SIAM J. OPTIM.
Vol. 29, No. 2, pp. 1603–1633

NOVEL REFORMULATIONS AND EFFICIENT ALGORITHMS FOR THE GENERALIZED TRUST REGION SUBPROBLEM*

RUJUN JIANG† AND DUAN LI‡

Abstract. We present a new solution framework to solve the generalized trust region subproblem (GTRS) of minimizing a quadratic objective over a quadratic constraint. More specifically, we derive a convex quadratic reformulation (CQR) via minimizing a linear objective over two convex quadratic constraints for the GTRS. We show that an optimal solution of the GTRS can be recovered from an optimal solution of the CQR. We further prove that this CQR is equivalent to minimizing the maximum of the two convex quadratic functions derived from the CQR for the case under investigation. Although the latter minimax problem is nonsmooth, it is well structured and convex. We thus develop two steepest descent algorithms corresponding to two different line search rules. We prove global sublinear convergence rates for both algorithms. We also obtain a local linear convergence rate of the first algorithm by estimating the Kurdyka–Łojasiewicz exponent at any optimal solution under mild conditions. We finally demonstrate the efficiency of our algorithms with numerical experiments.

Key words. generalized trust region subproblem, convex reformulation, minimax problems, large-scale problems, Kurdyka–Łojasiewicz inequality

AMS subject classifications. 90C20, 90C25, 90C26, 90C30, 90C47

DOI. 10.1137/18M1174313

1. Introduction.

We consider the following generalized trust region subproblem (GTRS):

$$\text{(P)} \qquad \begin{aligned} &\min f_1(x) := \frac{1}{2}x^\top Q_1 x + b_1^\top x \\ &\text{s.t. } f_2(x) := \frac{1}{2}x^\top Q_2 x + b_2^\top x + c \le 0, \end{aligned}$$

where Q_1 and Q_2 are $n \times n$ symmetric matrices (not necessarily positive semidefinite), $b_1, b_2 \in \mathbb{R}^n$, and $c \in \mathbb{R}$.

Problem (P) is known as the generalized trust region subproblem (GTRS) [44, 40]. When Q_2 is an identity matrix I and $b_2 = 0$, $c = -1/2$, problem (P) reduces to the classical trust region subproblem (TRS). The TRS first arose in the trust region method for nonlinear optimization [12, 50], and has found many applications, including robust optimization [6] and the least squares problems [51]. As a generalization, the GTRS also admits its own applications such as time of arrival problems [22] and subproblems of the consensus alternating direction method of multipliers in signal processing [25]. Over the past two decades, numerous solution methods have been developed for TRS (see [36, 34, 49, 41, 21, 18, 2] and the references therein).

*Received by the editors March 7, 2018; accepted for publication (in revised form) March 19, 2019; published electronically June 13, 2019.

http://www.siam.org/journals/siopt/29-2/M117431.html

Funding: This research was partially supported by the Shanghai Sailing Program, grant 18YF1401700, by the National Natural Science Foundation of China (NSFC), grant 11801087, and by the Hong Kong Research Grants Council under grants 14213716 and 14202017.

†School of Data Science, Fudan University, 220 Handan Road, Yangpu District, Shanghai 200433, People's Republic of China, (rjjiang@fudan.edu.cn).

‡School of Data Science, City University of Hong Kong, Kowloon Tong, Hong Kong (dli226@cityu.edu.hk).

Various methods have been developed for solving the GTRS under various assumptions (see [35, 44, 8, 45, 13, 40, 3] and the references therein). Although it appears to be nonconvex, the GTRS enjoys hidden convexity. The GTRS can be solved via a semidefinite programming (SDP) reformulation, due to the celebrated S-lemma [39], which was first established in [48]. However, as it suffers from relatively large computational complexity, the SDP algorithm is not practical for large-scale applications. To overcome this difficulty, several recent papers [27, 11, 23] have demonstrated that the TRS admits a second-order cone programming (SOCP) reformulation. Ben-Tal and den Hertog [5] further showed an SOCP reformulation for the GTRS under a simultaneous diagonalization (SD) condition of the quadratic forms. Jiang, Li, and Wu [28] derived an SOCP reformulation for the GTRS when the problem has a finite optimal value and further derived a closed-form solution when the SD condition fails. On the other hand, there is a rich literature on iterative algorithms to solve the GTRS directly under mild conditions; see, for example, [35, 44, 40, 42]. Pong and Wolkowicz [40] proposed an efficient algorithm based on minimum generalized eigenvalue of a parameterized matrix pencil for the GTRS, which extended the results in [15] and [41] for the TRS. Salahi and Taati [42] also derived a diagonalization-based algorithm under the SD condition of the quadratic forms. Recently, Adachi and Nakatsukasa [3] also developed a novel eigenvalue-based algorithm to solve the GTRS.

Our main contribution in this paper is a novel convex quadratic reformulation (CQR) for the GTRS that is simpler than [5, 28], and we further propose a minimax problem reformulation and develop an efficient algorithm to solve this reformulation. Numerical results demonstrate that our method outperforms all the existing methods in the literature for sparse problem instances. We acknowledge that our CQR was inspired by the following CQR in Flippo and Janson [14] for the TRS:

$$\min_{x}\left\{\frac{1}{2}x^\top(Q_1-\lambda_{\min}(Q_1)I)x+b_1^\top x+\frac{1}{2}\lambda_{\min}(Q_1)\,:\,x^\top x\le 1\right\}, \tag{1.1}$$

where $\lambda_{\min}(Q_1)$ is the smallest eigenvalue of matrix Q_1. Unfortunately, this CQR was underappreciated at that time. Recently, people have rediscovered this result; Wang and Xia [46] and Ho-Nguyen and Kilinç-Karzan [23] presented a linear time algorithm to solve the TRS by applying Nesterov's accelerated gradient descent algorithm to (1.1). We instead rewrite the epigraph reformulation for (1.1) as follows:

$$\min_{x,t}\left\{t\,:\,\frac{1}{2}x^\top(Q_1-\lambda_{\min}(Q_1)I)x+b_1^\top x+\frac{1}{2}\lambda_{\min}(Q_1)\le t,\ x^\top x\le 1\right\}.$$

Motivated by the above reformulation, we demonstrate that the GTRS is equivalent to exactly one of the following two convex quadratic reformulations under two different conditions,

$$\min_{x,t}\{t\,:\,h_1(x)\le t,\ h_2(x)\le t\}, \tag{P$_1$}$$

$$\min_{x,t}\{t\,:\,h_3(x)\le t,\ f_2(x)\le 0\}, \tag{P$_2$}$$

where $h_1(x)$, $h_2(x)$, and $h_3(x)$ defined in Theorem 2.10 and $f_2(x)$ defined in problem (P) are convex (but possibly not strongly convex) quadratic functions. To the best of our knowledge, it is the first time our proposed CQRs have been derived for the GTRS. The reformulation (P$_2$) only occurs when the quadratic constraint is convex and thus can be solved by a slight modification of [46, 23] in the accelerated gradient projection method by projecting, in each iteration, the current solution to the ellipsoid instead of the unit ball in the TRS case.

In this paper we focus on the problem reformulation (P_1). Although our CQR can be solved as an SOCP problem [7], it is not efficient when the problem size is large. Our main contribution is based on the recognition that problem (P_1) is equivalent to minimizing the maximum of the two convex quadratic functions in (P_1):

$$\text{(M)} \qquad \min\{H(x) := \max\{h_1(x), h_2(x)\}\}.$$

We further derive efficient algorithms to solve the above minimax problem. To the best of our knowledge, the current literature lacks studies on such a problem formulation for a large-scale setting except using a black box subgradient method with an $O(1/\epsilon^2)$ convergence rate [10], which is really slow. Note that section 2.3 in Nesterov's book [38] presents a gradient-based method with linear convergence rate for solving the minimization problem (M) under the condition that both $h_1(x)$ and $h_2(x)$ are strongly convex. However, Nesterov's algorithms cannot be applied to solve our problem since in our problem setting at least one function of $h_1(x)$ and $h_2(x)$ is not strongly convex. By using the special structure of problem (M), we derive a steepest descent method in section 3. More specifically, we choose as the descent direction either the negative gradient when the current point is smooth, or a vector in the subgradient set with the smallest norm (the steepest descent direction) when the current point is nonsmooth, and derive two steepest descent algorithms with two different line search rules accordingly. In the first algorithm we choose a special step size, and in the second algorithm we propose a modified Armijo line search rule. We also prove the global sublinear convergence rate for both algorithms. The first algorithm even admits a global convergence rate of $O(1/\epsilon)$, of the same order as the gradient descent algorithm, which is faster than the subgradient method. In addition, we demonstrate that the first algorithm also admits a local linear convergence rate, by a delicate analysis of the Kurdyka–Łojasiewicz (KL) [4, 9, 32, 16] property for problem (M). In our numerical experiments we illustrate the efficiency of the proposed algorithms compared with state-of-the-art methods for GTRS given in the literature.

The rest of this paper is organized as follows. In section 2, we derive an explicit CQR for problem (P) under different conditions and show how to recover an optimal solution of problem (P) from that of the CQR. In section 3, we reformulate the CQR as a convex nonsmooth unconstrained minimax problem and derive two efficient solution algorithms. We provide convergence analysis for both algorithms. In section 4, we demonstrate the efficiency of our algorithms with numerical experiments. We give conclusions in section 5.

Notation. We use $v(\cdot)$ to denote the optimal value of problem $(\cdot)$. The matrix transpose of matrix A is denoted by $A^\top$, and the inverse of matrix A by A^{-1}, respectively. Let $\mathcal{B}(x,\delta) = \{y : \|y-x\| \le \delta\}$ denote the ball centered at x with radius δ, where $\|\cdot\|$ denotes the Euclidean norm of a vector.

2. Blanket assumptions and convex quadratic reformulation.

In this section, we derive a novel convex quadratic reformulation for problem (P). To avoid some trivial cases, in Assumption 2.1 we assume, without loss of generality (w.l.o.g.), that the Slater condition holds for problem (P), i.e., there exists at least one interior feasible point. When both $f_1(x)$ and $f_2(x)$ are convex, problem (P) is already a convex quadratic problem. Hence, the assumption that not both $f_1(x)$ and $f_2(x)$ are convex is also w.l.o.g. We assume the following *blanket assumptions* hold throughout this paper.

Assumption 2.1. The Slater condition holds for problem (P), and at least one of $f_1(x)$ and $f_2(x)$ is nonconvex.

Assumption 2.2. The set $I_{PSD} := \{\lambda : Q_1 + \lambda Q_2 \succeq 0\} \cap \mathbb{R}_+$ is not empty, where $\mathbb{R}_+$ is the nonnegative orthant.

Assumption 2.3. The common null space of Q_1 and Q_2 is trivial, i.e., $\mathrm{Null}(Q_1) \cap \mathrm{Null}(Q_2) = \{0\}$.

Before introducing our CQR, let us first recall the celebrated S-lemma by defining $\tilde{f}_1(x) = f_1(x) + \gamma$ with an arbitrary constant $\gamma \in \mathbb{R}$.

LEMMA 2.4 (S-lemma [48, 39]). *The following two statements are equivalent:*
1. *the system of $\tilde{f}_1(x) < 0$ and $f_2(x) \le 0$ is not solvable,*
2. *there exists $\mu \ge 0$ such that $\tilde{f}_1(x) + \mu f_2(x) \ge 0$ for all $x \in \mathbb{R}^n$.*

Using the S-lemma, the following lemma shows a necessary and sufficient condition under which problem (P) is bounded from below.

LEMMA 2.5 (see [24, Theorem 4.1]). *When the Slater condition holds, problem* (P) *is bounded from below if and only if the following system has a solution for λ:*

$$Q_1 + \lambda Q_2 \succeq 0,\ \ \lambda \ge 0,\ \ b_1 + \lambda b_2 \in \mathrm{Range}(Q_1 + \lambda Q_2).$$

If Assumption 2.2 fails, there exists no nonnegative λ such that $Q_1 + \lambda Q_2 \succeq 0$ and problem (P) is unbounded from below due to Lemma 2.5. We make Assumption 2.3 w.l.o.g., because otherwise we can reduce the problem to a case that satisfies Assumption 2.3 or show that the problem admits a simple analytical solution or is unbounded from below (see Appendix A for detailed discussions). So both Assumptions 2.2 and 2.3 are made w.l.o.g. In the remainder of this section, we always assume that Assumptions 2.1–2.3 hold.

It was shown in [35] that $\{\lambda : Q_1 + \lambda Q_2 \succeq 0\}$ is an interval and thus that $\{\lambda : Q_1 + \lambda Q_2 \succeq 0\} \cap \mathbb{R}_+$ is also an interval (if not empty). Under Assumptions 2.1–2.3, we have the following three cases for I_{PSD}.

Condition 2.6. The set $I_{PSD} = [\lambda_1, \lambda_2]$ with $\lambda_1 < \lambda_2$.

Condition 2.7. The set $I_{PSD} = [\lambda_3, \infty)$.

Condition 2.8. The set $I_{PSD} = \{\lambda_4\}$ is a singleton.

Note that Condition 2.7 occurs only when Q_2 is positive semidefinite. Under Condition 2.8, Q_1 and Q_2 may not be simultaneously diagonalizable and may have 2×2 block pairs in a canonical form under congruence [28]. In this case, when λ is given, Jiang, Li, and Wu [28] showed how to recover an optimal solution if one is attainable, and how to obtain an ϵ optimal solution if the optimal solution is unattainable. So, in the following, we mainly focus on the cases where either Condition 2.6 or Condition 2.7 is satisfied.

LEMMA 2.9. *Under Condition* 2.6 *or Condition* 2.7*, problem* (P) *is bounded from below.*

Proof. Under Condition 2.6 or Condition 2.7, there exists a λ_0 such that $Q_1 + \lambda_0 Q_2 \succ 0$ and $\lambda_0 \ge 0$, which further implies $b_1 + \lambda_0 b_2 \in \mathrm{Range}(Q_1 + \lambda_0 Q_2)$, as $Q_1 + \lambda_0 Q_2$ is nonsingular. With Lemma 2.5, we complete the proof. □

2.1. Convex quadratic reformulation for GTRS. It is obvious that problem (P) is equivalent to its epigraph reformulation as follows:

$$(\mathrm{P}_0) \qquad \min\{t:\ f_1(x)\le t,\ f_2(x)\le 0\}.$$

Now we are ready to present the main result of this section.

THEOREM 2.10. *Under the blanket assumptions, by defining* $h_i(x) = f_1(x) + \lambda_i f_2(x)$, $i = 1, 2, 3$, *we can reformulate problem* (P) *as a convex quadratic problem under Conditions* 2.6 *and* 2.7, *respectively.*

1. *Under Condition* 2.6, *problem* (P) *is equivalent to the convex quadratic problem*

$$(\mathrm{P}_1) \qquad \min_{x,t}\{t:\ h_1(x)\le t,\ h_2(x)\le t\}.$$

2. *Under Condition* 2.7, *problem* (P) *is equivalent to the convex quadratic problem*

$$(\mathrm{P}_2) \qquad \min_{x,t}\{t:\ h_3(x)\le t,\ f_2(x)\le 0\} = \min_{x}\{h_3(x):\ f_2(x)\le 0\}.$$

Proof. Let us first consider the case where Condition 2.6 holds. Due to Lemma 2.9, (P_1) is bounded from below. Together with the assumed Slater conditions, problem (P_1) admits the same optimal value as its Lagrangian dual [7]. Due to the S-lemma, problem (P) also has the same optimal value as its Lagrangian dual [45],

$$(\mathrm{D}) \qquad \max_{\mu\ge 0}\min_{x} f_1(x) + \mu f_2(x).$$

Under Condition 2.6, i.e., $I_{PSD} = [\lambda_1, \lambda_2]$ with $\lambda_1 < \lambda_2$, it is easy to show that (P_1) is a relaxation of (P_0) since they have the same objective function and the feasible region of (P_1) contains that of (P_0) (note that $f_1 \le t$ and $f_2 \le 0$ imply that $f_1(x) - t + uf_2(x) \le 0$ for all $u \ge 0$). Thus,

$$v(\mathrm{P}_1) \le v(\mathrm{P}_0) = v(\mathrm{P}). \tag{2.1}$$

The Lagrangian dual problem of (P_1) is

$$(\mathrm{D}_1) \qquad \max_{s_1,s_2\ge 0}\min_{x,t} t + (s_1+s_2)(f_1(x)-t) + (\lambda_1 s_1 + \lambda_2 s_2) f_2(x).$$

For any primal and dual optimal solution pair (x^*, μ^*) of (P) and (D), due to $\mu^* \in [\lambda_1, \lambda_2]$ as $Q_1 + \mu^* Q_2 \succeq 0$ from Lemma 2.5, we can always find a convex combination $\lambda_1\bar{s}_1 + \lambda_2\bar{s}_2 = \mu^*$ with $\bar{s}_1 + \bar{s}_2 = 1$. Hence, $(x^*, \bar{s}, t)$, with an arbitrary t, is a feasible solution to (D_1), and the objective value of problem (D_1) at $(x^*, \bar{s}, t)$ is the same with the optimal value of (D). This in turn implies

$$v(\mathrm{D}_1) \ge v(\mathrm{D}). \tag{2.2}$$

Since (P_1) is convex and the Slater condition is satisfied (letting t go to infinity gives a strictly feasible point), $v(\mathrm{P}_1) = v(\mathrm{D}_1)$. Finally, by combining (2.1) and (2.2), we have $v(\mathrm{P}_1) = v(\mathrm{D}_1) \ge v(\mathrm{D}) = v(\mathrm{P}) = v(\mathrm{P}_0) \ge v(\mathrm{P}_1)$. So all the inequalities above become equalities and thus (P_1) is equivalent to (P).

Statement 2 can be proved in a similar way, and its proof is thus omitted. □

Remark 2.11. Reformulation (P$_2$) generalizes the approaches in [14, 46, 23] for the classical TRS with the unit ball constraint to the GTRS with a general convex quadratic constraint.

To the best of our knowledge, there is no method in the literature to compute λ_1 and λ_2 in Condition 2.6 for general Q_1 and Q_2. However, there exist efficient methods in the literature to compute λ_1 and λ_2 when a λ_0 is given such that $Q_1 + \lambda_0 Q_2 \succ 0$ is satisfied. More specifically, the method mentioned in [3, section 2.4.1] gives a way to compute λ_1 and λ_2: first detect a λ_0 such that $Q_0 := Q_1 + \lambda_0 Q_2 \succ 0$, and then compute λ_1 and λ_2 by some generalized eigenvalues that are nearest to 0 for a definite matrix pencil. The reader may refer to [20] for one of the state-of-the-art methods for detecting λ_0. In [35, section 5] we can also find another iterative method to compute $\lambda_0 \in \operatorname{int}(I_{PSD})$, by reducing the length of an interval $[\bar\lambda_1, \bar\lambda_2] \supset I_{PSD}$. We next report our new method to compute λ_1 and λ_2, which is motivated by [40]. Our first step is to find a λ_0 such that $Q_0 := Q_1 + \lambda_0 Q_2 \succ 0$. Then if Q_2 is negative semidefinite, under Assumptions 2.1–2.3, Condition 2.6 holds if and only if Q_1 is positive definite; the interval is then just $[0, u_0]$, where u_0 is the maximum generalized eigenvalue of $Q_1 + uQ_2$. Otherwise, we compute the maximum generalized eigenvalues for $Q_2 + \mu Q_0$ and $-Q_2 + \mu Q_0$, denoted by u_1 and u_2, respectively. Note that both $u_1 > 0$ and $u_2 > 0$ because $Q_0 \succ 0$ and Q_2 has at least one negative eigenvalue and one positive eigenvalue (otherwise λ_2 can be $+\infty$). So we have

$$Q_1 + \left(\frac{1}{u_1} + \lambda_0\right) Q_2 \succeq 0 \text{ and } Q_1 + \left(-\frac{1}{u_2} + \lambda_0\right) Q_2 \succeq 0.$$

Thus, $Q_1 + \eta Q_2 \succeq 0$ for all $\eta \in [\lambda_0 - \frac{1}{u_2}, \lambda_0 + \frac{1}{u_1}]$, which implies $\lambda_1 = \lambda_0 - \frac{1}{u_2}$ and $\lambda_2 = \lambda_0 + \frac{1}{u_1}$. In particular, when one of Q_1 and Q_2 is positive definite, we can skip the step detecting the definiteness, which will save significant time in implementation.

In fact, when λ_0 is given, we only need to compute one extreme eigenvalue—either λ_1 or λ_2—to obtain our convex quadratic reformulation. To see this, we first note that, under the blanket assumptions and Condition 2.6, $Q_1 + \lambda Q_2 \succ 0$ for all $\lambda \in (\lambda_1, \lambda_2)$. Now let us define $x(\lambda) = -(Q_1 + \lambda Q_2)^{-1}(b_1 + \lambda b_2)$ for all $\lambda \in \operatorname{int}(I_{PSD})$, noting that $(Q_1 + \lambda Q_2)^{-1}$ is well defined for all $\lambda \in (\lambda_1, \lambda_2)$, and $\gamma(\lambda) = f_2(x(\lambda))$. In fact, there are Newton-type methods in the literature (e.g., [35]) for solving the GTRS by finding the optimal λ through $\gamma(\lambda) = 0$. However, each step in [35] involves solving a linear system $-(Q_1 + \lambda Q_2)^{-1}(b_1 + b_2)$, which is time consuming for high-dimensional settings. Moreover, Newton's method does not converge in the so-called hard case.[1] On the other hand, for the easy case, an initial λ_0 in I_{PSD} is needed as a safeguard to guarantee the positive definiteness of $Q_1 + \lambda_0 Q_2$ [35]. It is shown in [35] that $\gamma(\lambda)$ is either a strictly decreasing function or a constant in $\operatorname{int}(I_{PSD})$. Following [3], we have the following three cases: if $\gamma(\lambda_0) > 0$, the optimal λ^* is located in $[\lambda_0, \lambda_2]$; if $\gamma(\lambda_0) = 0$, $x(\lambda_0)$ is an optimal solution; and if $\gamma(\lambda_0) < 0$, the optimal λ^* is located in $[\lambda_1, \lambda_0]$. Hence, we have the following corollary, whose proof is similar to that of Theorem 2.10 and thus omitted.

COROLLARY 2.12. *Define* $h_i(x) = f_1(x) + \lambda_i f_2(x)$, $i = 0, 1, 2$. *Under the blanket assumptions and Condition* 2.6, *the following results hold true.*

[1] The definition here follows [35]. In fact, the definitions of the hard case and the easy case of the GTRS are similar to those of the TRS. More specifically, if the null space of the Hessian matrix, $Q_1 + \lambda^* Q_2$, with λ^* the optimal Lagrangian multiplier of problem (P), is orthogonal to $b_1 + \lambda^* b_2$, we are in the hard case; otherwise we are in the easy case.

1. *If* $\gamma(\lambda_0) > 0$, *problem* (P) *is equivalent to the following convex quadratic problem:*

$$(\overline{\mathrm{P}}_1) \qquad \min_{x,t}\{t : h_0(x) \le t,\ h_2(x) \le t\}.$$

2. *If* $\gamma(\lambda_0) = 0$, $x(\lambda_0) = -(Q_1 + \lambda_0 Q_2)^{-1}(b_1 + \lambda_0 b_2)$ *is the optimal solution.*
3. *If* $\gamma(\lambda_0) < 0$, *problem* (P) *is equivalent to the following convex quadratic problem:*

$$(\underline{\mathrm{P}}_1) \qquad \min_{x,t}\{t : h_1(x) \le t,\ h_0(x) \le t\}.$$

Since both $(\overline{\mathrm{P}}_1)$ and $(\underline{\mathrm{P}}_1)$ have a similar form to (P_1) and can be solved in a way that is similar to the solution approach for (P_1), we only discuss how to solve (P_1) in the following.

2.2. Recovery of optimal solutions.

In this subsection, we will discuss the recovery of an optimal solution to problem (P) from an optimal solution to reformulation (P_1). Before that, we first introduce the following theorem. Let us assume from now on that $h_i(x) = \frac{1}{2}x^\top A_i x + a_i^\top x + r_i$, $i = 1, 2$.

From Lemma 2.9, Condition 2.6 implies the boundedness of problem (P) and thus the optimal solution is always attainable [28]. In the following theorem, we show how to recover the optimal solution of problem (P) from an optimal solution of problem (P_1).

THEOREM 2.13. *Assume that Condition* 2.6 *holds and* (x^*, t) *is an optimal solution of problem* (P_1). *Then an optimal solution of problem* (P) *can be obtained in the following ways.*

1. *If* (i) $h_1(x^*) = t$ *and* $h_2(x^*) = t$ *or* (ii) A_1 *is nonsingular,* $h_1(x^*) = t$, *and* $h_2(x^*) < t$, *then* x^* *is an optimal solution to* (P).
2. *Otherwise* (i) $h_1(x^*) < t$ *and* $h_2(x^*) = t$ *or* (ii) A_1 *is singular,* $h_1(x^*) = t$, *and* $h_2(x^*) < t$. *In case* (i), *for any vector* $v_l \in \mathrm{Null}(A_2)$, *letting* $\tilde{\theta}$ *be a solution of the equation*

$$(2.3) \qquad h_1(x^* + \theta v_l) = \frac{1}{2}v_l^\top A_1 v_l \theta^2 + (v_l^\top A_1 x^* + a_1^\top v_l)\theta + h_1(x^*) = t,$$

we have that $\{\tilde{x} : \tilde{x} = x^* + \tilde{\theta} v_l,\ v_l \in \mathrm{Null}(A_2),\ \tilde{\theta}$ *is a solution of* (2.3)$\}$ *forms the set of optimal solutions of* (P). *The recovery for case* (ii) *follows a symmetric rule.*

Proof. Note that at least one of $h_1(x^*) \le t$ and $h_2(x^*) \le t$ is an equality. Then we prove the theorem for the following two cases.

1. If $h_1(x^*) = t$ and $h_2(x^*) = t$, then $f_1(x^*) + \lambda_2 f_2(x^*) = f_1(x^*) + \lambda_1 f_2(x^*)$. Hence, $f_2(x^*) = 0$ due to $\lambda_2 - \lambda_1 > 0$. So we have $f_1(x^*) = t - \lambda_1 f_2(x^*) = t$. If A_1 is nonsingular, then we have $A_1 \succ 0$ from $A_1 \succeq 0$. Then we must have $\lambda_1 = 0$ (the case when the left end point of the interval $\{\lambda : Q_1 + \lambda Q_2 \succeq 0\}$ is negative) and thus $f_1(x^*) = t - \lambda_1 f_2(x^*) = t$. Meanwhile, $h_1(x^*) = t > h_2(x^*)$ implies that $f_1(x^*) + \lambda_1 f_2(x^*) > f_1(x^*) + \lambda_2 f_2(x^*)$, i.e., $f_2(x^*) < 0$.

   As the optimal value of problem (P) must be not smaller than the optimal value of its relaxation (P_1), the fact that $f_1(x^*) = t$ and $f_2(x^*) \le 0$, which means the feasibility of x^*, indicates x^* is an optimal solution for problem (P).

2. We first consider (i) $h_1(x^*) < t$ and $h_2(x^*) = t$. First, note that A_2 has a nontrivial null space, i.e., $\mathrm{Null}(A_2) \neq \{0\}$, which is true because I_{PSD} is an interval. Note also that Condition 2.6 and Assumption 2.3 imply that $Q_1 + \frac{\lambda_1+\lambda_2}{2} Q_2 \succ 0$, i.e., $\frac{A_1+A_2}{2} \succ 0$. Then we have $d^\top A_1 d > 0$ for all nonzero vectors $d \in \mathrm{Null}(A_2)$. We also claim that $a_2^\top d = 0$. Otherwise, setting d such that $a_2^\top d < 0$ (this can be done since we have $a_2^\top(-d) < 0$ if $a_2^\top d > 0$) yields

$$h_2(x^* + d) = h_2(x^*) + \frac{1}{2} d^\top A_2 d + (x^*)^\top A_2 d + a_2^\top d = h_2(x^*) + a_2^\top d < t,$$

where the second equality is due to $d \in \mathrm{Null}(A_2)$ and $h_1(x^* + d) < t$ for any sufficiently small d because $h_1(x^*) < t$. This implies that (x^*, t) is not optimal, which is a contradiction. Equation (2.3) has two solutions due to the positive parameter before the quadratic term, i.e., $v_l^\top A_1 v_l > 0$ and the negative constant, i.e., $h_1(x^*) - t < 0$. With the definition of $\tilde{\theta}$, we know $h_1(\tilde{x}) = t$ and $h_2(\tilde{x}) = t$. This further implies $f_1(\tilde{x}) = t$ and $f_2(\tilde{x}) = 0$, i.e., $\tilde{x}$ is an optimal solution to (P).

   The recovery for case (ii) follows a symmetric rule because case (ii) is symmetrical to case (i). □

Remark 2.14. In item 2 of the above proof, $a_2^\top d = 0$ indicates that problem (P) is in the hard case.

We next illustrate our recovery approach for the following simple example:

$$\min\left\{3x_1^2 - \frac{1}{2}x_2^2 - x_2 : -x_1^2 + \frac{1}{2}x_2^2 + x_2 + 1 \le 0\right\}.$$

Note that, for this example, Condition 2.6 holds, $\lambda_1 = 1$, and $\lambda_2 = 3$. Then we have the following CQR:

$$\min\{t : 2x_1^2 + 1 \le t,\ x_2^2 + 2x_2 + 3 \le t\}.$$

An optimal solution of the CQR is $x = (0, -1)^\top$, $t = 2$. However, this x is not feasible for (P). Using the approach in Theorem 2.13, we obtain an optimal solution, $\tilde{x} = (\frac{\sqrt{2}}{2}, -1)^\top$, to problem (P). In fact, this instance is in the hard case since the optimal Lagrangian multiplier, $\lambda^* = 3$, is at the end of the interval $\{\lambda : Q_1 + \lambda Q_2 \succeq 0,\ \lambda \ge 0\}$ and $a + \lambda^* b \in \mathrm{Range}(Q_1 + \lambda^* Q_2)$.

Finally, we point out that our method can be extended to the following variants of GTRS with equality constraint and interval constraint:

$$\text{(EP)} \qquad \begin{aligned} \min f_1(x) &:= \frac{1}{2}x^\top Q_1 x + b_1^\top x \\ \text{s.t. } f_2(x) &:= \frac{1}{2}x^\top Q_2 x + b_2^\top x + c = 0, \end{aligned}$$

and

$$\text{(IP)} \qquad \begin{aligned} \min f_1(x) &:= \frac{1}{2}x^\top Q_1 x + b_1^\top x \\ \text{s.t. } c_1 \le f_2(x) &:= \frac{1}{2}x^\top Q_2 x + b_2^\top x \le c_2. \end{aligned}$$

It is shown in [40, 28] that (IP) can be reduced to (EP) with minor computation. It is obvious that all our previous results for inequality constrained GTRS hold for (EP) if we remove the nonnegativity requirement for λ in I_{PSD}. We thus omit detailed discussion on (EP) to save space.

3. Efficient algorithms in solving the minimax problem reformulation of the CQR.

In this section, we propose efficient algorithms to solve the GTRS under Condition 2.6. As shown in Theorem 2.10 and Corollary 2.12, the GTRS is equivalent to (P_1), ($\overline{\mathrm{P}}_1$), or ($\underline{\mathrm{P}}_1$). The three problems have similar forms and can be solved by the method proposed in this section. Hence, to save space, we only consider solution algorithms for (P_1) in this section.

The convex quadratic problem (P_1) can be cast as an SOCP problem and solved by many existing solvers, e.g., CVX [19], CPLEX [26], and MOSEK [37]. However, the SOCP reformulation is not very efficient when the dimension is large (e.g., the SOCP solver will take about 1,000 seconds to solve a problem of dimension 10,000).[2] Fortunately, due to its simple structure, (P_1) is equivalent to the following minimax problem of two convex quadratic functions:

$$\text{(M)} \qquad \min\{H(x) := \max\{h_1(x), h_2(x)\}\}.$$

Hence, we aim to derive an efficient method to solve the above minimax problem, thus solving the original GTRS. Our method is a steepest descent method to find a critical point with $0 \in \partial H(x)$. It is well known that such a critical point is an optimal solution of problem (M).

The following theorem tells us how to find the steepest descent direction.

THEOREM 3.1. *Let $g_1 = \nabla h_1(x)$ and $g_2 = \nabla h_2(x)$. If g_1 and g_2 have opposite directions, i.e., $g_1 = -tg_2$ for some constant $t > 0$ and $h_1(x) = h_2(x)$, or if $g_i = 0$ and $h_i(x) \ge h_j(x)$ for $i \ne j$, $i, j \in \{1, 2\}$, then x is a global optimal solution. Otherwise, we can always find the steepest descent direction d in the following way.*

1. *When $h_1(x) \ne h_2(x)$, $d = -g_1$ if $h_1(x) > h_2(x)$ and $d = -g_2$ otherwise.*
2. *When $h_1(x) = h_2(x)$, $d = -(\alpha g_1 + (1-\alpha)g_2)$, where α is defined in the following three cases:*
   (a) *$\alpha = 0$ if $g_1^\top g_1 \ge g_1^\top g_2 \ge g_2^\top g_2$,*
   (b) *$\alpha = 1$ if $g_1^\top g_1 \le g_1^\top g_2 \le g_2^\top g_2$,*
   (c) *$\alpha = \frac{g_2^\top g_2 - g_1^\top g_2}{g_1^\top g_1 + g_2^\top g_2 - 2g_1^\top g_2}$ if $g_1^\top g_2 < g_2^\top g_2$ and $g_1^\top g_2 \le g_1^\top g_1$ or $g_1^\top g_2 \le g_2^\top g_2$ and $g_1^\top g_2 < g_1^\top g_1$.*

Proof. If $h_1(x) = h_2(x)$ and $g_1 = -tg_2$, then $0 \in \partial H(x)$. Hence, by the definition of the subgradient, we have

$$H(y) \ge H(x) + 0^\top (y - x) = H(x) \ \forall y,$$

which further implies that x is the optimal solution.

If $g_i = 0$ and $h_i(x) \ge h_j(x)$ for $i \ne j$, $i, j \in \{1, 2\}$, then for all $y \ne x$ we have $H(y) \ge h_i(y) \ge h_i(x) = H(x)$, i.e., x is a global optimal solution.

Otherwise, we have the following cases.

1. When $h_1(x) \ne h_2(x)$ (suppose, w.l.o.g., $h_1(x) > h_2(x)$) for all $y \in \mathcal{B}(x, \delta)$ with $\mathcal{B}(x, \delta) \subset \{u : h_2(u) < h_1(u)\}$, we have $H(y) = h_1(y)$ and thus $H(x)$ is differentiable at x and smooth in the neighborhood of x, where $\mathcal{B}(x, \delta)$ denotes the ball centered at x with radius δ. Hence, $d = -g_1$ if $h_1(x) > h_2(x)$. Symmetrically, the case with $h_2(x) > h_1(x)$ can be proved in the same way.

[2] The code was implemented in MATLAB 2016a, 64 bit and run on a Linux machine with 48 GB RAM, 2,600 MHz CPU and 64-bit CentOS release 7.1.1503, which is the same as in section 4.

2. When $h_1(x) = h_2(x)$, the steepest descent direction can be found by solving the following problem:

$$\min_{\|y\|\le 1} \max_{g\in\partial H(x)} g^\top y. \tag{3.1}$$

Because $g^\top y$ is bilinear, the first constraint set $\{y : \|y\| \le 1\}$ is convex and compact, and the second constraint set $\partial H(x)$ is convex, by Sion's minimax theorem [43] (see also Appendix B), we have

$$\min_{\|y\|\le 1} \max_{g\in\partial H(x)} g^\top y = \max_{g\in\partial H(x)} \min_{\|y\|\le 1} g^\top y = \max_{g\in\partial H(x)} -\|g\|,$$

where the last equality is due to $\operatorname{argmin}_{\|y\|\le 1} g^\top y = -g/\|g\|$. The above problem (3.1) is then equivalent to $\min_{g\in\partial H(x)} \|g\|^2$, which is exactly the following problem in minimizing a quadratic function of α:

$$\min_{0\le\alpha\le 1} (\alpha g_1 + (1-\alpha)g_2)^\top(\alpha g_1 + (1-\alpha)g_2). \tag{3.2}$$

The first-order optimality condition of the above objective function gives rise to

$$\alpha = \frac{g_2^\top g_2 - g_1^\top g_2}{g_1^\top g_1 + g_2^\top g_2 - 2g_1^\top g_2}.$$

Then if $\frac{g_2^\top g_2 - g_1^\top g_2}{g_1^\top g_1 + g_2^\top g_2 - 2g_1^\top g_2}$ is in the interval $[0, 1]$, the optimal α is given by

$$\frac{g_2^\top g_2 - g_1^\top g_2}{g_1^\top g_1 + g_2^\top g_2 - 2g_1^\top g_2}.$$

Otherwise, (3.2) takes its optimal solution on its boundary. By noting that $g_1^\top g_1 + g_2^\top g_2 - 2g_1^\top g_2 = \|g_1 - g_2\|^2 \ge 0$, we have, in particular,

- $\alpha = 1$ when $\frac{g_2^\top g_2 - g_1^\top g_2}{g_1^\top g_1 + g_2^\top g_2 - 2g_1^\top g_2} > 1$, i.e., $g_2^\top g_2 > g_1^\top g_2$ and $g_1^\top g_1 < g_1^\top g_2$, and
- $\alpha = 0$ when $\frac{g_2^\top g_2 - g_1^\top g_2}{g_1^\top g_1 + g_2^\top g_2 - 2g_1^\top g_2} < 0$, i.e., $g_1^\top g_2 > g_2^\top g_2$ (thus $g_1^\top g_1 > g_1^\top g_2$).

Note that $g_1^\top g_1 + g_2^\top g_2 - 2g_1^\top g_2 = \|g_1 - g_2\|^2 \ge 0$ also implies that we cannot have $g_1^\top g_2 > g_2^\top g_2$ and $g_1^\top g_2 \ge g_1^\top g_1$ or $g_1^\top g_2 \ge g_2^\top g_2$ and $g_1^\top g_2 > g_1^\top g_1$. And $g_1^\top g_2 \ge g_2^\top g_2$ and $g_1^\top g_2 \ge g_1^\top g_1$ if and only if $g_1 = g_2$. Hence, by including the case that $g_1 = g_2$ in (a) and (b), we complete the proof. ∎

Remark 3.2. The above theorem shows that the steepest descent direction at each point with $h_1(x) = h_2(x)$ is either the one with the smaller norm between $\nabla h_1(x)$ and $\nabla h_2(x)$ or the negative convex combination d of $\nabla h_1(x)$ and $\nabla h_2(x)$ such that $\nabla h_1(x)^\top d = \nabla h_2(x)^\top d$.

Using the descent direction *and a close variant* presented in Theorem 3.1, we propose to solve the minimax problem (M) using Algorithms 3.1 and 3.2, respectively: we first compute a descent direction using the idea in Theorem 3.1, then apply two different line search rules for choosing the step size, and finally terminate the algorithm if some termination criterion is met. The advantage of our algorithms is that each iteration is very cheap, thus yielding (with an acceptable number of iterations) a low cost in CPU time. The most expensive operation in each iteration is computing several matrix vector products, which could become cheaper when the matrices are sparse.

Algorithm 3.1 Line search with a special step size for problem (M).

```
Input: Parameters in the minimax problem (M)
Initialize $x_0$
for $k = 0, 1, \ldots,$ do
    if $h_1(x_k) > h_2(x_k)$ then
        set $d_k = -\nabla h_1(x_k)$
    else if $h_1(x_k) < h_2(x_k)$ then
        set $d_k = -\nabla h_2(x_k)$
    else
        set $d_k$ corresponding to Theorem 3.1(2)
    end if
    if termination criterion is met then return
    end if
    Choose a step size $\beta_k$ according to Theorem 3.5
    Update $x_{k+1} = x_k + \beta_k d_k$
end for
```

3.1. Line search with a special step size. In the following, we first derive a local linear convergence rate for Algorithm 3.1 and then demonstrate a global sublinear convergence rate for Algorithm 3.1. We analyze the local convergence rate by studying the growth in the neighborhood of any optimal solution to $H(x)$ in (M). In fact, $H(x)$ belongs to a more general class of piecewise quadratic functions. Error bounds and the KL property, which are two widely used techniques for convergence analysis, have been studied in the literature for several kinds of piecewise quadratic functions; see [31, 33, 52]. However, these results are based on piecewise quadratic functions separated by polyhedral sets, which is not the case for $H(x)$. Li, Mordukhovich, and Pham [29] demonstrated that the KL property holds for the maximum of finite polynomials, but their KL exponent depends on the problem dimension and is close to one, which leads to a very weak sublinear convergence rate. Gao et al. [16] studied the KL exponent for the TRS with the constraint replaced by an equality constraint $x^\top x = 1$. However, their technique depends on the convexity of the function $x^\top x$ and cannot be applied to analyze our problem. A significant result of this paper is to establish the KL exponent of 1/2 for function $H(x)$ when $\min_x H(x) > \max_i\{\min_x h_1(x), \min_x h_2(x)\}$. With this KL exponent, we are able to illustrate the linear convergence of our first algorithm with the proposed special step size.

For completeness, we give the following definition of the KL property.

Definition 3.3 (see [4, 30, 16]). *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a proper lower semicontinuous function satisfying the condition that the restriction of f to its domain is a continuous function. f is said to have the KL property at $\bar{x} \in \{x : 0 \in \partial f(x)\}$ if there exist $C, \epsilon > 0$ and $\theta \in [0, 1)$ such that*

$$C\,\|y\| \geq |f(x) - f(\bar{x})|^\theta \quad \forall x \in B(\bar{x}, \epsilon),\ \forall y \in \partial f(x),$$

where θ is known as the KL exponent.

Under Condition 2.6, we know that there exists $\lambda_0 \geq 0$ such that $Q_1 + \lambda_0 Q_2 \succ 0$ and thus $b_1 + \lambda_0 b_2 \in \mathrm{Range}(Q_1 + \lambda_0 Q_2)$ due to the nonsingularity of $Q_1 + \lambda_0 Q_2$. Hence, from Lemma 2.5, problem (P) (and thus problem (P$_1$)) is bounded from below. It follows that the two matrices are simultaneously diagonalizable and further that the optimal solution of problem (P) is attainable [28]. This further implies that problem (P_1) is bounded from below, with its optimal solution attainable. Assuming that x^* is an optimal solution, the following theorem shows that the KL inequality holds with an exponent of $1/2$ at x^* under some mild conditions.

THEOREM 3.4 (KL property for $H(x)$). *Assume that* $\min h_1(x) < \min H(x)$ *and* $\min h_2(x) < \min H(x)$. *Then, for any* $x^* \in \{x : \partial H(x) = 0\}$, *there exist* $C, \epsilon > 0$ *(depending on* x^**) such that*

$$C\left\|y\right\| \geq |H(x) - H(x^*)|^{1/2} \quad \forall x \in B(x^*, \epsilon),\ \forall y \in \partial H(x). \tag{3.3}$$

Proof. Note that $\min h_1(x) < \min H(x)$ and $\min h_2(x) < \min H(x)$ imply that, for any $x^* \in \{x : \partial H(x) = 0\}$, $\nabla h_1(x^*) \neq 0$ and $\nabla h_2(x^*) \neq 0$, respectively. Assume $L = \max\{\lambda_{\max}(A_1), \lambda_{\max}(A_2)\}$. We carry out our proof by considering the following two cases.

1. For any point with $h_1(x) \neq h_2(x)$, w.l.o.g., assume $h_1(x) > h_2(x)$. This gives rise to $\partial H(x) = \{\nabla h_1(x)\}$. Let $x^* = \operatorname{argmin}_{y\in\{y:\, \partial H(y)=0\}}\|x - y\|$, i.e., x^* is the projection of x onto the optimality solution set. Hence, by noting that h_1 is active at x^*, we have

$$\begin{aligned}|H(x) - H(x^*)| &= \frac{1}{2}(x - x^*)^\top A_1(x - x^*) + (x^*)^\top A_1(x - x^*) + a_1^\top (x - x^*)\\ &\leq \frac{1}{2}L\left\|x - x^*\right\|^2 + \left\|\nabla h_1(x^*)\right\|\left\|x - x^*\right\|.\end{aligned}$$

   On the other hand, $\nabla h_1(x) = A_1 x + a_1$ and

$$\begin{aligned}\left\|\nabla h_1(x)\right\|^2 &= \left\|\nabla h_1(x) - \nabla h_1(x^*) + \nabla h_1(x^*)\right\|^2\\ &= (x - x^*)^\top A_1 A_1(x - x^*) + \left\|\nabla h_1(x^*)\right\|^2\\ &\qquad + 2(\nabla h_1(x^*))^\top A_1(x - x^*)\\ &\geq \left\|\nabla h_1(x^*)\right\|^2 - 2L\left\|\nabla h_1(x^*)\right\|\left\|x - x^*\right\|.\end{aligned}$$

   Define $\epsilon_0 = \min\{1, \frac{\|\nabla h_1(x^*)\|}{4L}\}$. As $\nabla h_1(x^*) \neq 0$ for all $x \in \mathcal{B}(x^*, \epsilon_0)$, we then have

$$|H(x) - H(x^*)| \leq \frac{1}{2}L\epsilon_0^2 + \left\|\nabla h_1(x^*)\right\|\epsilon_0 \leq \frac{9}{32L}\left\|\nabla h_1(x^*)\right\|^2$$

   and

$$\left\|\nabla h_1(x)\right\|^2 \geq \left\|\nabla h_1(x^*)\right\|^2 - 2L\left\|\nabla h_1(x^*)\right\|\epsilon_0 \geq \frac{1}{2}\left\|\nabla h_1(x^*)\right\|^2.$$

   Hence, $|H(x) - H(x^*)|^{\frac{1}{2}} \leq \sqrt{\frac{9}{32L}}\left\|\nabla h_1(x^*)\right\| \leq \frac{3}{4\sqrt{L}}\left\|\nabla h_1(x)\right\|$. So we have the following inequality:

$$|H(x) - H(x^*)|^{1/2} \leq C_0\left\|y\right\|$$

   for all $y \in \partial H(x)$ (here $\{\nabla h_1(x)\} = \partial H(x)$) with $C_0 = \frac{3}{4\sqrt{L}}$.
2. Consider next a point x with $h_1(x) = h_2(x)$. Define

$$h_\alpha(x) = \alpha h_1(x) + (1 - \alpha)h_2(x)$$

for some parameter $\alpha \in [0,1]$. Let $I = \{i \mid (\nabla h_1(x^*))_i \neq 0\}$. The optimality condition $0 \in \partial H(x^*)$ implies that there exists some $\alpha_0 \in [0,1]$ such that $\alpha_0 \nabla h_1(x^*) + (1-\alpha_0)\nabla h_2(x^*) = 0$. Note that $\nabla h_1(x^*) \neq 0$ and $\nabla h_2(x^*) \neq 0$, as assumed, and thus $\alpha_0 \in (0,1)$. Define

$$j = \mathrm{argmax}_i\{|(\nabla h_1(x^*))_i|,\ i \in I\},\ M_1 = |(\nabla h_1(x^*))_j|,\ M_2 = |(\nabla h_2(x^*))_j|.$$

Note that $\alpha_0 \nabla h_1(x^*) + (1-\alpha_0)\nabla h_2(x^*) = 0$ implies that $\alpha_0 M_1 = (1-\alpha_0)M_2$. W.l.o.g, assume $M_1 \geq M_2$ and thus $\alpha_0 \leq \frac{1}{2}$. Since A_1x (respectively, A_2x) is a continuous function of x, there exists an $\epsilon_1 > 0$ (respectively, $\epsilon_2 > 0$) such that, for any $x \in \mathcal{B}(x^*, \epsilon_1)$ (respectively, $x \in \mathcal{B}(x^*, \epsilon_2)$), $\frac{3}{2}M_1 \geq |(\nabla h_1(x))_j| > \frac{1}{2}M_1$ (respectively, $\frac{3}{2}M_2 \geq |(\nabla h_2(x))_j| > \frac{1}{2}M_2$). Let $\epsilon_3 = \min\{\epsilon_1, \epsilon_2\}$. Then we have the following two subcases.

(a) For all $x \in \mathcal{B}(x^*, \epsilon_3)$ and $\alpha \in [0, \frac{1}{4}\alpha_0]$, noting that $h_1(x) = h_2(x)$, we have

$$\begin{aligned}\|\nabla h_\alpha(x)\| &\geq -\alpha|(\nabla h_1(x))_j| + (1-\alpha)|(\nabla h_2(x))_j| \\ &\geq -\frac{3}{2}\alpha M_1 + \frac{1}{2}(1-\alpha)M_2 \\ &\geq -\frac{3}{8}\alpha_0 M_1 + \frac{3}{8}(1-\alpha_0)M_2 + \left(\frac{1}{8} + \frac{1}{8}\alpha_0\right)M_2 \\ &= \left(\frac{1}{8} + \frac{1}{8}\alpha_0\right)M_2.\end{aligned}$$

The third inequality is due to the fact that $-\frac{3}{2}\alpha M_1 + \frac{1}{2}(1-\alpha)M$ is a decreasing function of α. The last equality is due to $\alpha_0 M_1 = (1-\alpha_0)M_2$. Symmetrically, for $\alpha \in [1 - \frac{1-\alpha_0}{4}, 1]$ we have $|(\nabla h_\alpha(x))| \geq (\frac{3}{8} - \frac{1}{4}\alpha_0)M_1$. Combining these two cases with $\alpha_0 \leq \frac{1}{2}$ yields $\|\nabla h_\alpha(x)\| \geq \frac{1}{8}M_2$.

On the other hand, $H(x) - H(x^*) \to 0$ as $x \to x^*$. More specifically, we have

$$\begin{aligned}&|H(x) - H(x^*)| \\ &= \frac{1}{2}(x-x^*)^\top A_1(x-x^*) + (x^*)^\top A_1(x-x^*) + a_1^\top(x-x^*) \\ &\leq \frac{1}{2}L\|x-x^*\|^2 + \|\nabla h_1(x^*)\|\,\|x-x^*\| \\ &\leq \left(\frac{1}{2}L\epsilon_3^2 + \|\nabla h_1(x^*)\|\right)\|x-x^*\|.\end{aligned}$$

Letting $\epsilon_4 = \min\{\epsilon_3, 1\}$ leads to $\frac{M_2^2}{32L\epsilon_3^2 + 64\|\nabla h_1(x^*)\|}|H(x) - H(x^*)| \leq \|\nabla h_\alpha(x)\|^2$. So

$$|H(x) - H(x^*)|^{1/2} \leq C_1\|\nabla h_\alpha(x)\| \qquad \forall \alpha \in \left[0, \frac{1}{4}\alpha_0\right] \cup \left[1 - \frac{1-\alpha_0}{4}, 1\right],\ \forall x \in \mathcal{B}(x^*, \epsilon_4),$$

where $C_1 = \frac{\sqrt{32L\epsilon_3^2 + 64\|\nabla h_1(x^*)\|}}{M_2}$.

(b) Next let us consider the case with $\alpha \in [\frac{\alpha_0}{4}, 1 - \frac{1-\alpha_0}{4}]$. In this case, defining $A_\alpha = \alpha A_1 + (1-\alpha)A_2$ and $a_\alpha = \alpha a_1 + (1-\alpha)a_2$ gives rise to

$$\begin{aligned}\|\nabla h_\alpha(x)\|^2 &= \|\nabla h_\alpha(x) - \nabla h_\alpha(x^*) + \nabla h_\alpha(x^*)\|^2\\ &= (x-x^*)^\top A_\alpha A_\alpha (x-x^*) + \|\nabla h_\alpha(x^*)\|^2\\ &\quad + 2(\nabla h_\alpha(x^*))^\top A_\alpha(x-x^*),\end{aligned}$$

and since $h_1(x) = h_2(x)$ and $h_1(x^*) = h_2(x^*)$,

$$\begin{aligned}|H(x) - H(x^*)| &= \frac{1}{2}(x-x^*)^\top A_\alpha (x-x^*)\\ &\quad + (x^*)^\top A_\alpha(x-x^*) + a_\alpha^\top(x-x^*)\\ &= \frac{1}{2}(x-x^*)^\top A_\alpha(x-x^*) + (\nabla h_\alpha(x^*))^\top(x-x^*).\end{aligned}$$

Define $\mu_0 = \lambda_{\min}(A_\alpha)$. Then

$$\begin{aligned}&\|\nabla h_\alpha(x)\|^2 - 2\mu_0|H(x) - H(x^*)|\\ &\quad = (x-x^*)^\top A_\alpha(A_\alpha - \mu_0 I)(x-x^*) + \|\nabla h_\alpha(x^*)\|^2\\ &\qquad + 2(\nabla h_\alpha(x^*))^\top(A_\alpha - \mu_0 I)(x-x^*)\\ &\quad = \|(A_\alpha - \mu_0 I)(x-x^*) + \nabla h_\alpha(x^*)\|^2\\ &\qquad + \mu_0(x-x^*)^\top(A_\alpha - \mu_0 I)(x-x^*)\\ &\quad \geq 0.\end{aligned}$$

Also note that μ_0 is bounded from below because $\alpha A_1 + (1 - \alpha A_2) \succ 0$ for α in the compact interval $[\frac{\alpha_0}{4}, 1 - \frac{1-\alpha_0}{4}]$. Let $C_2 = \sqrt{1/(2\mu_0)}$. We have

$$C_2\|\nabla h_\alpha(x)\| \geq |H(x) - H(x^*)|^{1/2} \quad \forall \alpha \in \left[\frac{\alpha_0}{4}, 1 - \frac{1-\alpha_0}{4}\right],\ x \in \mathcal{B}(x^*, \epsilon_4).$$

Combining cases (a) and (b) gives rise to

$$|H(x) - H(x^*)|^{1/2} \leq C_3\|\nabla h_\alpha(x)\|$$

with $C_3 = \max\{C_1, C_2\}$ for all $x \in \mathcal{B}(x^*, \epsilon_4)$.

Combining cases 1 and 2 yields that the KL inequality (3.3) holds for all $x \in \mathcal{B}(x^*, \epsilon)$, $\epsilon = \max\{\epsilon_0, \epsilon_4\}$, with $C = \max\{C_0, C_3\}$. □

Note that the assumption of $\min h_1(x) < \min H(x)$ and $\min h_2(x) < \min H(x)$ means that we are in the easy case of GTRS, as in this case λ^* is an interior point of I_{PSD} and $Q_1 + \lambda^* Q_2$ is nonsingular, where λ^* is the optimal Lagrangian multiplier of the GTRS [35]. However, there are two situations for the hard case. Let us consider the KL property for $H(x)$ at the optimal solution x^*. When $h_i(x^*) > h_j(x^*)$ for $i = 1, 2$ and $j = \{1, 2\}/\{i\}$ in the neighborhood x^*, $H(x)$ is just $h_i(x)$ (a convex quadratic function), and the KL exponent is also $1/2$ [4]. In such a case, our algorithm performs asymptotically like the gradient descent method for unconstrained quadratic minimization. However, when $h_i(x^*) = h_j(x^*)$ (note that $\min h_j(x) < H(x^*)$ can still hold in this situation), the KL exponent is not always $1/2$ for $H(x)$. Consider the following counterexample with $h_1(x) = x_1^2$ and $h_2(x) = (x_1+1)^2 + x_2^2 - 1$. The optimal solution is $(0,0)$ and is attained by both h_1 and h_2. Let $x_2 = -\epsilon$, where ϵ is a small positive number. Consider the curve where $h_1(x) = h_2(x)$, which further implies $x_1 = -\epsilon^2/2$. Then we have

$$(1-\beta)\nabla h_1 + \beta\nabla h_2 = 2\begin{pmatrix} -(1-\beta)\frac{\epsilon^2}{2} + \beta(-\frac{\epsilon^2}{2}+1) \\ -\beta\epsilon \end{pmatrix} = 2\begin{pmatrix} -\frac{\epsilon^2}{2} + \beta \\ -\beta\epsilon \end{pmatrix},$$

and thus

$$\begin{aligned} \min_{y\in\partial H(x)} \|y\|^2 &= \min_{\beta} 4\left(\beta^2\epsilon^2 + \beta^2 - \epsilon^2\beta + \frac{\epsilon^4}{4}\right) \\ &= \min_{\beta} 4\left((1+\epsilon^2)\left(\beta - \frac{\epsilon^2}{2(1+\epsilon^2)}\right)^2 - \frac{\epsilon^4}{4(1+\epsilon^2)} + \frac{\epsilon^4}{4}\right) \\ &= \frac{\epsilon^6}{(1+\epsilon^2)} = \epsilon^6 + O(\epsilon^8). \end{aligned}$$

Thus, $\min_{y\in\partial H(x)} \|y\| = O(\epsilon^3)$. On the other hand,

$$H(x) - H(x^*) = x_1^2 = \frac{\epsilon^4}{4}.$$

Hence, the KL inequality cannot hold with $\theta = 1/2$. However, the KL exponent is $\theta = 3/4$ because $\theta = 3/4$ if $h_1(x) = h_2(x)$, and when $h_1(x) \neq h_2(x)$, $\theta = 1/2$ holds for the same reason as in unconstrained convex quadratic minimization [31] if h_1 is active and for the same reason as in case 1 in the proof of Theorem 3.4 if h_2 is active.

One of the anonymous referees pointed out that the results in Theorem 3.4 can also be deduced directly from existing results in [47, Theorem 3.1] and [9, Theorem 5]: let H^* be the optimal value of problem (M). Then consider the system

$$h_1(x) - H^* \leq 0, \quad h_2(x) - H^* \leq 0.$$

Theorem 3.1 of [47] states that a growth condition holds locally (note that the degree of singularity of the system $d = 1$, and locally the term in [47, Theorem 3.1] with exponent $1/2$ is dominant). Combining this with [9, Theorem 5] gives a KL exponent of $1/2$. Moreover, one can further demonstrate that the KL exponent would be $3/4$, as the degree of the singularity of the system will be 2 (e.g., the example in the above paragraph) when our assumption in Theorem 3.4 fails. So the KL exponent in general should be $3/4$. We should point out that our proof of Theorem 3.4 is derived in an independent way and also gives an explicit constant C.

For convergence analysis with error bounds or the KL property, we still need a sufficient descent property to achieve the convergence rate. We next propose an algorithm with such a property. We further show that our algorithm converges locally linearly with the descent direction chosen as in Theorem 3.1 and the step size specified in the following theorem.

THEOREM 3.5. *Assume that the conditions in Theorem* 3.4 *hold and that the initial point $x^0 \in \mathcal{B}(x^*, \epsilon)$. Assume that h_i is the active function when $h_1(x_k) \neq h_2(x_k)$ and h_j, $j = \{1,2\} \setminus \{i\}$, is thus inactive. Let the descent direction be chosen as in Theorem* 3.1 *and the associated step size be chosen as follows.*

1. *When* $h_1(x_k) = h_2(x_k)$,
   - *if there exists* $g_\alpha = \alpha\nabla h_1(x_k) + (1-\alpha)\nabla h_2(x_k)$ *with* $\alpha \in [0,1]$ *such that* $\nabla h_1(x_k)^\top g_\alpha = \nabla h_2(x_k)^\top g_\alpha$, *then set* $d_k = -g_\alpha$ *and* $\beta_k = 1/L$, *where* $L = \max\{\lambda_{\max}(A_1), \lambda_{\max}(A_2)\}$.
   - *otherwise set* $d_k = -\nabla h_i(x_k)$ *for* i *such that*

     $$\nabla h_1(x_k)^\top \nabla h_2(x_k) > \nabla h_i(x_k)^\top \nabla h_i(x_k), \quad i = 1, 2,$$

     and $\beta_k = 1/L$.
2. *When* $h_1(x_k) \neq h_2(x_k)$ *and the quadratic equation for* γ, *i.e.*,

   $$\mathbf{a}\gamma^2 + \mathbf{b}\gamma + \mathbf{c} = 0, \tag{3.4}$$

   where

   $$\begin{aligned}\mathbf{a} &= \frac{1}{2}\nabla h_i(x_k)^\top (A_i - A_j)\nabla h_i(x_k),\\ \mathbf{b} &= (\nabla h_i(x_k)^\top - \nabla h_j(x_k)^\top)\nabla h_i(x_k), \text{ and}\\ \mathbf{c} &= h_i(x_k) - h_j(x_k),\end{aligned}$$

   has no positive solution, or all positive solutions of (3.4) *satisfy* $\gamma \geq 1/L$, *set* $d_k = -\nabla h_i(x_k)$ *and* $\beta_k = 1/L$.
3. *When* $h_1(x_k) \neq h_2(x_k)$ *and the quadratic equation* (3.4) *has a positive solution* $\gamma < 1/L$, *set* $\beta_k = \gamma$ *and* $d_k = -\nabla h_i(x_k)$.

Then the sequence $\{x_k\}$ *generated by Algorithm* 3.1 *satisfies, for any* $k \geq 1$,

$$H(x_k) - H(x^*) \leq \left(\sqrt{\frac{2C^2L - 1}{2C^2L}}\right)^{k-1} (H(x^0) - H(x^*)) \tag{3.5}$$

and

$$\operatorname{dist}(x_k, \overline{X})^2 \leq \frac{2}{L}(H(x_k) - H(x^*)) \leq \frac{2}{L}\left(\sqrt{\frac{2C^2L-1}{2C^2L}}\right)^{k-1} (H(x^0) - H(x^*)),$$

where C *is the same constant as in* (3.3).

Before going to the proof of the above theorem, in the following remark we provide some intuition for how to choose the step size.

Remark 3.6. We here give some intuition for the quadratic equation (3.4). This equation is mainly to detect whether h_i is still active at iteration $k+1$. In fact, the left-hand side (LHS) of the equation is just

$$\frac{1}{2}\nabla h_i(x_k)^\top (A_i - A_j)\nabla h_i(x_k)\gamma^2 + (\nabla h_i(x_k)^\top - \nabla h_j(x_k)^\top)\nabla h_i(x_k)\gamma + h_i(x_k) - h_j(x_k),$$

and is further equivalent to

$$h_i(x_k + \gamma\nabla h_i(x_k)) - h_j(x_k + \gamma\nabla h_i(x_k)).$$

First, note that we assumed h_i is active at iteration k, i.e., $h_i(x_k) - h_j(x_k) > 0$. Note also that the quadratic function in the LHS of (3.4) is still nonnegative, i.e., h_i is still active at iteration $k+1$, for step size less than or equal to $1/L$ if (3.4) has no positive solution or if all positive solutions of (3.4) satisfy $\gamma \geq 1/L$ (step 2). Hence, with the step size $1/L$, h_i is still active at iteration $k+1$ in step 2. And when (3.4) has a positive solution $\gamma < 1/L$, setting $\beta_k = \gamma$ and $d_k = -\nabla h_i(x_k)$ makes $h_1(x_k + \beta_k d_k) = h_2(x_k + \beta_k d_k)$ (step 3). As $h_1(x_{k+1}) = h_2(x_{k+1})$ after every step 3, we must have step 1 at the $(k+1)$st iteration. The main purpose of introducing step 3 is to make sure that (3.7) (in the proof) holds for more than half the steps (this is true for our algorithm, as step 3 is followed by step 1) to guarantee the convergence rate. The reason for this step is that the second equality in (3.7) may fail as the step size may be tiny and instead we have

$$H(x_l) - H(x^*) \leq \frac{2C^2}{L\gamma^2}(H(x_l) - H(x_{l+1})),$$

which is equivalent to

$$H(x_{l+1}) - H(x^*) \leq \frac{2C^2 - L\gamma^2}{2C^2}(H(x_l) - H(x^*)),$$

and thus the function value decrease in step 3 may be very tiny because γ may be very close to 0.

It is worth noting that step 3 in the above theorem is somehow similar to the retraction step in manifold optimization [1]. In manifold optimization, in every iteration each point is retracted to the manifold. In step 3, every point is drawn to the curve $h_1(x) = h_2(x)$.

We next give a proof of Theorem 3.5.

Proof. For simplicity, let us define $g_i = \nabla h_i(x_k)$ for $i = 1, 2$. We claim the following sufficient descent property for steps 1, 2, and 3:

$$H(x_k) - H(x_{k+1}) \geq \frac{L}{2}\left\|x_k - x_{k+1}\right\|^2. \tag{3.6}$$

Hence, if the step size is $1/L$ (i.e., steps 1 and 2), we have

$$H(x_l) - H(x^*) \leq C^2\left\|d_l\right\|^2 = C^2L^2\left\|x_l - x_{l+1}\right\|^2 \leq 2C^2L\left(H(x_l) - H(x_{l+1})\right), \tag{3.7}$$

where the first inequality is due to the KL inequality in Theorem 3.4, the second equality is due to $x_{l+1} = x_l - \frac{1}{L}d_l$, and the last inequality is due to the sufficient descent property. Rearranging the above inequality yields

$$H(x_{l+1}) - H(x^*) \leq \frac{2C^2L - 1}{2C^2L}(H(x_l) - H(x^*)).$$

And due to (3.6), we have $H(x_{l+1}) - H(x^*) \leq H(x_l) - H(x^*)$ for all iterations. Suppose that there are p iterates of step 1, q iterates of step 2, and r iterates of step 3. From the definitions of the steps, every step 3 is followed by a step 1 if we terminate our algorithm at step 1 or 2 and thus $r \leq p+1$. So, for all $k \geq 1$, after $k = p+q+r$ steps we have

$$\begin{aligned} H(x_k) - H(x^*) &\leq \left(\frac{2C^2L - 1}{2C^2L}\right)^{p+q}(H(x^0) - H(x^*)) \\ &\leq \left(\frac{2C^2L - 1}{2C^2L}\right)^{\frac{k-1}{2}}(H(x^0) - H(x^*)). \end{aligned}$$

The sufficient descent property further implies that

$$\frac{L}{2}\sum_{i=k}^{\infty}\|x_i-x_{i+1}\|^2 \le H(x_k)-H(x^*).$$

Hence, with $\sum_{i=k}^{\infty}\|x_i-x_{i+1}\|^2 \ge \operatorname{dist}(x_k,\overline{X})^2$, we have $\frac{L}{2}\operatorname{dist}(x_k,\overline{X})^2 \le H(x_k)-H(x^*)$. Thus,

$$\operatorname{dist}(x_k,\overline{X})^2 \le \frac{2}{L}(H(x_k)-H(x^*)).$$

By noting $g_i = A_i x_k + a_i$, for any $u\in\mathbb{R}^n$ we have

$$h_i(x_k+u)-h_i(x_k) = \frac{1}{2}u^\top A_i u + g_i^\top u,$$

which will be used several times in this proof and other proofs in the remaining part of this section. We next prove our claim of sufficient descent property (3.6) according to the three cases in our updating rule.

1. When $h_1(x_k)=h_2(x_k)$, assuming that h_m, $m\in\{1,2\}$, is active at x_{k+1}, we have

$$H(x_k)-H(x_{k+1}) = h_m(x_k)-h_m(x_{k+1}).$$

- If there exists an α such that $g_\alpha^\top g_1 = g_\alpha^\top g_2$, we have $g_\alpha^\top g_j = g_\alpha^\top g_\alpha$. And by noting that $d_k=-g_\alpha$ and that the step size is $1/L$, we further have

$$\begin{aligned} h_m(x_{k+1})-h_m(x_k) &= \frac{1}{2L^2}d_k^\top A_m d_k + \frac{1}{L}g_m^\top d_k \\ &\le \frac{1}{2L}g_\alpha^\top g_\alpha - \frac{1}{L}g_\alpha^\top g_\alpha \\ &= -\frac{1}{2L}g_\alpha^\top g_\alpha, \end{aligned}$$

where the inequality follows because $\lambda_{\max}(A_m)\le L$ and $g_\alpha^\top g_m = g_\alpha^\top g_\alpha$. Substituting $g_\alpha = L(x_k-x_{k+1})$ into the above expression, we have the following sufficient descent property:

$$H(x_k)-H(x_{k+1}) = h_m(x_k)-h_m(x_{k+1}) \ge \frac{L}{2}\|x_k-x_{k+1}\|^2.$$

- If there does not exist an $\alpha\in[0,1]$ such that $g_\alpha^\top g_1 = g_\alpha^\top g_2$, then we must have $\alpha g_1^\top g_1 + (1-\alpha)g_2^\top g_1 > \alpha g_1^\top g_2 + (1-\alpha)g_2^\top g_2$ for all $\alpha\in[0,1]$ or $\alpha g_1^\top g_1 + (1-\alpha)g_2^\top g_1 < \alpha g_1^\top g_2 + (1-\alpha)g_2^\top g_2$ for all $\alpha\in[0,1]$. As the above relation is linear, the inequality holds if and only if it holds at the end points $\alpha=0$ and $\alpha=1$. And thus we must have $g_1^\top g_1 > g_1^\top g_2 > g_2^\top g_2$ or $g_2^\top g_2 > g_1^\top g_2 > g_1^\top g_1$. If $g_i^\top g_i > g_i^\top g_j > g_j^\top g_j$, we set $d_k = -g_j$. Then

$$\begin{aligned} &H(x_{k+1})-H(x_k) \\ &\quad\le \max\{h_i(x_{k+1})-h_i(x_k),\ h_j(x_{k+1})-h_j(x_k)\} \\ &\quad\le \max\left\{\frac{1}{2L^2}g_j^\top A_i g_j - \frac{1}{L}g_i^\top g_j,\ \frac{1}{2L^2}g_j^\top A_j g_j - \frac{1}{L}g_j^\top g_j\right\} \end{aligned}$$

$$\begin{aligned}
&\le \max\left\{\frac{1}{2L^2}g_j^\top A_i g_j - \frac{1}{L}g_j^\top g_j,\ \frac{1}{2L^2}g_j^\top A_j g_j - \frac{1}{L}g_j^\top g_j\right\}\\
&\le \max\left\{\frac{1}{2L}g_j^\top g_j - \frac{1}{L}g_j^\top g_j,\ \frac{1}{2L}g_j^\top g_j - \frac{1}{L}g_j^\top g_j\right\}\\
&= -\frac{1}{2L}g_j^\top g_j = -\frac{L}{2}\left\|x_k - x_{k+1}\right\|^2.
\end{aligned}$$

Symmetrically, if $g_j^\top g_j > g_i^\top g_j > g_i^\top g_i$, setting $d_k = -g_i$ yields the same sufficient descent property.

2. When $h_1(x_k) \neq h_2(x_k)$ and the quadratic equation (3.4) for γ has no positive solution or all its positive solutions $\gamma \geq 1/L$, we have $h_i(x_{k+1}) > h_j(x_{k+1})$ for $x_{k+1} = x_k + \beta_k d_k$, where $d_k = -\nabla h_i(x_k)$ and $\beta_k = \frac{1}{L}$. Moreover,

$$\begin{aligned}
H(x_{k+1}) - H(x_k) &= h_i(x_{k+1}) - h_i(x_k)\\
&= \frac{1}{2L^2}g_i^\top A_i g_i - \frac{1}{L}g_i^\top g_i\\
&\le -\frac{1}{2L}g_i^\top g_i.
\end{aligned}$$

Hence, $H(x_k) - H(x_{k+1}) \geq \frac{1}{2L}g_i^\top g_i \geq \frac{L}{2}\left\|x_k - x_{k+1}\right\|^2$.

3. When $h_1(x_k) \neq h_2(x_k)$ and the quadratic equation (3.4) has a positive solution, we have $\gamma < 1/L$. With $\beta_k = \gamma$ and $d_k = -\nabla h_i(x_k)$, from the definition of γ, we see that the step size γ makes $h_1(x_{k+1}) = h_2(x_{k+1})$. Then we have

$$\begin{aligned}
H(x_{k+1}) - H(x_k) &= h_i(x_{k+1}) - h_i(x_k)\\
&= \frac{1}{2}\gamma^2 d_k^\top A_i d_k + \gamma g_i^\top d_k\\
&\le \frac{1}{2}L\gamma^2 g_i^\top g_i - \gamma g_i^\top g_i\\
&= \left(\frac{L}{2} - \frac{1}{\gamma}\right)\left\|x_k - x_{k+1}\right\|^2,
\end{aligned}$$

which further implies $H(x_k) - H(x_{k+1}) \geq \frac{L}{2}\left\|x_k - x_{k+1}\right\|^2$ due to $\gamma \leq \frac{1}{L}$. □

We will next show that in general a global sublinear convergence rate, of the same order as the gradient descent algorithm, can also theoretically be guaranteed for Algorithm 3.1.

THEOREM 3.7. *Assume that x^* is an optimal solution and x_0 is an arbitrary initial point. Then we have*

$$H(x_N) - H(x^*) \leq \frac{L}{N}\left\|x_0 - x^*\right\|^2.$$

That is, the required number of iterations for $H(x_N) - H(x^*) \leq \epsilon$ is at most $O(1/\epsilon)$.

Proof. From the proof of Theorem 3.5, for any step size $\gamma \leq 1/L$, we have

$$H(x_{k+1}) - H(x_k) \leq -\gamma g^\top g + \frac{1}{2}L\gamma^2 g^\top g \leq -\frac{\gamma}{2}g^\top g.$$

From the convexity of $H(x)$ and $g \in \partial H(x_k)$, we have

$$\begin{aligned} H(x_{k+1}) &\le H(x_k) - \frac{\gamma}{2} g^\top g \\ &\le H(x^*) + g^\top (x_k - x^*) - \frac{\gamma}{2} g^\top g \\ &= H(x^*) + \frac{1}{2\gamma}\left(\|x_k - x^*\|^2 - \|x_k - x^* - \gamma g\|^2 \right) \\ &= H(x^*) + \frac{1}{2\gamma}\left(\|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2 \right). \end{aligned}$$

Since $H(x_{k+1}) \ge H(x^*)$, we have

$$\|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2 \ge 0. \tag{3.8}$$

Let us use indices i_k, $k = 0, \dots, K$, to denote those in steps 1 and 2. By noting that $\gamma = 1/L$, we have

$$H(x_{i_{k+1}}) \le H(x^*) + \frac{L}{2}\left(\|x_{i_k} - x^*\|^2 - \|x_{i_k+1} - x^*\|^2 \right).$$

Note that every step 3 is followed by step 1. Hence, $N \le 2K + 1$. Adding the above inequalities from i_0 to i_K, we have

$$\begin{aligned} &\sum_{k=0}^{K} H(x_{i_k}) - H(x^*) \\ &\le \frac{L}{2}\sum_{k=0}^{K}\left(\|x_{i_k} - x^*\|^2 - \|x_{i_k+1} - x^*\|^2 \right) \\ &= \frac{L}{2}\left(\|x_{i_0} - x^*\|^2 - \|x_{i_K+1} - x^*\|^2 + \sum_{k=1}^{K}\left(-\left\|x_{i_{k-1}+1} - x^*\right\|^2 + \|x_{i_k} - x^*\|^2 \right) \right) \\ &\le \frac{L}{2}(\|x_{i_0} - x^*\|^2 - \|x_{i_K+1} - x^*\|^2) \\ &\le \frac{L}{2}\|x_{i_0} - x^*\|^2 \\ &\le \frac{L}{2}\|x_0 - x^*\|^2, \end{aligned}$$

where the second inequality is due to (3.8) and $i_{k-1} < i_k$. Since $H(x_k)$ is nonincreasing, by noting that $N \le 2K + 1$, we have

$$H(x_N) - H(x^*) \le \frac{1}{K+1}\sum_{k=0}^{K} H(x_{i_k}) - H(x^*) \le \frac{L}{N}\|x_0 - x^*\|^2. \qquad \square$$

3.2. Line search with a modified Armijo rule.

An alternative way to choose the step size in the classical gradient descent type methods is a line search with the Armijo rule. A natural thought is then to extend the Armijo rule in our minimax problem (M) as in the proposed Algorithm 3.2. In particular, we set the following modified Armijo rule to choose the smallest nonnegative integer t such that the following inequality holds for the step size $\beta_k = \xi s^t$ with $0 < \xi \le 1$ and $0 < s < 1$:

$$f(x_k + \beta_k p_k) \le f(x_k) - \sigma \beta_k p_k^\top d_k, \tag{3.9}$$

Algorithm 3.2 Line search with a modified Armijo rule for problem (M).

```
Input: Parameters in the minimax problem (M) and $\rho > 0$
 1: Initialize $x_0$
 2: for $k = 0, 1, \dots,$ do
 3:     if $h_1(x_k) > h_2(x_k) + \rho$ then
 4:         set $d_k = -\nabla h_1(x_k)$
 5:     else if $h_1(x_k) < h_2(x_k) - \rho$ then
 6:         set $d_k = -\nabla h_2(x_k)$
 7:     else
 8:         set $d_k = -(\alpha g_1 + (1-\alpha) g_2)$, where $g_1 = \nabla h_1(x_k)$, $g_2 = \nabla h_2(x_k)$, and $\alpha$ is
            defined in the following three cases:
            $$\alpha = \begin{cases} 0 & \text{if } g_1^\top g_1 \geq g_1^\top g_2 \geq g_2^\top g_2, \\ 1 & \text{if } g_1^\top g_1 \leq g_1^\top g_2 \leq g_2^\top g_2, \\ \dfrac{g_2^\top g_2 - g_1^\top g_2}{g_1^\top g_1 + g_2^\top g_2 - 2 g_1^\top g_2} & \text{otherwise.} \end{cases}$$
 9:     end if
10:     if termination criterion is met then
11:         return
12:     end if
13:     Choose a step size $\beta_k > 0$ according to the modified Armijo rule (3.9)
14:     Update $x_{k+1} = x_k + \beta_k d_k$
15: end for
```

where $0 \leq \sigma \leq 0.5$ and d_k is the steepest descent direction defined in line 8 of Algorithm 3.2, which is similar to the steepest descent direction in Algorithm 3.1. In particular, we set the search direction $p_k = d_k$ at iterate k. Our numerical result in the next section shows that Algorithm 3.2 has a performance comparable to (or even better than) Algorithm 3.1. For completeness, we present the convergence result for Algorithm 3.2. Before that, we generalize the definition of a critical point to a (ρ, δ) critical point.

Definition 3.8. *A point x is called a (ρ, δ) critical point of*

$$H(x) = \max\{h_1(x), h_2(x)\}$$

if $\exists \, \|g\| < \delta$ for some $g \in \partial H_\rho(x)$, where $\partial H_\rho(x)$ is defined as follows:

1. $\partial H_\rho(x) = \{\alpha \nabla h_1(x) + (1-\alpha)\nabla h_2(x) : \alpha \in [0,1]\}$ *if* $|h_1(x) - h_2(x)| \leq \rho$,
2. $\partial H_\rho(x) = \{\nabla h_1(x)\}$ *if* $h_1(x) - h_2(x) > \rho$, *or*
3. $\partial H_\rho(x) = \{\nabla h_2(x)\}$ *if* $h_2(x) - h_1(x) > \rho$.

The following proposition shows the relationship between a critical point and a (ρ, δ) critical point. As this result is pretty obvious, we omit its proof.

Proposition 3.9. *Assume that $\{x_k\}$ is a sequence in $\mathbb{R}^n$, that $(\rho^t, \delta^t) \to (0, 0)$ for $t \to \infty$, and that there exists a positive integer $K(t)$ such that x_k is a (ρ^t, δ^t) critical point of $H(x)$ for all $k \geq K(t)$ and $t \geq 1$. Then, every accumulation point of the sequence $\{x_k\}$ is a critical point of $H(x)$.*

Slightly different from Algorithm 3.1, our goal in Algorithm 3.2 is to find a (ρ,δ) critical point. With Proposition 3.9, we conclude that Algorithm 3.2 outputs a solution that is sufficiently close to a critical point of $H(x)$.

THEOREM 3.10. *Assume that* (i) $d=-\mathrm{argmin}_{y\in\partial H_\rho(x_k)}\|y\|$ *with* $\rho>0$, (ii) *the termination criterion is* $\|d\|<\delta$ *for some* $\delta>0$, *and* (iii) x^* *is an optimal solution. Then the line search procedure in Algorithm* 3.2 *is well defined, i.e., the step size* β_k *is bounded from below in each iteration. And for any given positive numbers* ρ *and* δ, *Algorithm* 3.2 *generates a* (ρ,δ) *critical point in at most*

$$\frac{H(x_0)-H(x^*)}{\sigma s\min\{1/L,\xi,\frac{\rho}{2G^2}\}\delta^2}$$

iterations, where G is some positive constant depending only on the initial point and problem setting.

Proof. Suppose at iterations 1 to k we have $\|d\|\geq\delta$. Consider the following different cases with $\|d\|\geq\delta$.

1. If $|h_1(x_k)-h_2(x_k)|\leq\rho$, then, as assumed, $\|d\|\geq\delta$, and from the proof of Theorem 3.1 we know that $d=-(\alpha\nabla h_1(x_k)+(1-\alpha)\nabla h_2(x_k))$, where

$$\alpha=\mathrm{argmin}_{\alpha\in[0,1]}\|\alpha\nabla h_1(x_k)+(1-\alpha)\nabla h_2(x_k)\|$$

is just the parameter α we chose in Algorithm 3.2. We next show that the step size β_k is bounded from below such that

$$H(x_{k+1})-H(x_k)\leq-\sigma\beta_k d^\top d.$$

Since

$$\begin{aligned}H(x_{k+1})-H(x_k)&=\max\{h_1(x_{k+1}),h_2(x_{k+1})\}-\max\{h_1(x_k),h_2(x_k)\}\\&\leq\max\{h_1(x_{k+1})-h_1(x_k),h_2(x_{k+1})-h_2(x_k)\},\end{aligned}$$

it suffices to show that β_k is bounded from below such that for, both $i=1$ and $i=2$,

$$h_i(x_{k+1})-h_i(x_k)=\beta_k\nabla h_i(x_k)^\top d+\frac{1}{2}\beta_k^2d^\top A_id\leq-\sigma\beta_kd^\top d. \tag{3.10}$$

By noting that $-\nabla h_i^\top d\geq d^\top d$ from Remark 3.2 and $\lambda_{\max}(A_i)\leq L$, $i=1$ or 2, the inequality in (3.10) holds true for all $\beta_k\leq 2(1-\sigma)/L$. Then, due to (3.10), the step size chosen by the modified Armijo rule satisfies $\beta_k\geq s\min\{2(1-\sigma)/L,\xi\}$, since if $\xi>2(1-\sigma)/L$, there exists an integer t such that $2s(1-\sigma)/L\leq\xi s^t\leq2(1-\sigma)/L$ and if $\xi\leq2(1-\sigma)/L$, the Armijo condition holds at $t=0$. That is, β_k must be lower bounded. This further implies that

$$H(x_k)-H(x_{k+1})\geq\sigma\beta_kd^\top d\geq\sigma s\min\{2(1-\sigma)/L,\xi\}\delta^2.$$

2. If $h_1(x_k)-h_2(x_k)>\rho$ and $\|\nabla h_1(x_k)\|\geq\delta$, we have $d=-\nabla h_1(x_k)$. Because $H(x_k)$ is decreasing, under Condition 2.6 $h_1(x_k)+h_2(x_k)=\frac{1}{2}x_k^\top(A_1+A_2)x_k+(a_1+a_2)^\top x_k+r_1+r_2\leq2h_1(x_k)=2H(x_k)\leq2H(x_0)$, and thus x_k is bounded due to $A_1+A_2=2(Q_1+\frac{\lambda_1+\lambda_2}{2}Q_2)\succ0$. This further implies that $\nabla h_i(x_k)=A_ix_k+b_i$, $i=1$ or 2, is bounded for all k. So there exists some

positive constant depending only on the initial point and problem parameters such that $\|\nabla h_i(x_k)\| \leq G$, $i = 1$ or 2. Hence, $\|d\| \leq G$ because d is just $-\nabla h_1(x_k)$. Then we have, for any $\beta_k \leq 1/L$,

$$\begin{aligned} h_1(x_{k+1}) - h_1(x_k) &\leq \beta_k \nabla h_1(x_k)^\top d + \frac{1}{2}\beta_k^2 d^\top A_1 d \\ &\leq -\beta_k d^\top d + \frac{1}{2}\beta_k^2 L d^\top d \leq -\frac{1}{2}\beta_k d^\top d \end{aligned}$$

and

$$\begin{aligned} h_2(x_{k+1}) - h_2(x_k) &\leq \beta_k \nabla h_2(x_k)^\top d + \frac{1}{2}\beta_k^2 d^\top A_i d \\ &\leq \beta_k G \|d\| + \frac{1}{2}\beta_k^2 L \|d\|^2 \\ &\leq \beta_k G^2 \left(1 + \frac{1}{2}\beta_k L\right) \\ &\leq \frac{3}{2}\beta_k G^2. \end{aligned}$$

Note that, for all $\beta_k \leq \frac{\rho}{2G^2}$, we have $\frac{3}{2}\beta_k G^2 + \frac{1}{2}\beta_k d^\top d \leq \rho$. Thus, for $\beta_k \leq \min\{1/L,\ \frac{\rho}{2G^2}\}$, we have

$$\begin{aligned} &h_1(x_{k+1}) \leq h_1(x_k) - \frac{1}{2}\beta_k d^\top d, \\ &h_2(x_{k+1}) \leq h_2(x_k) + \frac{3}{2}\beta_k G^2 \leq h_1(x_k) - \rho + \frac{3}{2}\beta_k G^2 \leq h_1(x_k) - \frac{1}{2}\beta_k d^\top d. \end{aligned}$$

Hence, for $\beta_k \leq \min\{1/L,\ \frac{\rho}{2G^2}\}$, we have

$$\begin{aligned} H(x_{k+1}) - H(x_k) &= \max\{h_1(x_{k+1}), h_2(x_{k+1})\} - h_1(x_k) \\ &= \max\{h_1(x_{k+1}) - h_1(x_k),\ h_2(x_{k+1}) - h_1(x_k)\} \\ &\leq -\frac{1}{2}\beta_k d^\top d. \end{aligned}$$

So the Armijo rule implies $\beta_k \geq s\min\{1/L, \xi, \frac{\rho}{2G^2}\}$ by reasoning similar to the previous case. That is, β_k is lower bounded. Then, according to the modified Armijo rule, we have

$$H(x_k) - H(x_{k+1}) \geq \sigma\beta_k d^\top d \geq \sigma s \min\left\{1/L, \xi, \frac{\rho}{2G^2}\right\}\delta^2. \tag{3.11}$$

3. Symmetrically, the case when $h_2(x_k) - h_1(x_k) > \rho$ yields the same result as in (3.11).

The above three cases show that $H(x_k) - H(x_{k+1}) \geq \sigma s \min\{1/L, \xi,\ \frac{\rho}{2G^2}\}\delta^2$ (as $1 - \sigma \geq 1/2$, the decrease in case 1 also admits this bound). Since the decrease in each iterate is larger than $\sigma s \min\{1/L, \xi, \frac{\rho}{2G^2}\}\delta^2$, the total number of iterations is bounded by

$$\frac{H(x_0) - H(x^*)}{\sigma s \min\{1/L, \xi, \frac{\rho}{2G^2}\}\delta^2}. \qquad \square$$

At the current stage, we cannot demonstrate a theoretical convergence rate for Algorithm 3.2 that is as good as the sublinear rate $O(1/\rho)$ for Algorithm 3.1 in Theorem 3.7. But our numerical tests show that Algorithm 3.2 converges as fast as Algorithm 3.1. Proposition 3.9 and Theorem 3.10 offer our main convergence result for Algorithm 3.2 as follows.

THEOREM 3.11. *Assume that* $(\phi_k, \psi_k) \to 0$ *and that* $\{x^{(k)}\}$ *is a sequence of solutions generated by Algorithm* 3.2 *with* $\rho = \phi_k$ *and* $\delta = \psi_k$. *Then any accumulation point of* $\{x^{(k)}\}$ *is an optimal solution of problem* (M).

4. Numerical tests.

In this section, we illustrate the efficiency of our algorithm with numerical experiments. All the numerical tests were implemented in MATLAB 2016a, 64 bit and were run on a Linux machine with 48 GB RAM, 2,600 MHz CPU and 64-bit CentOS release 7.1.1503. We compare both Algorithms 3.1 and 3.2 with the extended Rendl–Wolkowicz (ERW) algorithm in [40]. We disable the parallel setting in MATLAB for a fair comparison. If the parallel setting is allowed, our algorithm shows a significant improvement, while the ERW algorithm does not.

We use the following test problem, as in [40], to show the efficiency of our algorithms:

$$\text{(IP)} \qquad \begin{aligned} &\min\ x^\top Ax - 2a^\top x \\ &\text{s.t. } c_1 \le x^\top Bx \le c_2, \end{aligned}$$

where A is an $n \times n$ positive definite matrix and B is an $n \times n$ (nonsingular) symmetric indefinite matrix. We first reformulate problem (IP) to a formulation of problem (P) in the following procedure, which is motivated by [40] (the proof in [40] is also based on the monotonicity of $\gamma(\lambda)$, which is defined in section 2.1), in order to apply the CQR for problem (P) and then invoke Algorithms 3.1 and 3.2 to solve the CQR.

THEOREM 4.1. *Let* $x_0 = -A^{-1}a$. *Then the following hold.*

1. *If* $x_0^\top Bx_0 < c_1$, *problem* (IP) *is equivalent to*

$$(\text{IP}_1) \qquad \min\{x^\top Ax - 2a^\top x :\ \text{s.t. } c_1 \le x^\top Bx\}.$$

2. *Else if* $c_1 \le x_0^\top Bx_0 \le c_2$, *problem* (IP) *admits an interior solution* x_0.
3. *Otherwise* $c_2 < x_0^\top Bx_0$, *problem* (IP) *is equivalent to*

$$(\text{IP}_2) \qquad \min\{x^\top Ax - 2a^\top x :\ \text{s.t. } x^\top Bx \le c_2\}.$$

Proof. Item 2 is obvious. Item 1 and item 3 are symmetric. So, in the following, we only prove item 1.

In our problem set, matrix A is positive definite and B is indefinite. Hence, in the definition $I_{PSD} = \{\lambda : Q_1 + \lambda Q_2 \succeq 0\}$, we have $\lambda_1 < 0$, $\lambda_2 > 0$. Thus, from Case 1 in [40, section 2.2.2] we know, when $x_0^\top Bx_0 < c_1$, that problem (IP) is equivalent to

$$(\text{EP}_1) \qquad \min\{x^\top Ax - 2a^\top x : \text{s.t. } c_1 = x^\top Bx\}.$$

Since $x_0^\top Bx_0 < c_1$, the optimal solution of (IP_1) must be at its boundary [35]. This further yields the equivalence between problems (IP) and (IP_1). ◻

Theorem 4.1 helps us solve problem (IP) as an inequality constrained GTRS instead of solving two GTRSs with equality constraints. Before showing the numerical results, let us illustrate some functions used in our initialization. To obtain the CQR, the generalized eigenvalue problem is solved by `eigifp` in MATLAB, which was developed in [17] for computing the maximum generalized eigenvalues for sparse definite matrix pencils. In our numerical setting `eigifp` is usually faster than the MATLAB function `eigs`, though `eigs` will outperform `eigifp` when the condition number is large or the density is low. We use the MATLAB commands `sprandsym(n,density,cond,2)` and `sprandsym(n,density)` to generate Q_1

and Q_2. We set the density of matrices at 0.01 and use three levels of condition number for matrix Q_1, i.e., 10, 100, and 1,000 and, in such settings, `eigifp` always dominates `eigs` (this may be because `eigs` is developed for computing extreme generalized eigenvalues for arbitrary matrices and does not utilize the definiteness and symmetry properties of the matrix pencils in our problem setting). In general, the main cost in estimating L is to compute the maximum eigenvalues of matrices A_1 and A_2, which may be time consuming for large-scale matrices. To conquer this difficulty, we can estimate a good upper bound with very cheap cost instead. Specially, we can run the function `eigifp` with precision 0.1, which is much more efficient than computing the true maximum eigenvalue with `eigifp`, and, assuming M is the output, $M + 0.1$ is then a good upper bound for L. In our numerical tests, we just use `eigifp` to estimate L since our main goal is to illustrate the efficiency of Algorithm 3.2. In Algorithm 3.1, to avoid some numerical accuracy problem, we approximate $h_1(x_k) = h_2(x_k)$ by $|h_1(x_k) - h_2(x_k)|/(|h_1(x_k)| + |h_2(x_k)|) \le \epsilon_1$. Also, we use $|h_1(x_k) - h_2(x_k)|/(|h_1(x_k)| + |h_2(x_k)|) \le \epsilon_1$ instead of $|h_1(x_k) - h_1(x_k)| \le \rho$ in Algorithm 3.2 for stableness consideration. In our numerical tests for both Algorithms 3.1 and 3.2, we use the following termination criteria (if any one of the following three conditions is met, we terminate our algorithm), which are slightly different from the presented algorithms for robust consideration:

1. $H(x_{k-1}) - H(x_k) < \epsilon_2$,
2. $|h_1(x_k) - h_2(x_k)|/(|h_1(x_k)| + |h_2(x_k)|) \le \epsilon_1$ and
$$\min_{\alpha\in[0,1]} \|\alpha \nabla h_1(x_k) + (1-\alpha)\nabla h_2(x_k)\| \le \epsilon_3,$$
3. $\|\nabla h_i(x_k)\| \le \epsilon_3$ and $|h_1(x_k) - h_2(x_k)|/(|h_1(x_k)| + |h_2(x_k)|) > \epsilon_1$ with $i \in \{1, 2\}$,

where ϵ_1, ϵ_2, and $\epsilon_3 > 0$ are some small positive numbers for termination of the algorithm. In particular, we set $\epsilon_1 = 10^{-8}$, $\epsilon_2 = 10^{-11}$, and $\epsilon_3 = 10^{-8}$ in Algorithm 3.1, and $\epsilon_1 = 10^{-8}$, $\epsilon_2 = 10^{-11}$, $\epsilon_3 = 10^{-8}$, $s = 0.5$, $\sigma = 10^{-4}$, and $\xi = 1$ (for the modified Armijo rule) in Algorithm 3.2.

To improve the accuracy of the solution, we apply the Newton refinement process in [3, section 4.1.2]. More specifically, assuming x^* is the solution returned by our algorithm, we update x^* by

$$\delta = \frac{(x^*)^\top B x^*}{2\|Bx^*\|^2} Bx^*, \quad x^* = x^* - \delta.$$

In general, the ERW algorithm can achieve a higher precision than our method (after the Newton refinement process); the precision in the former is about 10^{-14}, while ours is slightly less precise. Letting v_1 denote the objective value of the ERW algorithm and v_2 denote the objective value of our algorithm, we have at least $|v_2 - v_1|/|v_1| \approx 10^{-10}$ for most cases. The iteration number reduces to 1/5 if we reduce the precision from $\epsilon_1 = 10^{-8}$, $\epsilon_2 = 10^{-11}$, $\epsilon_3 = 10^{-8}$ to $\epsilon_1 = 10^{-5}$, $\epsilon_2 = 10^{-8}$, $\epsilon_3 = 10^{-5}$. This observation seems reasonable as our method is just a first-order method.

We report our numerical results in Table 1. We use "Alg1" and "Alg2" to denote Algorithms 3.1 and 3.2, respectively. For each n and each condition number, we generate 10 easy case and 10 hard case 1 examples. The reader should refer to [40, Table 1] for detailed definitions of the easy case and hard cases 1 and 2. There are small differences in the definitions of easy and hard cases between [40] and [35]. Our analysis in the above sections uses the definitions in [35]. In fact, the easy case and

TABLE 1
Numerical results for positive definite A and indefinite B.

		Easy case								Hard case 1							
Cond	n	Alg1		Alg2		time_{eig}	ERW			Alg1		Alg2		time_{eig}	ERW		
		Iter	Time	Iter	Time		Iter	Time	Fail	Iter	Time	Iter	Time		Iter	Time	Fail
10	10000	90	**1.03**	109.3	1.24	1.45	5.9	4.89	0	1490	16.7	609.6	**6.81**	1.19	6	11.1	1
10	20000	52	**2.82**	72.2	3.91	9.20	6.8	25.1	0	530.3	27.9	313.9	**16.7**	7.56	6.5	53.9	0
10	30000	60.9	**9.81**	83.2	13.4	25.2	6.6	75.0	0	1014.6	157	270.6	**41.0**	30.1	7.3	170	1
10	40000	58.3	**17.1**	95.2	27.8	49.7	6.8	153	0	1866.4	520	782.7	**219**	54.0	7.1	356	1
100	10000	417.7	**4.26**	424.9	4.34	3.99	5.9	11.4	0	3328.2	33.9	1131.6	**13.6**	3.63	5.7	24.6	3
100	20000	474.3	24.6	342.4	**17.8**	18.4	6.1	69.4	0	6494.9	350	1410	**76.8**	42.2	6.4	123	5
100	30000	196.9	28.0	162.1	**23.1**	51.8	6.2	147	0	2836.6	420	1197.9	**176**	44.2	5.2	388	0
100	40000	135.8	40.1	114.7	**33.9**	153.6	6.3	309	0	906.7	257	506.1	**143**	173.5	6.5	639	0
1000	10000	4245	44.7	1706.7	**17.8**	14.2	5.3	56.7	0	25982.6	261	5090.7	**51.3**	24.0	5.75	81.1	6
1000	20000	4177.3	216	1182.7	**61.2**	70.8	6.10	368	0	26214.8	1360	2726.8	**139**	98.1	5.8	346	5
1000	30000	2023.8	289	813.7	**116**	189	5.9	1220	0	15311.4	2190	2591.9	**385**	195	5.8	1530	3
1000	40000	2519.8	652	1003	**301**	640.9	6.8	2960	0	8735.8	3060	1343	**1020**	853	6.25	3280	2

hard case 1 are the easy case mentioned above, and hard case 2 is the hard case mentioned in the above sections and [35]. We use the notation "time" to denote the average CPU time (in units of seconds) and "iter" to denote the average iteration numbers for all three algorithms. For "Alg1" and "Alg2," "time" is just the average CPU time for Algorithms 3.1 and 3.2, respectively. The notation "time_{eig}" denotes the average CPU time for computing the generalized eigenvalue for our algorithm. So the total time for solving problem (P) should be the summation of the time to reformulate (P) into (M) and the "time" of Algorithm 3.1 or 3.2, whose main cost is just "time" + "time_{eig}." And "fail" denotes the failure times in the 10 examples in each case for the ERW algorithm. One reason for the failures may be that the ERW algorithm terminates in 10 iterations even when it does not find a good approximated solution. We point out that, for randomly generated test examples, our method always succeeds in finding an approximated solution to the prescribed precision, while the ERW algorithm fails frequently in hard case 1. Another disadvantage of the ERW algorithm is the requirement of an efficient prior estimation of the initialization, which is unknown in general. In our numerical test, we assume that such an initialization is the same as that given in [40].

We also need to point out that in hard case 2, our algorithms do not outperform the ERW algorithm, which uses the shift and deflation technique. The main time cost of the shift and deflate operation is the computation of the extreme generalized eigenvalue of the matrix pencil (A, B) and its corresponding generalized eigenvectors. In the test instances, as the dimension of the eigenspace of the extreme generalized eigenvalue is one, the shift and deflation technique directly finds the optimal solution by calling `eigifp` once. Our algorithm reduces to an unconstrained quadratic minimization in hard case 2. However, the condition number of this unconstrained quadratic minimization is so large that our algorithm performs badly as a classical gradient method. To remedy this disadvantage, we can add a step with an almost free time cost, which claims that either we are in hard case 2 and outputs an optimal solution or we are in the easy case or hard case 1. Recall that the hard case (or equivalently, hard case 2) states that $b_1 + \lambda^* b_2$ is orthogonal to the null space of $Q_1 + \lambda^* Q_2$, which means that λ^* must be a boundary point of I_{PSD}. Suppose $\lambda_i = \lambda^*$. Then we must have that $x^* = \arg\min H(x)$ and $H(x^*) = h_i(x^*)$ for some $i = 1$ or 2. In fact, if $\nabla h_i(x) = 0$ and $h_i(x) \geq h_j(x)$, $j \in \{1, 2\}/\{i\}$, for some x, then x is optimal and we are in the hard case. So $\nabla h_i(x) = 0$ and $h_i(x) \geq h_j(x)$ is sufficient and necessary for x to be optimal to (M) and to be in the hard case. Hence, we can construct an optimal solution for problem (M) as $\bar{x} = (Q_1 + \lambda_i Q_2)^{\dagger}(b_1 + \lambda_i b_2) + \sum_i^k \alpha_j v_j$ (where $A^{\dagger}$ denotes the *Moore–Penrose* pseudoinverse of A) if v_j, $j = 1, \ldots, k$, are the generalized eigenvectors of matrix pencil (Q_1, Q_2) with respect to the generalized eigenvalue λ_i such that $h_i(\bar{x}) \geq h_j(\bar{x})$ and $\alpha \geq 0$. This is equal to identifying whether a small dimensional convex quadratic programming problem (with variable α) has an optimal value less than $h_i((Q_1+\lambda_i Q_2)^\dagger(b_1+\lambda_i b_2))$. And if such an α does not exist, we are in the easy case (or equivalently, the easy case or hard case 1). This technique is very similar to the shift and deflation technique in [15, 40]. Hence, we can solve hard case 2 within almost the same CPU time as the ERW algorithm. So we do not make further comparisons for hard case 2.

Our numerical tests show that both Algorithms 3.1 and 3.2 are much more efficient than the ERW algorithm in the easy case and for most cases in hard case 1. The efficiency of our algorithms is mainly due to the fact that we call the generalized eigenvalue solver only once, and every iteration involves several matrix vector products (which are very cheap for sparse matrices). We also note that, in the easy case, Algorithm 3.1 is faster than Algorithm 3.2 when the condition number is small, and slower than Algorithm 3.2 when the condition number is large. This may be because Algorithm 3.2 is equipped with the modified Armijo rule, which makes it more aggressive in choosing the step size and thus yields a fast convergence. In hard case 1, Algorithm 3.2 is still much more efficient than the ERW algorithm, while Algorithm 3.1 is slower than the ERW algorithm in about half the cases. This is because Algorithm 3.2 has a moderate number of iterations due to the aggressiveness in choosing the step size, and Algorithm 3.1 has a much larger number of iterations for these cases. Moreover, our algorithms always succeed, while the ERW algorithm fails frequently in hard case 1. A more detailed analysis with condition number for Algorithm 3.1 will be given below.

We note that several (of the 10) examples in the easy cases admit a much larger iteration number than average. This motivates us to analyze the main factor that affects the convergence rate (reflected by the iteration number) of Algorithm 3.1 (the analysis for Algorithm 3.2 seems hard due to the nonsmoothness of the problem). We then find that the main factor is $\sqrt{\lambda_{\max\alpha}/2\lambda^2_{\min nnz\alpha}}$, as evidenced by the fact that examples in the easy case and hard case 1 with more iterates all have a larger $\sqrt{\lambda_{\max\alpha}/2\lambda^2_{\min nnz\alpha}}$, where $\lambda_{\max\alpha}$ denotes the maximum eigenvalue of matrix $\alpha A_1 + (1-\alpha)A_2$ and $\lambda_{\min nnz\alpha}$ denotes the smallest nonzero eigenvalue of matrix $\alpha A_1 + (1-\alpha)A_2$, with α being defined in Theorem 3.1 in the last iteration. In fact, when $x^k \to x^* \in \{x : \partial H(x) = 0\}$ (in our examples, the optimal solution is unique), let the value of α at iterate k be α^k. Then $\alpha^k \to \alpha^*$, where α^* is the solution of $\alpha\nabla h_1(x^*) + (1-\alpha)\nabla h_2(x^*) = 0$. From the definition of the KL exponent, we have

$$C \times \min_\alpha \left\|\alpha\nabla h_1(x^k) + (1-\alpha)\nabla h_2(x^k)\right\| \geq |H(x^k) - H(x^*)|^{1/2}.$$

Intuitively, the smallest value of C should be at least

$$\frac{|H(x^k)-H(x^*)|^{\frac{1}{2}}}{\min_\alpha \|\alpha\nabla h_1(x^k) + (1-\alpha)\nabla h_2(x^k)\|} \\ \to \frac{|\alpha(h_1(x^k)-h_2(x^*)) + (1-\alpha)(h_1(x^k)-h_2(x^*))|^{\frac{1}{2}}}{\min_\alpha \|\alpha\nabla h_1(x^k) + (1-\alpha)\nabla h_2(x^k)\|},$$

which is upper bounded by $\sqrt{\lambda_{\max\alpha}/2\lambda^2_{\min nnz\alpha}}$. Thus, the asymptotic value of C can be seen as roughly $\sqrt{\lambda_{\max\alpha}/2\lambda^2_{\min nnz\alpha}}$. Hence, both the easy case and hard case 1 admit local linear convergence and the convergence rate is

$$\left(\sqrt{1-\frac{1}{2C^2L}}\right)^k = \left(\sqrt{1-\frac{\lambda_{\min nnz\alpha}^2}{L\lambda_{\max\alpha}}}\right)^k$$

from Theorem 3.5. We also observe from our numerical tests that in most cases the values of $\lambda_{\max\alpha}$ are similar and that $\lambda_{\min nnz\alpha}$ in the easy case is much larger than $\lambda_{\min nnz\alpha}$ in hard case 1, and $\lambda_{\max\alpha}$ in the easy case is very close to $\lambda_{\max\alpha}$ in hard case 1. Hence, $\sqrt{1-\lambda_{\min nnz\alpha}^2/(L\lambda_{\max\alpha})}$ in the easy case is usually much smaller than that in hard case 1. (As Q_2 is random in our setting, the larger the condition number of Q_1, the greater the expectation of $\sqrt{1-\lambda_{\min nnz\alpha}^2/(L\lambda_{\max\alpha})}$.) This explains why the condition number of matrix Q_1 measures, to a large degree, the hardness of our algorithms in solving problem (M). Since the easy case has a smaller $\sqrt{1-(\lambda_{\min nnz\alpha}^2/L\lambda_{\max\alpha})}$ than hard case 1 for the same condition number and problem dimension, the easy case can be solved faster than hard case 1. This coincides with our numerical results, i.e., the easy case admits a smaller number of iterations than hard case 1.

We also tried to apply MOSEK [37] to solve the CQR. But our numerical results showed that MOSEK is much slower than both our algorithms and the ERW algorithm, which took about 833 seconds for the easy case and 960 seconds for hard case 1 with $n = 10{,}000$ and cond $= 10$. So, we did not run further numerical experiments with MOSEK. We also tested the SOCP reformulation [5] under the SD condition of the quadratic forms of the GTRS and the DB algorithm in [42] based on the SD condition of the quadratic forms. The SD condition naturally holds for problem (IP) when A is positive definite. Our preliminary result shows that our method is much more efficient than the two methods based on SD when $n \geq 10{,}000$ and density $= 0.01$ and thus we do not report the numerical comparison in this paper. We believe this efficiency is mainly because the simultaneous diagonalization of the matrices involves a matrix inverse, a matrix-matrix product, a full Cholesky decomposition, and a spectral decomposition (of a dense matrix), which is more time consuming than the matrix vector product operations in our algorithm. Hence, we do not report the numerical results based on the SD technique here.

5. Concluding remarks.

In this paper, we derived a simple convex quadratic reformulation for the GTRS that only involves a linear objective function and two convex quadratic constraints under mild assumptions. We further reformulated the CQR as an unconstrained minimax problem under Condition 2.6, which is the case of interest. The minimax reformulation is a well-structured convex (albeit nonsmooth) problem. By investigating its inherent structure, we proposed two efficient matrix-free algorithms to solve this minimax reformulation. Moreover, we offered a theoretical guarantee of the global sublinear convergence rate for both our algorithms and demonstrated a local linear convergence rate for Algorithm 3.1 by proving the KL property for the minimax problem with an exponent of 1/2 under some mild conditions. Our numerical results clearly demonstrate that our algorithms outperformed the state-of-the-art algorithm for the GTRS.

In future research, we would like to show whether the CQR and the minimax reformulation and the algorithms for the minimax problem can be extended to solve GTRSs with additional linear constraints. As the numerical observations and preliminary analytical observation in section 4 indicate that our algorithms have similar performance to unconstrained quadratic minimization, i.e., both algorithms admit a locally linear convergence rate with the steepest descent method, we would like to generalize existing algorithms that are efficient in solving unconstrained quadratic minimization to solve our minimax reformulation, e.g., the conjugate gradient method or Nesterov's accelerated gradient descent algorithm. Another line of future research is to investigate whether our algorithm can be extended to general minimax problems with more (but a finite number of) functions. It would also be interesting to verify whether the KL property still holds and whether the KL exponent is still 1/2 when more functions are involved.

Appendix A. Discussion of Assumption 2.3. Assume that Assumptions 2.1 and 2.2 hold. Define $\mathrm{Range}(Q_1) + \mathrm{Range}(Q_2)$ as the summation of two subspaces, i.e., $\mathrm{Range}(Q_1) + \mathrm{Range}(Q_2) = \{u + v : u \in \mathrm{Range}(Q_1) \text{ and } v \in \mathrm{Range}(Q_1)\}$. When Assumption 2.3 fails, we have either $\mathrm{Null}(Q_1)\cap\mathrm{Null}(Q_2) \neq \{0\}$ and $b_i \in \mathrm{Range}(Q_1)+\mathrm{Range}(Q_2)$, $i = 1, 2$, or $\mathrm{Null}(Q_1)\cap\mathrm{Null}(Q_2) \neq \{0\}$ and $b_i \notin \mathrm{Range}(Q_1)+\mathrm{Range}(Q_2)$ for some $i \in \{1, 2\}$. The former case can be reduced to one that satisfies Assumption 2.3 using null space representation for $\binom{Q_1}{Q_2}x = 0$. We next show that the latter case leads to the situation where problem (P) can be either solved analytically or unbounded from below. To see this, note that the system in Lemma 2.5 is equivalent to the existence of a $\lambda^* \in I_{PSD}$ such that $b_1 + \lambda^* b_2 \in \mathrm{Range}(Q_1 + \lambda^* Q_2)$, which implies $(b_1 + \lambda^* b_2) \perp \bigcap_{i=1,2} \mathrm{Null}(Q_i)$. And λ^* can be derived analytically by solving the univariable linear system $\mathcal{N}(b_1 + \lambda^* b_2) = 0$, where $\mathcal{N}$ is a linear operator that projects a vector to the space of $\bigcap_{i=1,2} \mathrm{Null}(Q_i)$. If the linear system $\mathcal{N}(b_1 + \lambda^* b_2) = 0$ has no solution, the system in Lemma 2.4 has no solution, and hence problem (P) is unbounded from below. In summary, our assumption is made w.l.o.g.

Appendix B. Sion's minimax theorem [43].

Lemma B.1. *Let X be a compact convex subset of a finite-dimensional Euclidean space, Y be a convex subset of a finite-dimensional Euclidean space, and f be a real-valued function on $X \times Y$. If*

1. *$f(x, \cdot)$ is upper semicontinuous and quasi-concave on Y $\forall x \in X$ and*
2. *$f(\cdot, y)$ is lower semicontinuous and quasi-convex on X $\forall y \in Y$,*

then

$$\min_{x\in X} \max_{y\in Y} f(x, y) = \max_{y\in Y} \min_{x\in X} f(x, y).$$

When f is bilinear, which is the case we used in the proof of Theorem 3.1, items 1 and 2 in the above lemma are satisfied.

Acknowledgments. The authors would like to thank Dr. Zirui Zhou and Dr. Huikang Liu for their insightful discussions. The authors would also like to thank the two anonymous referees for their invaluable comments that improved the quality of the paper a lot, and especially for bringing the reference [47] to the authors' attention.

REFERENCES

[1] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, Princeton, NJ, 2009.

[2] S. Adachi, S. Iwata, Y. Nakatsukasa, and A. Takeda, *Solving the trust-region subproblem by a generalized eigenvalue problem*, SIAM J. Optim., 27 (2017), pp. 269–291, https://doi.org/10.1137/16M1058200.

[3] S. Adachi and Y. Nakatsukasa, *Eigenvalue-based algorithm and analysis for nonconvex QCQP with one constraint*, Math. Program., 173 (2019), pp. 70–116, https://doi.org/10.1007/s10107-017-1206-8.

[4] H. Attouch and J. Bolte, *On the convergence of the proximal algorithm for nonsmooth functions involving analytic features*, Math. Program., 116 (2009), pp. 5–16.

[5] A. Ben-Tal and D. den Hertog, *Hidden conic quadratic representation of some nonconvex quadratic optimization problems*, Math. Program., 143 (2014), pp. 1–29.

[6] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust Optimization*, Princeton University Press, Princeton, NJ, 2009.

[7] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, MOS-SIAM Ser. Optim. 2, SIAM, Philadelphia, PA, 2001.

[8] A. Ben-Tal and M. Teboulle, *Hidden convexity in some nonconvex quadratically constrained quadratic programming*, Math. Program., 72 (1996), pp. 51–63.

[9] J. Bolte, T. P. Nguyen, J. Peypouquet, and B. W. Suter, *From error bounds to the complexity of first-order descent methods for convex functions*, Math. Program., 165 (2017), pp. 471–507.

[10] S. Boyd and A. Mutapcic, *Subgradient Methods*, Lecture notes for EE364b, Stanford University, Winter Quarter, 2006/7.

[11] S. Burer and F. Kılınç-Karzan, *How to convexify the intersection of a second order cone and a nonconvex quadratic*, Math. Program., 162 (2017), pp. 393–429.

[12] A. R. Conn, N. I. Gould, and P. L. Toint, *Trust Region Methods*, MOS-SIAM Ser. Optim. 1, SIAM, Philadelphia, PA, 2000.

[13] J.-M. Feng, G.-X. Lin, R.-L. Sheu, and Y. Xia, *Duality and solutions for quadratic programming over single non-homogeneous quadratic constraint*, J. Global Optim., 54 (2012), pp. 275–293.

[14] O. E. Flippo and B. Jansen, *Duality and sensitivity in nonconvex quadratic optimization over an ellipsoid*, European J. Oper. Res., 94 (1996), pp. 167–178.

[15] C. Fortin and H. Wolkowicz, *The trust region subproblem and semidefinite programming*, Optim. Methods Softw., 19 (2004), pp. 41–67.

[16] B. Gao, X. Liu, X. Chen, and Y.-x. Yuan, *On the Łojasiewicz Exponent of the Quadratic Sphere Constrained Optimization Problem*, preprint, https://arxiv.org/abs/1611.08781, 2016.

[17] G. Golub and Q. Ye, *An inverse free preconditioned Krylov subspace method for symmetric generalized eigenvalue problems*, SIAM J. Sci. Comput., 24 (2002), pp. 312–334, https://doi.org/10.1137/S1064827500382579.

[18] N. I. M. Gould, D. P. Robinson, and H. S. Thorne, *On solving trust-region and other regularised subproblems in optimization*, Math. Program. Comput., 2 (2010), pp. 21–57, https://doi.org/10.1007/s12532-010-0011-7.

[19] M. Grant and S. Boyd, *CVX: MATLAB Software for Disciplined Convex Programming*, Version 2.1, http://cvxr.com/cvx (March 2014).

[20] C.-H. Guo, N. J. Higham, and F. Tisseur, *An improved arc algorithm for detecting definite Hermitian pairs*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 1131–1151.

[21] E. Hazan and T. Koren, *A linear-time algorithm for trust region problems*, Math. Program., 158 (2016), pp. 363–381, https://doi.org/10.1007/s10107-015-0933-y.

[22] H. Hmam, *Quadratic Optimization with One Quadratic Equality Constraint*, Technical report DSTO-TR-2416, Warfare and Radar Division, Defence Science and Technology Organisation, Australia, 2010.

[23] N. Ho-Nguyen and F. Kılınç-Karzan, *A second-order cone based approach for solving the trust-region subproblem and its variants*, SIAM J. Optim., 27 (2017), pp. 1485–1512.

[24] Y. Hsia, G.-X. Lin, and R.-L. Sheu, *A revisit to quadratic programming with one inequality quadratic constraint via matrix pencil*, Pac. J. Optim., 10 (2014), pp. 461–481.

[25] K. Huang and N. D. Sidiropoulos, *Consensus-ADMM for general quadratically constrained quadratic programming*, IEEE Trans. Signal Process., 64 (2016), pp. 5297–5310.

[26] IBM, *ILOG CPLEX Optimizer*, http://www.ibm.com/software/commerce/optimization/cplex optimizer, 2017.

[27] V. Jeyakumar and G. Li, *Trust-region problems with linear inequality constraints: Exact SDP relaxation, global optimality and robust optimization*, Math. Program., 147 (2014), pp. 171–206.

[28] R. Jiang, D. Li, and B. Wu, *SOCP reformulation for the generalized trust region subproblem via a canonical form of two symmetric matrices*, Math. Program., 169 (2018), pp. 531–563.

[29] G. Li, B. S. Mordukhovich, and T. Pham, *New fractional error bounds for polynomial systems with applications to Hölderian stability in optimization and spectral theory of tensors*, Math. Program., 153 (2015), pp. 333–362.

[30] G. Li and T. K. Pong, *Calculus of the exponent of Kurdyka–Łojasiewicz inequality and its ap-*

plications to linear convergence of first-order methods, Found. Comput. Math., 18 (2018), pp. 1199–1232, https://doi.org/10.1007/s10208-017-9366-8.

[31] W. Li, *Error bounds for piecewise convex quadratic programs and applications*, SIAM J. Control Optim., 33 (1995), pp. 1510–1529.

[32] H. Liu, W. Wu, and A. M.-C. So, *Quadratic optimization with orthogonality constraints: Explicit Lojasiewicz exponent and linear convergence of line-search methods*, in Proceedings of the International Conference on Machine Learning 2016, Proc. Mach. Learn. Res. 48, 2016, pp. 1158–1167; available at http://proceedings.mlr.press/v48/.

[33] Z.-Q. Luo and J. F. Sturm, *Error bounds for quadratic systems*, in High Performance Optimization, Appl. Optim. 33, Springer, New York, 2000, pp. 383–404.

[34] J. M. Martínez, *Local minimizers of quadratic functions on Euclidean balls and spheres*, SIAM J. Optim., 4 (1994), pp. 159–176.

[35] J. J. Moré, *Generalizations of the trust region problem*, Optim. Methods Softw., 2 (1993), pp. 189–209.

[36] J. J. Moré and D. C. Sorensen, *Computing a trust region step*, SIAM J. Sci. and Stat. Comput., 4 (1983), pp. 553–572.

[37] MOSEK ApS, *The MOSEK Optimization Software*, Version 7.1.0.63, http://www.mosek.com, 2017.

[38] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, Appl. Optim. 87, Springer, New York, 2003.

[39] I. Pólik and T. Terlaky, *A survey of the S-lemma*, SIAM Rev., 49 (2007), pp. 371–418.

[40] T. K. Pong and H. Wolkowicz, *The generalized trust region subproblem*, Comput. Optim. Appl., 58 (2014), pp. 273–322.

[41] F. Rendl and H. Wolkowicz, *A semidefinite framework for trust region subproblems with applications to large scale minimization*, Math. Program., 77 (1997), pp. 273–299.

[42] M. Salahi and A. Taati, *An efficient algorithm for solving the generalized trust region subproblem*, Comput. Appl. Math., 37 (2018), pp. 395–413.

[43] M. Sion, *On general minimax theorems*, Pacific J. Math., 8 (1958), pp. 171–176.

[44] R. J. Stern and H. Wolkowicz, *Indefinite trust region subproblems and nonsymmetric eigenvalue perturbations*, SIAM J. Optim., 5 (1995), pp. 286–313.

[45] J. F. Sturm and S. Zhang, *On cones of nonnegative quadratic functions*, Math. Oper. Res., 28 (2003), pp. 246–267.

[46] J. Wang and Y. Xia, *A linear-time algorithm for the trust region subproblem based on hidden convexity*, Optim. Lett., 11 (2017), pp. 1639–1646.

[47] T. Wang and J.-S. Pang, *Global error bounds for convex quadratic inequality systems*, Optimization, 31 (1994), pp. 1–12.

[48] V. A. Yakubovich, *S-procedure in nonlinear control theory*, Vestn. Lenin. Univ. Mat. Mekh. Astron., 1 (1971), pp. 62–77 (in Russian); Vestn. Leningrad Univ. Math., 4 (1977), pp. 73–93 (in English).

[49] Y. Ye, *A new complexity result on minimization of a quadratic function with a sphere constraint*, in Recent Advances in Global Optimization, Princeton University Press, Princeton, NJ, 1992, pp. 19–31.

[50] Y. Yuan, *Recent advances in trust region algorithms*, Math. Program., 151 (2015), pp. 249–281.

[51] H. Zhang, A. R. Conn, and K. Scheinberg, *A derivative-free algorithm for least-squares minimization*, SIAM J. Optim., 20 (2010), pp. 3555–3576.

[52] Z. Zhou and A. M.-C. So, *A unified approach to error bounds for structured convex optimization problems*, Math. Program., 165 (2017), pp. 689–728.