{\lambda}^\dagger$. Moreover, $\bar{\boldsymbol{x}}_{\boldsymbol{\lambda}^\dagger}$ denotes an optimal solution of $\phi(\bar{\mu}_{\boldsymbol{\lambda}^\dagger}\boldsymbol{\lambda}^\dagger)$. From (21) and (20), we have

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

On the other hand, it can be confirmed that

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

is a subgradient vector for $\phi(\boldsymbol{\xi}^\dagger)$, where $\boldsymbol{\xi}^\dagger = \bar{\mu}_{\boldsymbol{\lambda}^\dagger}\boldsymbol{\lambda}^\dagger$. Hence, (42) implies that $\bar{\mu}_{\boldsymbol{\lambda}^\dagger}\boldsymbol{\lambda}^\dagger$ is a stationary point of ϕ. From the properties of concave functions, all stationary points are global optimal solutions. Therefore, $\bar{\mu}_{\boldsymbol{\lambda}^\dagger}\boldsymbol{\lambda}^\dagger$ 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 $\mathrm{conv}(\Delta)$ be the convex hull of Δ. We will prove that $\Delta_{\mathrm{rel}} = \mathrm{conv}(\Delta)$. 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 $\Delta \subseteq \Delta_{\mathrm{rel}}$ holds. Meanwhile, the definition of the convex hull is the minimum convex set that includes Δ, which implies $\mathrm{conv}(\Delta) \subseteq \Delta_{\mathrm{rel}}$ is obvious. The convex hull consists of all the points obtained as convex combinations of any points in Δ. Therefore, if the proposition,

$$\forall z \in \Delta_{\mathrm{rel}}, \quad \exists z_1, z_2 \in \Delta; \quad z \xleftarrow{conv} \{z_1, z_2\} \tag{43}$$

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

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

holds, then for any $\delta_t \geq 0$, $(\boldsymbol{x}^*, t^* + \delta_t) \in \Delta_{\mathrm{rel}} \xleftarrow{conv} \{(\boldsymbol{x}_1, t_1 + \delta_t), (\boldsymbol{x}_2, t_2 + \delta_t)\}$. 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 $(\boldsymbol{x}, t) \in \Delta_{\mathrm{rel}}$ satisfies both inequalities of (26) with equality, then the point is also in Δ. Since $\underline{\sigma} < \bar{\sigma}$, we can see that $\boldsymbol{x}^\top Q_{\boldsymbol{\lambda}} \boldsymbol{x} + 2\boldsymbol{q}_{\boldsymbol{\lambda}}^\top \boldsymbol{x} + \gamma_{\boldsymbol{\lambda}} = 0$ 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 $Q_{\boldsymbol{\lambda}} \succeq O$ and $Q_{\boldsymbol{\lambda}} \not\succeq O$. However, due to space limitations, we will only show the harder case, i.e., when $Q_{\boldsymbol{\lambda}} \not\succeq O$, 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 Δ_{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 $Q_0 + \sigma Q_{\boldsymbol{\lambda}}$ ($\sigma \in \{\bar{\sigma}, \underline{\sigma}\}$) in the

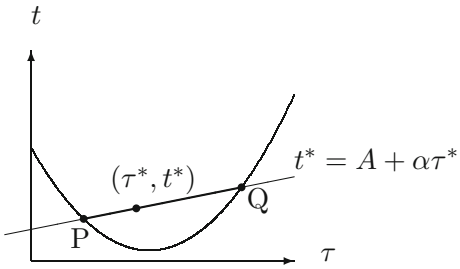

Fig. 9 The feasible region of (τ, t)

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:

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

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

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

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

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

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 Fig. 9. The feasible region 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 Fig. 9 satisfy both formulas of (48) with equality, so these points are in Δ. Furthermore, it is obvious from Fig. 9 that $(\tau^*, t^*) \xleftarrow{conv} \{(\tau_1, t_1), (\tau_2, t_2)\}$. 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), $\gamma = \boldsymbol{v}^{*\top}(Q_0 + \sigma' Q_\lambda)\boldsymbol{v}^* \ge 0$. 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 $\sigma \ne \sigma'$, 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 3 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 Fig. 9, 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 Fig. 9. 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.

References

1. Adachi, S., Iwata, S., Nakatsukasa, Y., Takeda, A.: Solving the trust region subproblem by a generalized eigenvalue problem. SIAM J. Optim. **27**, 269–291 (2017)
2. Al-Khayyal, F.A., Larsen, C., Voorhis, T.V.: A relaxation method for nonconvex quadratically constrained quadratic programs. J. Global Optim. **6**, 215–230 (1995)
3. Anstreicher, K.M.: Semidefinite programming versus the reformulation-linearization technique for nonconvex quadratically constrained quadratic programming. J. Global Optim. **43**, 471–484 (2009)
4. Armijo, L.: Minimization of functions having Lipschitz continuous first partial derivatives. Pac. J. Math. **16**, 1–3 (1966)
5. Audet, C., Hansen, P., Jaumard, B., Savard, G.: A branch and cut algorithm for nonconvex quadratically constrained quadratic programming. Math. Program. **87**, 131–152 (2000)
6. Bai, L., Mitchell, J.E., Pang, J.S.: Using quadratic convex reformulation to tighten the convex relaxation of a quadratic program with complementarity constraints. Optim. Lett. **8**, 811–822 (2014)
7. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2010)
8. Boyd, S., Xiao, L., Mutapcic, A.: Subgradient Methods. Lecture Notes of EE392o, Autumn Quarter. Stanford University, Stanford (2003)
9. Burer, S., Kim, S., Kojima, M.: Faster, but weaker, relaxations for quadratically constrained quadratic programs. Comput. Optim. Appl. **59**, 27–45 (2014)
10. Chen, Y., Ye, X.: Projection onto a simplex (2011), ArXiv:1101.6081v2. Accessed 17 May 2016
11. Fujie, T., Kojima, M.: Semidefinite programming relaxation for nonconvex quadratic programs. J. Global Optim. **10**, 367–380 (1997)
12. Goemans, M.X.: Semidefinite programming in combinatorial optimization. Math. Program. **79**, 143–161 (1997)
13. Goemans, M.X., Williamson, D.P.: Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. J. Assoc. Comput. Mach. **42**, 1115–1145 (1995)
14. Hu, Y., Yang, X., Sim, C.: Inexact subgradient methods for quasi-convex optimization problems. Eur. J. Oper. Res. **240**, 315–327 (2015)
15. Hu, Y., Yu, C., Li, C.: Stochastic subgradient method for quasi-convex optimization problems. J. Nonlinear Convex Anal. **17**, 711–724 (2016)
16. Jiang, R., Li, D.: Convex relaxations with second order cone constraints for nonconvex quadratically constrained quadratic programming (2016). ArXiv:1608.02096v1. Accessed 8 Dec 2016
17. Kim, S., Kojima, M.: Second order cone programming relaxation of nonconvex quadratic optimization problems. Optim. Methods Softw. **15**, 201–224 (2001)
18. Lu, C., Fang, S., Jin, Q., Wang, Z., Xing, W.: KKT solution and conic relaxation for solving quadratically constrained quadratic programming problems. SIAM J. Optim. **21**, 1475–1490 (2011)
19. Luo, Z.Q., Ma, W.K., So, A.M.C., Ye, Y., Zhang, S.: Semidefinite relaxation of quadratic optimization problems. IEEE Signal Process. Mag. **27**, 20–34 (2010)
20. Moré, J.J., Sorensen, D.C.: Computing a trust region step. SIAM J. Sci. Stat. Comput. **4**, 553–572 (1983)
21. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, Berlin (2006)
22. Novak, I.: Dual bounds and optimality cuts for all-quadratic programs with convex constraints. J. Global Optim. **18**, 337–356 (2000)

23. Pardalos, P.M., Vavasis, S.A.: Quadratic programming with one negative eigenvalue is NP-hard. J. Global Optim. **1**, 15–22 (1991)
24. Rendl, F., Rinaldi, G., Wiegele, A.: Solving max-cut to optimality by intersecting semidefinite and polyhedral relaxations. Math. Program. Ser. A **121**, 307–335 (2010)
25. SeDuMi optimization over symmetric cones. http://sedumi.ie.lehigh.edu/
26. Sherali, H.D., Fraticelli, B.M.P.: Enhancing RLT relaxation via a new class of semidefinite cuts. J. Global Optim. **22**, 233–261 (2002)
27. Sturm, J.F., Zhang, S.: On cones of nonnegative quadratic functions. Math. Oper. Res. **28**, 246–267 (2003)
28. Tuy, H.: On solving nonconvex optimization problems by reducing the duality gap. J. Global Optim. **32**, 349–365 (2005)
29. Vandenberghe, L., Boyd, S.: Semidefinite programming. SIAM Rev. **38**, 49–95 (1996)
30. Voorhis, T.V.: A global optimization algorithm using Lagrangian underestimates and the interval newton method. J. Global Optim. **24**, 349–370 (2002)
31. Zheng, X.J., Sun, X.L., Li, D.: Convex relaxations for nonconvex quadratically constrained quadratic programming: matrix cone decomposition and polyhedral approximation. Math. Program. **129**, 301–329 (2011)
32. Zheng, X.J., Sun, X.L., Li, D.: Nonconvex quadratically constrained quadratic programming: best D.C.decompositions and their SDP representations. J. Global Optim. **50**, 695–712 (2011)