SIAM J. OPTIM.
Vol. 30, No. 1, pp. 915–932




# A LINEAR-TIME ALGORITHM FOR GENERALIZED TRUST REGION SUBPROBLEMS*

RUJUN JIANG† AND DUAN LI‡

**Abstract.** In this paper, we provide the first provable linear-time (in terms of the number of nonzero entries of the input) algorithm for approximately solving the generalized trust region subproblem (GTRS) of minimizing a quadratic function over a quadratic constraint under some regularity condition. Our algorithm is motivated by and extends a recent linear-time algorithm for the trust region subproblem by Hazan and Koren [*Math. Program.*, 158 (2016), pp. 363–381]. However, due to the nonconvexity and noncompactness of the feasible region, such an extension is nontrivial. Our main contribution is to demonstrate that under some regularity condition, the optimal solution is in a compact and convex set and lower and upper bounds of the optimal value can be computed in linear time. Using these properties, we develop a linear-time algorithm for the GTRS.

**Key words.** generalized trust region subproblem, semidefinite programming, linear-time complexity, approximation algorithms

**AMS subject classifications.** 90C20, 90C22, 90C26, 68W25

**DOI.** 10.1137/18M1215165

## 1. Introduction.

We consider in this paper the following generalized trust region subproblem (GTRS),

$$\text{(GTRS)} \qquad \begin{aligned} &\min f(x) := x^TAx + 2a^Tx \\ &\text{s.t. } h(x) := x^TBx + 2b^Tx + d \le 0, \end{aligned}$$

where $A$ and $B$ are $n \times n$ symmetric matrices which are not necessarily positive semidefinite, $a, b \in \mathbb{R}^n$, and $d \in \mathbb{R}$.

When the constraint in (GTRS) is a unit ball, the problem reduces to the classical trust region subproblem (TRS). The TRS first arose in trust region methods for nonlinear optimization [6] and also finds applications in the least square problems [31] and robust optimization [2]. Various approaches have been derived to solve the TRS and its variant with additional linear constraints; see [17, 19, 29, 22, 25, 30, 4, 5, 28]. Recently, Hazan and Koren [9] proposed the first linear-time algorithm (with respect to the nonzero entries in the input) for the TRS, via a linear-time eigenvalue oracle and a linear-time semidefinite programming (SDP) solver based on approximate eigenvector computations [16]. After that, Wang and Xia [26] and Ho-Nguyen and Kilinc-Karzan [11] also presented linear-time algorithms to solve the TRS by applying Nesterov's accelerated gradient descent algorithm to a convex reformation of the TRS, which can also be obtained in linear time.

As a generalization of the TRS, the GTRS has received a lot of attention in the literature. The GTRS also admits its own applications such as time of arrival problems

*Received by the editors September 21, 2018; accepted for publication (in revised form) January 30, 2020; published electronically March 12, 2020.
https://doi.org/10.1137/18M1215165
**Funding:** This research was partially supported by Shanghai Sailing Program 18YF1401700, Natural Science Foundation of China (NSFC) 11801087, Science and Technology Commission of Shanghai Municipality Project 19511120700, and Hong Kong Research Grants Council under grants 14213716 and 14202017.
†School of Data Science, Fudan University, Shanghai, China (rjjiang@fudan.edu.cn).
‡School of Data Science, City University of Hong Kong, Hong Kong (dli226@cityu.edu.hk).

[10] and subproblems of consensus ADMM in solving quadratically constrained quadratic programming in signal processing [12]. Numerous methods have been developed for solving the GTRS under various assumptions; see, for example, [18, 24, 3, 25, 7]. Recently, Ben-Tal and den Hertog [2] showed that if the two matrices in the quadratic forms are simultaneously diagonalizable (SD) (see [13] for more details about SD conditions), the GTRS can then be transformed into an equivalent second order cone programming (SOCP) formulation and thus can be solved efficiently. Salahi and Taati [23] also derived an efficient algorithm for solving (GTRS) under the SD condition. Jiang, Li, and Wu [15] 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. Pong and Wolkowicz [21] proposed an efficient algorithm based on extreme generalized eigenvalues of a parameterized matrix pencil for the GTRS, extending the ERW algorithm for the TRS [22]. Recently, Adachi and Nakatsukasa [1] also developed a novel eigenvalue-based algorithm to solve the GTRS. Jiang and Li [14] proposed a novel convex reformulation for the GTRS and derived an efficient first order method to solve the reformulation. However, there is no linear-time algorithm for the GTRS (see section 4.1 for more details on the literature), while Hazan and Koren [9] have already proposed their linear-time algorithm for the TRS. Although it is more general than the TRS, the GTRS still enjoys hidden convexity the same as the TRS due to the celebrated S-lemma [27, 20]. There is also evidence that the closely related generalized eigenvalue problem for a positive definite matrix pencil can be solved in linear time [8]. Then a natural question is whether or not there exists a linear-time algorithm for the GTRS. We offer a positive answer to this question in this paper.

In this paper, we derive a linear-time algorithm, Algorithm 4.1, to approximately solve the GTRS with high probability. The main difficulties in deriving a linear-time algorithm for the GTRS come from the nonconvex constraint of the GTRS; however, this challenging point is not present in the TRS as the constraint in the TRS is convex. More specifically, the nonconvexity of the constraint implies the unboundedness of the feasible region and makes it hard to derive nontrivial initial lower and upper bounds for the GTRS in linear time. These difficulties make the direct generalization of the linear-time algorithm for the TRS in [9] inapplicable to the GTRS. By addressing these difficulties, we are able to propose a linear-time algorithm for the GTRS based on the work in [9]. Moreover, our algorithm also inherits the good property of the algorithm in [9] that avoids the so-called hard case[1] by using approximate eigenvector methods.

The basic idea in our method is to check the feasibility of the following system and then find an $\epsilon$ optimal solution with a binary search over $c$:

$$\begin{aligned} x^TAx + 2a^Tx \quad & \leq\ c, \\ x^TBx + 2b^Tx + d\ & \leq\ 0, \end{aligned} \tag{1.1}$$

where $c \in [l, u]$ and $l$ and $u$ are some lower and upper bounds for (GTRS), respectively. In the TRS, initial $l$ and $u$ can be trivially estimated in linear time as the objective function is continuous on the compact feasible region [9]. However, a linear-time estimation of the lower and upper bounds is nontrivial in the GTRS. We propose

[1]If the null space of the Hessian matrix of the Lagrangian function, $A + \lambda^* B$, with $\lambda^*$ being the optimal Lagrangian multiplier of problem (P), is orthogonal to $a + \lambda^* b$, we are in the hard case, which can happen only when $\lambda^*$ is an extreme eigenvalue of the matrix pencil $(A, B)$; otherwise we are in the easy case.

linear-time subroutines that can find a dual feasible solution that in turn helps identify a lower bound for the primal problem by the weak duality and an upper bound by constructing a feasible solution to the primal problem. The heart of the binary search is that if system (1.1) is feasible, then $c$ is an upper bound for (GTRS); otherwise system (1.1) is infeasible and $c$ is a lower bound. In addition, to apply the linear time SDP solver in [9], we introduce a shift $\epsilon$ to system (1.1), i.e.,

$$\begin{array}{rcl} x^TAx+2a^Tx & \leq & c-\epsilon, \\ x^TBx+2b^Tx+d & \leq & -\epsilon/K. \end{array} \tag{1.2}$$

where $K$ is some parameter that can be estimated in linear time (to be defined in Lemma 2.5). Introducing the parameter $\epsilon$ in the first inequality shifts the value of the objective function with an error $\epsilon$ and introducing the parameter $\epsilon/K$ in the second inequality shifts the value of the objective function at most $\epsilon$ (Lemma 2.5). We then invoke the linear-time SDP solver in [9] that either returns a vector $x$ satisfying system (1.1) or correctly declares that the direct SDP relaxation of (1.2) is infeasible, i.e., a perturbed version of (1.1) is infeasible. Then via a binary search over $c$, we demonstrate that we can obtain an approximate optimal solution. However, there are still issues to address when borrowing the SDP solver in [9]: The SDP solver in [9] requires the feasible set of $X$ to be $\{X : X \succeq 0, \mathrm{tr}(X) \leq 1\}$, where $\mathrm{tr}(X)$ denotes the trace of matrix $X$, and the direct SDP relaxation of (1.2) lies in an unbounded feasible region. We will remedy this by showing that the optimal solution of the GTRS must be in a compact set and further that the optimal solution of the corresponding SDP relaxation should also be in a compact set.

The rest of the paper is organized as follows. In section 2, we propose preliminary results that will be used in our algorithm and analysis and present the statement of our main theorem. We then illustrate the subroutines to support our main algorithm in section 3. We propose our main algorithm and present its analysis in section 4. Finally, we conclude our paper in section 5.

**Notation.** The notation $A \succ 0$ and $A \succeq 0$ represent that the symmetric matrix $A$ is positive definite and positive semidefinite, respectively. We use the notation $x(i)$ to denote the $i$th entry of a vector $x$. We denote by $v^*$ the optimal value of problem (GTRS). Notation $\|A\|_2$ denotes the operator norm of matrix $A$ and notation $\|a\|$ denotes the Euclidean $l_2$ norm of a vector $a$. Notation $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$ denote the largest and smallest eigenvalues of matrix $A$, respectively. Let $N$ be the total number of nonzero entries of matrices $A$ and $B$ in (GTRS) and assume, without loss of generality, $N \geq n$.

## 2. Preliminaries and main results.

In this section, we review some basic properties of the GTRS and some oracles and lemmas that will be used later in our algorithms. Besides, we will also state our main complexity result of our algorithm for solving the GTRS.

### 2.1. Preliminaries.

Besides the linear-time algorithm for the TRS, Hazan and Koren [9] also demonstrated that their algorithm can be extended to solving the GTRS when the quadratic form in the constraint is pure quadratic and positive definite. Now let us consider the more general case where $B$ is indefinite. To avoid some degenerate cases, we do not discuss the case that $B$ is positive semidefinite but singular.

A central tool in this paper is the following linear-time procedures for approximating eigenvectors of sparse matrices, which is based on [16].

LEMMA 2.1 (Lemmas 3 and 5 in [9]). *Given symmetric matrix $C \in \mathbb{R}^{n\times n}$ with $\|C\|_2 \leq \rho$ and nonzero entries $N$ and parameters $\epsilon, \delta > 0$, there exists an approximate* *eigenvector oracle, denoted as* ApproxmaxEV*, that returns a unit vector $x$ with probability of at least $1-\delta$ such that $x^TCx \geq \lambda_{\max}(C) - \epsilon$ and a scalar $\lambda = x^TCx$ in time*

$$O\left(\frac{N\sqrt{\rho}}{\sqrt{\epsilon}}\log\frac{n}{\delta}\right).$$

With the same principle, this oracle can be used to compute an approximate eigenvector associated with the smallest eigenvalue. In fact, the negativity of an approximate largest eigenvalue of matrix $-C$ gives an approximate smallest eigenvalue of matrix $C$.

Lemma 2.2. *Given symmetric matrix $C \in \mathbb{R}^{n\times n}$ with $\|C\|_2 \leq \rho$ and nonzero entries $N$ and parameters $\epsilon, \delta > 0$, there exists an approximate eigenvector oracle, denoted as* ApproxminEV*, that returns a unit vector $x$ with probability of at least $1-\delta$ such that $x^TCx \leq \lambda_{\min}(C) + \epsilon$ and a scalar $\lambda = x^TCx$ in time*

$$O\left(\frac{N\sqrt{\rho}}{\sqrt{\epsilon}}\log\frac{n}{\delta}\right).$$

We further make the following assumption to ensure the boundedness and existence of an optimal solution.

*Assumption* 2.3. Nontrivial upper bounds $\rho_A$ and $\rho_B$ for $\|A\|_2$ and $\|B\|_2$ are given, i.e., $\|A\|_2 \leq \rho_A \leq 3\|A\|_2$ and $\|B\|_2 \leq \rho_B \leq 3\|B\|_2$. Matrix $B$ has at least one negative eigenvalue with $\lambda_{\min}(B) \leq -\xi < 0$ for some $\xi > 0$. For the same $\xi$, there exists a $\mu$ with $\mu \in (0,1]$ such that $\mu A + (1-\mu)B \succeq \xi I$.

Note that an upper bound $\rho_A$ (or $\rho_B$) of $\|A\|_2$ (or $\|B\|_2$) with $\|A\|_2 \leq \rho_A \leq 3\|A\|_2$ (or $\|B\|_2 \leq \rho_B \leq 3\|B\|_2$) can be estimated in linear time with high probability, by Theorem 5 in [9], and hence the first statement of Assumption 2.3 can be made without loss of generality. The second statement of Assumption 2.3 is just a numerically stable consideration for the indefiniteness of matrix $B$. The last statement of Assumption 2.3 is closely related to the so-called regular case, i.e., there exists a $\lambda \geq 0$ such that $A + \lambda B \succ 0$, under which (GTRS) is bounded below and admits a unique optimal solution [18, 21]. Note that Assumption 2.3 also implies the SD condition, since the last statement in Assumption 2.3 implies the regular case, which further implies the SD condition [13]. Hence our assumption that requires some positive lower bound for the smallest eigenvalue of the positive definite matrix $A + \lambda B$ is reasonable for numerically stable consideration. We assume in the following that Assumption 2.3 always holds. For ease of notation, we also define $\tilde{\xi} = \min\{\xi, 1\}$ and

$$\phi = \rho_A + \rho_B + \|a\| + \|b\| + |d| + 1. \tag{2.1}$$

When $B$ is indefinite, we can always find a positive constant $K$ in system (1.2) to ensure that the objective value of (GTRS) shifts at most $\epsilon$ if the constraint shifts $\epsilon/K$ when system (1.2) is infeasible. To demonstrate this, let us first recall the celebrated S-lemma [27, 20].

Lemma 2.4. *For both $i = 1,2$, let $g_i(x) = x^TQ_ix + 2p_i^Tx + q_i$, where $Q_i$ is an $n\times n$ symmetric matrix, $p_i \in \mathbb{R}^n$, and $q_i \in \mathbb{R}$. Assume that there exists an $\bar{x} \in \mathbb{R}^n$ such that $g_2(\bar{x}) < 0$. Then the following two statements are equivalent:*

- ($S_1$) *There is no $x \in \mathbb{R}^n$ such that $g_1(x) \leq 0$ and $g_2(x) < 0$.*
- ($S_2$) *There exists a nonnegative multiplier $\lambda \geq 0$ such that $g_1(x) + \lambda g_2(x) \geq 0$ for all $x \in \mathbb{R}^n$.*

Note that the assumption of the existence of an $\bar{x} \in \mathbb{R}^n$ such that $g_2(\bar{x}) < 0$ automatically holds when $Q_2$ has at least one negative eigenvalue. Using the S-lemma, we have the following results.

LEMMA 2.5. *Let $K > -\lambda_{\max}(A)/\lambda_{\min}(B)$. If the system* (1.2) *is infeasible for some $\epsilon > 0$, then the following system is also infeasible:*

$$\begin{array}{ll} x^T A x + 2a^T x & \le c - 2\epsilon, \\ x^T B x + 2b^T x + d & \le 0. \end{array} \tag{2.2}$$

*Proof.* Note that under Assumption 2.3, $K > -\lambda_{\max}(A)/\lambda_{\min}(B) > 0$. Since $K$ is an upper bound for $\lambda \ge 0$ that satisfies $A + \lambda B \succeq 0$, we have $\lambda/K < 1$. Because $\lambda_{\min}(B) < 0$, there exists an $\bar{x}$ such that $\bar{x}^T B \bar{x} + 2b^T \bar{x} + d + \epsilon/K < 0$. By the S-lemma, the infeasibility of (1.2) implies that there exists a $\lambda \ge 0$ such that

$$\begin{aligned} 0 &\le x^T A x + 2a^T x - c + \epsilon + \lambda\left(x^T B x + 2b^T x + d + \epsilon/K\right) \\ &= x^T A x + 2a^T x - c + 2\epsilon + \lambda\left(x^T B x + 2b^T x + d\right) - \varepsilon \end{aligned} \tag{2.3}$$

for all $x \in \mathbb{R}^n$, where $\varepsilon = (1 - \lambda/K)\epsilon$. Note that (2.3) implies that $A + \lambda B \succeq 0$ and this, together with the definition of $K$, further implies $\varepsilon > 0$. Hence from (2.3), we have

$$x^T A x + 2a^T x - c + 2\epsilon + \lambda\left(x^T B x + 2b^T x + d\right) \ge 0 \ \forall x \in \mathbb{R}^n.$$

Moreover, because $\lambda_{\min}(B) < 0$, there exists an $\bar{x}$ such that $\bar{x}^T B \bar{x} + 2b^T \bar{x} + d < 0$. From the S-lemma we conclude that the system (2.2) is infeasible. ∎

The above lemma shows that if (1.2) is infeasible, then $c - 2\epsilon$ is a lower bound for (GTRS). Since an upper bound of $\lambda_{\max}(A)$ is assumed in Assumption 2.3, i.e., $\lambda_{\max}(A) \le \rho_A$, and $\lambda_{\min}(B)$ can be approximately estimated in linear time, parameter $K$ can be estimated in linear time. From Assumption 2.3, we have $\mu\lambda_{\max}(A) + (1 - \mu)\lambda_{\min} B > 0$ due to $\mu A + (1-\mu)B \succ 0$. This, together with $\|A\|_2 \le \rho_A$ and $\lambda_{\min}(B) \le -\xi$, gives rise to $0 < -\lambda_{\max}(A)/\lambda_{\min}(B) \le \rho_A/\xi$ and thus the estimated $K$ can be restricted to be upper bounded by some positive constant (e.g., $\rho_A/\xi + 1$). Hence we set

$$K = \rho_A/\xi + 1 \tag{2.4}$$

in our algorithm for simplicity.

**2.2. Main results.** Next we state the main theorem (Theorem 2.6) of our paper, showing that an $\epsilon$ optimal solution of (GTRS) can be achieved in linear time by Algorithm 4.1 developed later in this paper.

THEOREM 2.6. *Let $\epsilon > 0$ and $0 < \delta < 1$. Under Assumption* 2.3*, with probability of at least $1 - \delta$, Algorithm* 4.1 *returns an $\epsilon$ optimal solution $\tilde{x}$ to* (GTRS)*, i.e., a feasible solution $\tilde{x}$ with $f(\tilde{x}) \le v^* + \epsilon$, where $v^*$ is the optimal value of* (GTRS)*. The total runtime is*

$$O\left(\frac{N\phi^3}{\sqrt{\epsilon\tilde{\xi}^5}} \log\left(\frac{n}{\delta}\log\frac{\phi}{\epsilon\tilde{\xi}}\right)\log\frac{\phi}{\epsilon\tilde{\xi}}\log\frac{\phi}{\epsilon\tilde{\xi}}\right).$$

The proof of the above theorem will be given in section 4.

## 3. Subroutines.

In this section, we present several subroutines to support our main algorithm, Algorithm 4.1.

**Algorithm 3.1.** Compute parameter $\mu$ such that $\mu A + (1-\mu)B \succeq \frac{\xi}{2} I$.

**Input:** symmetric $A, B \in \mathbb{R}^{n\times n}$ with $\|A\|_2 \le \rho_A$ and $\|B\|_2 \le \rho_B$, $\phi$ in (2.1), $\xi > 0$, and $0 < \delta < 1$

**Output:** $\mu \in (0,1]$ such that $\mu A + (1-\mu)B \succeq \frac{\xi}{2} I$ and a unit vector $x$ and $\lambda = x^T(\mu A+(1-\mu)B)x$ such that $\lambda_{\min}(\mu A+(1-\mu)B) \le \lambda \le \lambda_{\min}(\mu A+(1-\mu)B)+\xi/4$; output is correct with probability of at least $1-\delta$

**function** PsdPencil$(A, B, \xi, \phi, \delta)$
  initialize $T = \log_2 \frac{8\phi}{\xi}$, $\mu_1 = 0$, $\mu_2 = 1$
  **for** $i = 1 : T$ **do**
    $\mu = (\mu_1 + \mu_2)/2$
    invoke $(\lambda, x) \leftarrow$ ApproxminEv$(\mu A + (1-\mu)B, \xi/4, \delta/T)$
    **if** $\lambda \ge 3\xi/4$ **then**
      **return** $(\mu, \lambda, x)$
    **else if** $x^T A x > x^T B x$ **then**
      update $\mu_1 \leftarrow \mu$ ▷ update $x_1 \leftarrow x$ for the proof of Theorem 3.1
    **else**
      update $\mu_2 \leftarrow \mu$ ▷ update $x_2 \leftarrow x$ for the proof of Theorem 3.1
    **end if**
  **end for**
  **return** "Assumption 2.3 fails."
**end function**

**3.1. Algorithm to compute parameter $\mu$ such that $\mu A+(1-\mu)B \succeq \frac{\xi}{2}I$.** In this subsection, a bisection algorithm, Algorithm 3.1, is proposed to find a $\mu$ with $\mu \in (0,1]$ such that $\mu A+(1-\mu)B \succeq \frac{\xi}{2} I$ under Assumption 2.3, which is of linear time. Such a $\mu$ helps us find a lower bound for problem (GTRS) and a compact set in which the optimal solution is located. We first identify an interval $(\mu_1, \mu_2]$ where the targeted $\mu$ is located, if it exists. In Algorithm 3.1, we initialize $(\mu_1, \mu_2]$ as $(0, 1]$. In each step, we invoke the eigenvalue oracle $(\lambda, x) =$ ApproxminEV$(\mu A + (1-\mu)B, \xi/4, \delta/T)$ to find an approximate smallest eigenvalue of the midpoint of this interval, where $x$ is a unit vector such that $\lambda = x^T A x$ and $T$ is a prescribed maximum iteration number. When $\lambda < 3\xi/4$, we cut off half of the interval by eliminating either $(\mu_1, \mu]$ if $x^T A x > x^T B x$ or $(\mu, \mu_2]$ if $x^T A x \le x^T B x$. The intuition of this step is that for all $\nu \in (\mu_1, \mu]$, we have $\lambda_{\min}(\nu A + (1-\nu)B) \le x^T(\nu A + (1-\nu)B)x \le x^T(\mu A + (1-\mu)B)x = \lambda$ when $x^T A x > x^T B x$. This means that the target $\mu$, if it exists, must be in the other half of the interval, i.e., $(\mu, \mu_2]$. The other situation of this step follows the same argument. We prove in the following theorem that under Assumption 2.3, such a $\mu$ can be found correctly in linear time with high probability.

Theorem 3.1. *Let* $0 < \delta < 1$. *Suppose that Assumption* 2.3 *holds. Then* PsdPencil$(A, B, \xi, \phi, \delta)$ *takes at most* $\log_2(\frac{8\phi}{\xi})$ *iterations of* ApproxminEV *and returns* $(\mu, \lambda, x)$ *such that* $\mu \in (0, 1]$, $\lambda = x^T(\mu A + (1-\mu)B)x \ge 3\xi/4$ *and* $\lambda_{\min}(\mu A + (1-\mu)B) \ge \lambda - \xi/4$. *The output is correct with probability of at least* $1-\delta$ *and the total runtime is*

$$O\left(\frac{N\sqrt{\phi}}{\sqrt{\xi}} \log\left(\frac{n}{\delta} \log \frac{\phi}{\xi}\right) \log \frac{\phi}{\xi}\right).$$

*Proof. Runtime.* First note that the algorithm invokes ApproxminEV for at most $T = \log_2 \frac{8\phi}{\xi}$ iterations. In each iteration, ApproxminEV is invoked once for matrix $\mu A + (1-\mu)B$, and the runtime of other main operations (i.e., the matrix vector products $x^T Ax$ and $x^T Bx$) is dominated by ApproxminEV. Since $\|\mu A + (1-\mu)B\|_2 \le \mu \|A\|_2 + (1-\mu)\|B\|_2 \le \phi$, the runtime for each call of ApproxminEV is $O(\frac{N\sqrt{\phi}}{\sqrt{\xi}} \log \frac{nT}{\delta})$ from Lemma 2.2. Hence due to $T = \log_2 \frac{8\phi}{\xi}$, each iteration runs in time $O(\frac{N\sqrt{\phi}}{\sqrt{\xi}} \log(\frac{n}{\delta} \log \frac{\phi}{\xi}))$. Thus, the total runtime (with at most $T$ iterations) of PsdPencil$(A, B, \xi, \phi, \delta)$ is

$$O\left(\frac{N\sqrt{\phi}}{\sqrt{\xi}} \log\left(\frac{n}{\delta}\log\frac{\phi}{\xi}\right)\log\frac{\phi}{\xi}\right).$$

*Correctness.* If the algorithm returns $(\mu, \lambda, x)$ at some iteration $i \le T$, then the returned $\mu$ is the one as required, i.e., $\lambda_{\min}(\mu A + (1-\mu)B) \ge \lambda - \xi/4$. Now it suffices to prove that under Assumption 2.3, the algorithm must terminate at some iteration $i \le T$.

Recall that Assumption 2.3 states that $\exists \mu_0 \in (0,1]$ such that $\mu_0 A + (1-\mu_0)B \succeq \xi I$. Since $\|A\|_2 \le \rho_A$ and $\|B\|_2 \le \rho_B$, we have $A + B \preceq (\rho_A + \rho_B)I$. Hence for any $\varrho \in [-\frac{\xi}{4\phi}, \frac{\xi}{4\phi}]$, we have $(\mu_0 + \varrho)A + (1 - \mu_0 - \varrho)B \succeq \xi I - |\varrho|(\rho_A + \rho_B)I \succeq \frac{3\xi}{4} I$ by noting that $\phi \ge \rho_A + \rho_B$ from (2.1). So for all $\mu' \in [\mu_0 - \frac{\xi}{4\phi}, \mu_0 + \frac{\xi}{4\phi}] \cap (0,1]$, we have $\mu' A + (1-\mu')B \succeq \frac{3\xi}{4} I$. And the length of the interval $[\mu_0 - \frac{\xi}{4\phi}, \mu_0 + \frac{\xi}{4\phi}] \cap (0,1]$ is between $\frac{\xi}{4\phi}$ and $\frac{\xi}{2\phi}$. From the above analysis we know that under Assumption 2.3, the interval length of $\{\mu : \mu A + (1-\mu)B \succeq \frac{3\xi}{4} I\}$ is at least $\frac{\xi}{4\phi}$.

If the algorithm does not terminate in the "for loop," at the end of the loop, we have $x_1^T C(\mu_1) x_1 < 3\xi/4$ and $x_2^T C(\mu_2) x_2 < 3\xi/4$ (note that $x_1$ and $x_2$ are defined in the comments in lines 9 and 11 of Algorithm 3.1, respectively), where $C(\mu_i) = \mu_i A + (1-\mu_i)B$ $(i = 1, 2)$. Then $x_1^T C(\mu) x_1 = \mu x_1^T A x_1 + (1-\mu) x_1^T B x_1 \le \mu_1 x_1^T A x_1 + (1 - \mu_1) x_1^T B x_1 < 3\xi/4$ for $\mu \le \mu_1$ because $x_1^T A x_1 > x_1^T B x_1$. Furthermore, $\lambda_{\min}(C(\mu)) \le x_1^T C(\mu) x_1 < 3\xi/4$ for $\mu \le \mu_1$. Similarly, we have $\lambda_{\min}(C(\mu)) < 3\xi/4$ for $\mu \ge \mu_2$. So if the algorithm terminates at line 14, we have $\mu_2 - \mu_1 \le \frac{\xi}{8\phi}$ as the binary search runs for $\log_2 \frac{8\phi}{\xi}$ iterations and the initial length of $\mu_2 - \mu_1 = 1$. Since for any $\mu$ outside the interval $[\mu_1, \mu_2]$, we have $\lambda_{\min}(\mu A + (1-\mu)B) < 3\xi/4$. The length of interval $\{\mu : \lambda_{\min}(\mu A + (1-\mu)B) \ge 3\xi/4\}$ after the $T$th iteration is then at most $\mu_2 - \mu_1$, which is less than or equal to $\xi/8\phi$. This contradicts Assumption 2.3.

The output is correct with probability of at least $1 - \delta$ (from the union bound) because each output of ApproxminEv is correct with probability of at least $1 - \delta/T$ and the total iteration number is at most $T$. ∎

**3.2. Lower and upper bounds.** In this subsection, we will show that Algorithm 3.2 supplies an estimation of initial lower and upper bounds for the optimal value in linear time. The main principle here is that an upper bound can be found by a feasible solution of (GTRS) and a lower bound can be found by a feasible solution of the Lagrangian dual problem of (GTRS).

An upper bound for problem (GTRS) is given by $u = f(\alpha y)$, where $\alpha$ is a feasible point of the following quadratic inequality and $y$ is the unit eigenvector returned by $(\varpi, y) \leftarrow$ApproxminEV$(B, \xi/2, \delta/2)$:

$$(\alpha y)^T B(\alpha y) + 2b^T(\alpha y) + d \le 0.$$

Under Assumption 2.3, we have $\varpi = y^T B y \le \lambda_{\min}(B) + \xi/2 \le -\xi/2$. Consider the quadratic equation

$$\varpi \alpha^2 + 2b^T y \alpha + d = 0. \tag{3.1}$$

The roots of the above equation are $\alpha = \frac{-2b^T y \pm \sqrt{4(b^T y)^2 - 4\varpi d}}{2\varpi}$ with $|\alpha| \le \frac{4\|b\| + \sqrt{4|\varpi d|}}{2|\varpi|} \le \frac{4\|b\|}{\xi} + \frac{\sqrt{2|d|}}{\sqrt{\xi}}$. This yields a feasible solution $\alpha y$ for (GTRS) (either choice of $\alpha$ is fine). Then we have $|f(\alpha y)| = |\alpha^2 y^T Ay + 2\alpha a^T y| \le \alpha^2 \|A\|_2 + 2|\alpha|\|a\| \le (\frac{4\|b\|}{\xi} + \frac{\sqrt{2|d|}}{\sqrt{\xi}})^2 \rho_A + 2(\frac{4\|b\|}{\xi} + \frac{\sqrt{2|d|}}{\sqrt{\xi}})\|a\| = O(\frac{\phi^3}{\tilde{\xi}^2})$, recalling that $\tilde{\xi} = \min\{\xi, 1\}$. This gives an upper bound

$$u = O\left(\frac{\phi^3}{\tilde{\xi}^2}\right). \tag{3.2}$$

The vector product $b^T y$ can be done in $O(n)$ and finding a feasible solution of the quadratic equation can be done in $O(1)$. The runtime for computing the approximate smallest eigenvalue is $O(\frac{N\sqrt{\phi}}{\sqrt{\xi}} \log \frac{n}{\delta})$. Hence the runtime for computing the upper bound is

$$O\left(\frac{N\sqrt{\phi}}{\sqrt{\xi}} \log \frac{n}{\delta}\right). \tag{3.3}$$

The output is correct with probability of at least $1 - \delta/2$ as we invoke ApproxminEV$(B, \xi/2, \delta/2)$ once.

Next we illustrate that a lower bound can be found by a feasible solution of the Lagrangian dual problem of problem (GTRS). Note that

$$(\mu_0, \lambda, x) \leftarrow \textsc{PsdPencil}(A, B, \xi, \phi, \delta/2)$$

from line 2 in Algorithm 3.2. From Theorem 3.1 and Algorithm 3.2, $\lambda$ is a $\xi/4$ approximate smallest eigenvalue satisfying $\lambda \ge 3\xi/4$ under Assumption 2.3 and the output is correct with probability of at least $1 - \delta/2$. Since the smallest eigenvalue of $\mu_0 A + (1 - \mu_0)B$ satisfies $\lambda_{\min}(\mu_0 A + (1 - \mu_0)B) \ge \lambda - \xi/4 \ge \xi/2$, we have $\mu_0 \rho_A + (1 - \mu_0)(-\xi) \ge \xi/2$ and thus $\frac{3\xi}{2(\rho_A + \xi)} \le \mu_0 \le 1$. The Lagrangian dual problem of (GTRS) is

---

**Algorithm 3.2.** Compute initial lower and upper bounds for (GTRS).

**Input:** symmetric $A, B \in \mathbb{R}^n$ with $\|A\|_2 \le \rho_A$ and $\|B\|_2 \le \rho_B$, $a, b \in \mathbb{R}^n$, $d \in \mathbb{R}$, $\delta > 0$, $\phi$ in (2.1) and $\xi > 0$

**Output:** $\mu_0 \in (0, 1]$ such that $\mu_0 A + (1 - \mu_0)B \succeq \frac{\xi}{2} I$, a unit vector $x$ and $\lambda = x^T(A + \mu_0 B)x$ such that $\lambda_{\min}(\mu_0 A + \mu_0 B) \le \lambda \le \lambda_{\min}(\mu_0 A + \mu_0 B) + \xi/4$ and lower and upper bounds $l$ and $u$ for (GTRS); output is correct with probability of at least $1 - \delta$

1: **function** Bounds$(A, B, \xi, a, b, d, \phi, \delta)$
2:     invoke $(\mu_0, \lambda, x) \leftarrow \textsc{PsdPencil}(A, B, \xi, \phi, \delta/2)$
3:     define $\bar{\lambda} = \lambda - \xi/4$, $\nu_0 = (1 - \mu_0)/\mu_0$ and $p = a + \nu_0 b$
4:     set $l = \frac{d\mu_0}{1 - \mu_0} - \frac{\mu_0 \|p\|^2}{\bar{\lambda}}$     ▷ initial lower bound
5:     invoke $(\varpi, y) \leftarrow$ ApproxminEV$(B, \xi/2, \delta/2)$
6:     find a root $\alpha$ to (3.1)
7:     set $u = f(\alpha y)$     ▷ initial upper bound
8:     **return** $(\mu_0, \lambda, l, u)$
9: **end function**

---

$$
\begin{aligned}
\text{(L)} \qquad & \max_{\nu\geq 0}\min_{x} f(x)+\nu h(x) \\
&= \max_{\nu\geq 0}(x(\nu))^T(A+\nu B)x(\nu)+2(a+\nu b)^T x(\nu)+d\nu,
\end{aligned}
$$

where $x(\nu)$ is the optimal solution of the inner minimization problem. Letting $\nu_0 = (1-\mu_0)/\mu_0$, we have $P = A+\nu_0 B \succeq \frac{\xi}{2\mu_0}I$. So $\nu_0$ is a feasible solution for problem (L). Define $L(\nu) := \min_x f(x)+\nu h(x)$. Thus from the weak duality, we have the following inequality:

$$L(\nu_0) \leq v^*.$$

Letting $p = a+\nu_0 b$, a lower bound of $L(\nu_0)$ can be found by the following formulation:

$$
\begin{aligned}
L(\nu_0) &= \min_x \left(x+P^{-1}p\right)^T P\left(x+P^{-1}p\right)+d\nu_0 - p^T P^{-1} p \\
&= d\nu_0 - p^T P^{-1}p \\
&\geq d\nu_0 - \frac{\|p\|^2}{\lambda_{\min}(P)} \\
&\geq d\nu_0 - \frac{2\mu_0\|p\|^2}{\xi}.
\end{aligned}
$$

Note that $\frac{3\xi}{2(\rho_A+\xi)} \leq \mu_0 \leq 1$ implies $0\leq \nu_0 \leq 2\rho_A/3\xi$ and thus $\|p\| \leq \|a\|+2\rho_A\|b\|/3\xi$. Hence $l = d\nu_0 - \frac{2\mu_0\|p\|^2}{\xi}$ gives a lower bound of (GTRS). And we have

$$l \leq |l| \leq |d\nu_0| + \frac{2\mu_0\|p\|^2}{\xi} \leq 2\rho_A|d|/3\xi + \frac{2\mu_0(\|a\|+\|b\|(1/\mu_0-1))^2}{\xi},$$

which, due to $\rho_A|d|\leq \phi^2$ and $\mu_0(\|a\|+\|b\|(1/\mu_0-1))^2 = \mu_0\|a\|^2+2\|a\|\|b\|(1-\mu_0)+\|b\|^2(\mu_0-2+1/\mu_0) \leq \phi^2+2\phi^2+\frac{2\phi^3+2\phi^2\xi}{3\xi}$, further implies that

$$l \leq |l| \leq O(\phi^3/\xi^2). \tag{3.4}$$

This, together with (3.2), further implies

$$u - l \leq O\left(\frac{\phi^3}{\tilde{\xi}^2}\right).$$

Note also that the main time cost for computing the lower bound is in calling PsdPencil, which runs in time, by Theorem 3.1,

$$O\left(\frac{N\sqrt{\phi}}{\sqrt{\xi}}\log\left(\frac{n}{\delta}\log\frac{\phi}{\xi}\right)\log\frac{\phi}{\xi}\right). \tag{3.5}$$

As the runtime for computing the lower bound (3.5) dominates the runtime for computing the upper bound (3.3), the total runtime of Bounds$(A,B,\xi,a,b,\phi,\delta)$ is

(3.6)
$$O\left(\frac{N\sqrt{\phi}}{\sqrt{\xi}}\log\frac{n}{\delta}+\frac{N\sqrt{\phi}}{\sqrt{\xi}}\log\left(\frac{n}{\delta}\log\frac{\phi}{\xi}\right)\log\frac{\phi}{\xi}\right) = O\left(\frac{N\sqrt{\phi}}{\sqrt{\xi}}\log\left(\frac{n}{\delta}\log\frac{\phi}{\xi}\right)\log\frac{\phi}{\xi}\right).$$

As we call PsdPencil$(A,B,\xi,\phi,\delta/2)$ and ApproxminEV$(B,\xi/2,\delta/2)$ once each, the output of Bounds$(A,B,\xi,a,b,\phi,\delta)$ is correct with probability of at least $1-\delta$ from the union bound.

In summary, we have the following theorem.

THEOREM 3.2. *Let $\epsilon > 0$ and $0 < \delta < 1$. Under Assumption* 2.3, *with probability of at least $1-\delta$, the algorithm* BOUNDS$(A, B, \xi, a, b, d, \phi, \delta)$ *computes lower and upper bounds, $l$ and $u$, for* (GTRS) *satisfying*

$$u - l \leq O\left(\frac{\phi^3}{\tilde{\xi}^2}\right),$$

*and the total runtime is*

$$O\left(\frac{N\sqrt{\phi}}{\sqrt{\xi}} \log\left(\frac{n}{\delta}\log\frac{\phi}{\xi}\right)\log\frac{\phi}{\xi}\right).$$

**3.3. Identify feasibility of quadratic systems.** This subsection shows that line 6 of Algorithm 4.1 returns an upper bound $R$ of the Euclidean norm of the optimal solution, where

$$R = \frac{\mu_0\|a+\nu_0 b\|}{\bar{\lambda}} + \sqrt{\left(\left\|\frac{\mu_0}{\bar{\lambda}}(a+\nu_0 b)\right\|^2 + \frac{\mu_0}{\bar{\lambda}}(u-\nu_0 d)\right)}, \tag{3.7}$$

and using this bound, the subroutine FEAS can help us solve the GTRS via bisection.

The subroutine FEAS utilizes a linear-time SDP solver, RELAXSOLVE, developed in [9], to approximately solve the following feasibility problem:

$$D_i \bullet X \geq \epsilon, \quad i = 1, 2, \qquad X \in \mathcal{K}, \tag{3.8}$$

where $D_i$, $i = 1, 2$, are symmetric matrices with $\|D_i\|_2 \leq 1$ and $\mathcal{K} = \{X : X \succeq 0,\ \mathrm{tr(X)} \leq 1\}$.

LEMMA 3.3 (Theorem 2 in [9]). *Given symmetric matrices $D_1,\ D_2 \in \mathbb{R}^{n\times n}$ with $\|D_i\|_2 \leq 1$ and $\epsilon, \delta > 0$, with probability of at least $1-\delta$,* RELAXSOLVE *outputs a matrix $X \in \mathcal{K}$ of rank 2 that satisfies $D_i \bullet X \geq \epsilon/2, i = 1, 2,$ or correctly declares that* (3.8) *is infeasible. The algorithm calls the oracle* APPROXMAXEV *at most $O(\log\frac{1}{\epsilon})$ times and can be implemented to run in total time*

$$O\left(\frac{N}{\sqrt{\epsilon}}\log\frac{1}{\epsilon}\log\left(\frac{n}{\delta}\log\frac{1}{\epsilon}\right)\right).$$

The direct SDP relaxation of (1.2) is

$$P_i \bullet X \geq \epsilon,\ i = 1, 2,\ X_{11} = 1, \quad X \succeq 0, \tag{3.9}$$

where $P_1 = \left(\begin{smallmatrix} c & -a^T \\ -a & -A\end{smallmatrix}\right)$, $P_2 = K\left(\begin{smallmatrix} -d & -b^T \\ -b & -B\end{smallmatrix}\right)$. However, the feasible region for problem (GTRS) may not be contained in $\mathcal{K}$ and thus we cannot directly utilize the SDP solver in [9]. To address this issue, we prove in the next theorem that the optimal solution of (GTRS) must be in a compact set.

THEOREM 3.4. *Let $\mu_0$ be returned by line* 2 *of* BOUNDS$(A, B, \xi, a, b, \phi, \delta/2)$ *(Algorithm* 3.2*), $R$ be defined by* (3.7)*, i.e., line* 6 *of Algorithm* 4.1*, and $x^*$ be an optimal solution of problem* (GTRS)*. Then, under Assumption* 2.3*, it holds that $\|x^*\| \leq R$ with probability of at least $1-\delta/2$. Moreover, $R \leq O(\phi^{3/2}/\tilde{\xi}^{3/2})$.*

*Proof.* Let the notation be the same as in section 3.2. Hence the output is correct with probability of at least $1-\delta/2$ from Theorem 3.2. Also we have $\lambda_{\min}(\mu_0 A + (1-\mu_0)B) \geq \bar{\lambda} = \lambda - \xi/4 \geq \xi/2 > 0$. This gives rise to $\lambda_{\min}(A+\nu_0 B) \geq \bar{\lambda}/\mu_0$, where

$\nu_0 = 1/\mu_0 - 1$. From the optimality of $x^*$, we have $f_1(x^*) - u + \nu_0 f_2(x^*) \le 0$, where $u$ is an upper bound of (GTRS) derived from line 3 or 11 of Algorithm 4.1, and thus

$$(x^*)^T(A + \nu_0 B)x^* + 2(a + \nu_0 b)^T x^* - u + \nu_0 d \le 0.$$

Let $\mathcal{X} = \{x : x^T(A + \nu_0 B)x + 2(a + \nu_0 b)^T x - u + \nu_0 d \le 0\}$. Then

$$\begin{aligned}\mathcal{X} &\subset \left\{x : \frac{\bar\lambda}{\mu_0} x^T x + 2(a + \nu_0 b)^T x - u + \nu_0 d \le 0\right\}\\ &= \left\{x : \left\|x - \frac{\mu_0(a + \nu_0 b)}{\bar\lambda}\right\|^2 \le \frac{\mu_0(u - \nu_0 d)}{\bar\lambda} + \frac{\mu_0^2\|a + \nu_0 b\|^2}{\bar\lambda^2}\right\}.\end{aligned}$$

This further implies $\mathcal{X} \subset \{x : \|x\| \le R\}$, where $R$ is defined in (3.7). From the arguments in section 3.2, we have $3\xi/2(\rho_A + \xi) \le \mu_0 \le 1$. Hence, together with $\phi = \rho_A + \rho_B + \|a\| + \|b\| + |d| + 1$, $\bar\lambda \ge \xi/2$, $\mu_0\nu_0 = 1 - \mu_0 \le 1$, and $u \le O(\phi^3/\tilde\xi^2)$, we have

$$\frac{\mu_0\|a + \nu_0 b\|}{\bar\lambda} \le \frac{\|\mu_0 a\| + \|\mu_0\nu_0 b\|}{\bar\lambda} \le \frac{\|a\| + \|b\|}{\bar\lambda} \le \frac{2\phi}{\xi} = O\left(\frac{\phi}{\xi}\right)$$

and

$$\frac{\mu_0}{\bar\lambda}(u - \nu_0 d) \le O\left(\frac{\phi^3}{\tilde\xi^2\xi}\right) + \frac{2\phi}{\xi}.$$

Recalling $\tilde\xi = \min\{\xi, 1\}$ and $\phi \ge 1$, we have $R \le O(\phi^{3/2}/\tilde\xi^{3/2})$. □

Let $x^*$ be an optimal solution of problem (GTRS). The above theorem shows us that $\|x^*\|^2 \le R^2$. By defining $Y^* = \binom{1}{x^*}\binom{1}{x^*}^T/S$, we have

$$\mathrm{tr}(Y^*) = \mathrm{tr}\left(\begin{pmatrix}1\\ x^*\end{pmatrix}^T \begin{pmatrix}1\\ x^*\end{pmatrix} \Big/ S\right) = \frac{\|x^*\|^2 + 1}{S} \le 1,$$

i.e., $Y^* \in \mathcal{K}$, where $S = R^2 + 1$. This motivates us to solve the following SDP system instead of (3.9):

$$\frac{1}{\kappa} P_i \bullet Y \ge \frac{\epsilon}{\kappa S}, \quad i = 1, 2, \quad Y \in \mathcal{K}, \tag{3.10}$$

where $\kappa = \max\{\mu_A, K\mu_B\}$, $\mu_A = \rho_A + 2\|a\| + |c|$, $\mu_B = \rho_B + 2\|b\| + |d|$. These parameters make $\|P_i/\kappa\|_2 \le 1, i = 1, 2$, and the optimal solution, if it exists, $Y^* \in \mathcal{K}$. Therefore, the SDP feasibility problem (3.10) can be solved by the linear-time SDP solver, RelaxSolve. Then due to Lemma 3.3, with probability of at least $1 - \delta$, RelaxSolve$(\frac{P_1}{\kappa}, \frac{P_2}{\kappa}, \frac{\epsilon}{\kappa S}, \delta)$ either declares that (3.10) is infeasible, which further implies the infeasibility of (3.9), or returns $Y \in \mathcal{K}$ such that $\frac{1}{\kappa} P_i \bullet Y \ge \frac{\epsilon}{2\kappa S}, i = 1, 2$, which is further equivalent to that $X = SY$ satisfies $P_i \bullet X \ge \epsilon/2,\ i = 1, 2,\ X \succeq 0$. When RelaxSolve$(\frac{P_1}{\kappa}, \frac{P_2}{\kappa}, \frac{\epsilon}{\kappa S}, \delta)$ returns $Y \in \mathcal{K}$ such that $\frac{1}{\kappa} P_i \bullet Y \ge \frac{\epsilon}{2\kappa S}$, we further invoke the SZRotation algorithm in [9], which is a variant of the matrix decomposition procedure in Sturm and Zhang [25], to find a vector $z$ such that $z^T P_i z \ge \epsilon/(2r), i = 1, 2$ with $r = \mathrm{rank}(Y) \le 2$ as there are only two inequalities in (3.10).

Lemma 3.5 (see [9]). *Given a decomposition $X = \sum_{i=1}^r x_i x_i^T$ of a positive semidefinite matrix $X$ of rank $r$ and an arbitrary matrix $M$ with $M \bullet X \ge a$, SZRotation$(M, X)$ outputs a decomposition $X = \sum_{i=1}^r y_i y_i^T$ such that $y_i^T M y_i \ge a/r$ for all $i = 1, \ldots, r$. The procedure runs in time $O(Nr)$, where $N \ge n$ is the number of nonzero entries in $M$.*

**Algorithm 3.3.** Find a feasible solution for (1.1) or declare the infeasibility of (2.2).

**Input:** symmetric $A, B \in \mathbb{R}^{n\times n}$ with $\|A\|_2 \le \rho_A$ and $\|B\|_2 \le \rho_B$, $a, b, c, d, \epsilon, \delta$, $\mu_A = \rho_A + 2\|a\| + |c|$, $\mu_B = \rho_B + 2\|b\| + |d|$, $K$ in (2.4), $R > 0$

**Output:** find a feasible solution $x$ for (1.1) or declare the infeasibility of (2.2); output is correct with probability of at least $1 - \delta$

**function** FEAS$(A, B, a, b, c, d, \epsilon, \delta, \mu_A, \mu_B, K, R)$
  define $S = (R+1)^2$
  define $\kappa = \max\{\mu_A, K\mu_B\}$
  define $(n+1)\times(n+1)$ symmetric matrices,

$$Q_1 = \frac{1}{\kappa}\begin{pmatrix} c & -a^T \\ -a & -A \end{pmatrix} \text{ and } Q_2 = \frac{K}{\kappa}\begin{pmatrix} -d & -b^T \\ -b & -B \end{pmatrix}$$

  invoke RelaxSolve$(Q_1, Q_2, \epsilon/(\kappa S), \delta)$
  **if** RelaxSolve returns “infeasible” **then**
    **return** “infeasible”
  **else** $\{$RelaxSolve returns $Y$ such that $Q_i \bullet Y \ge \frac{\epsilon}{2\kappa S}, i = 1, 2\}$
    invoke SZRotation$(Q_1, SY)$ that return $X = \sum_{i=1}^{r} z_i z_i^T$ as output
    **if** $r = 1$ **then**
      set $z = z_1$ and $\tilde{z} = z(2 : n+1)$
    **else** ▷ r=2
      find a vector $z \in \{z_1, z_2 \ldots, z_r\}$ for which $z^T Q_2 z \ge \epsilon/(2r\kappa)$ and let $\tilde{z} = z(2 : n+1)$
    **end if**
    **if** $z(1) \ne 0$ **then**
      $x = \tilde{z}/z(1)$
    **else**
      set $\alpha = \min\{\frac{\epsilon}{2r(|2b^T\tilde{z}|+|d|)}, \frac{\epsilon}{2Kr(|2a^T\tilde{z}|+|c|)}, 1\}$; $x = \tilde{z}/\alpha$
    **end if**
    **return** $x$
  **end if**
**end function**

Theorem 3.6. *Given the linear-time SDP solver* RelaxSolve *and parameters* $\epsilon, \delta > 0$ *and* $K = \rho_A/\xi + 1$ *as in* (2.4)*, Algorithm* 3.3*, with probability of at least* $1 - \delta$*, returns a vector* $x \in \mathbb{R}^n$ *satisfying system* (1.1)*, or correctly declares that* (2.2) *is infeasible. The total runtime of Algorithm* 3.3 *is*

$$O\left(\frac{N\phi^3}{\sqrt{\epsilon\tilde{\xi}^5}}\log\left(\frac{n}{\delta}\log\frac{\phi}{\epsilon\tilde{\xi}}\right)\log\frac{\phi}{\epsilon\tilde{\xi}}\right).$$

*Proof. Runtime.* The main runtime is in the two subalgorithms RelaxSolve and SZRotation with $O(\frac{N}{\sqrt{\epsilon'}}\log(\frac{n}{\delta}\log\frac{1}{\epsilon'})\log\frac{1}{\epsilon'})$ and $O(Nr)$, respectively, where $\epsilon' = \epsilon/(\kappa S)$ and $r \le 2$ is the rank of $X$ returned by RelaxSolve. Since $O(Nr)$ is dominated by $O(\frac{N}{\sqrt{\epsilon'}}\log(\frac{n}{\delta}\log\frac{1}{\epsilon'})\log\frac{1}{\epsilon'})$, the total runtime is

$$O\left(\frac{N\sqrt{\kappa S}}{\sqrt{\epsilon}}\log\left(\frac{n}{\delta}\log\frac{\kappa S}{\epsilon}\right)\log\frac{\kappa S}{\epsilon}\right) = O\left(\frac{N\phi^3}{\sqrt{\epsilon\tilde{\xi}^5}}\log\left(\frac{n}{\delta}\log\frac{\phi}{\epsilon\tilde{\xi}}\right)\log\frac{\phi}{\epsilon\tilde{\xi}}\right),$$

where we use $\kappa = \max\{\mu_A, K\mu_B\} \le O(\phi^3/\tilde{\xi}^2)$ and $S = R^2 + 1 \le O(\phi^3/\tilde{\xi}^3)$. The bound for $\kappa$ follows that $\mu_A = \rho_A + 2\|a\| + |c| \le 2\phi + |u| + |l| \le O(\phi^3/\tilde{\xi}^2)$ (we use $c \le |l| + |u|$ in the first inequality and (3.2) and (3.4) in the second inequality), $K = O(\phi/\xi)$, and $\mu_B = \rho_B + 2\|b\| + |d| = O(\phi)$.

*Correctness.* If RelaxSolve returns "infeasible," the SDP relaxation (3.10) is infeasible and thus (3.9) is infeasible. This implies the infeasibility of (1.2) since (3.9) is a relaxation of (1.2). And the infeasibility of (1.2) further implies the infeasibility of (2.2) by Lemma 2.5.

Now let us assume that RelaxSolve returns $Y \in \mathcal{K}$ such that $Q_i \bullet Y \ge \frac{\epsilon}{2\kappa S}$, $i = 1, 2$. Then $X = SY$ satisfies $P_i \bullet X \ge \epsilon/2$, $i = 1, 2$, $X \succeq 0$. As shown in RelaxSolve in [9], we have $Y = qy_1y_1^T + (1-q)y_2y_2^T$ for $q \in [0, 1]$, which means that $\operatorname{rank}(Y) \le 2$ and thus $\operatorname{rank}(X) \le 2$. It follows from Lemma 3.5 that invoking SZRotation$(Q_1, X)$ yields a solution $X = \sum_{i=1}^r z_iz_i^T$ such that $z_i^TQ_1z_i \ge \epsilon/(2r\kappa)$, $i = 1, 2$. Together with $Q_2 \bullet X \ge \epsilon/(2\kappa)$, we conclude that at least one of $z_i$ satisfies $z_i^TQ_2z_i \ge \epsilon/(2r\kappa)$. So SZRotation$(Q_1, X)$ indeed finds a vector $z \in \{z_1, z_2\}$ such that $z^TQ_jz \ge \epsilon/(2r\kappa)$ $(j = 1, 2)$, which further implies that

$$\begin{aligned} \tilde{z}^TA\tilde{z} + 2z(1)a^T\tilde{z} &\le cz(1)^2 - \epsilon/(2r), \\ \tilde{z}^TB\tilde{z} + 2z(1)b^T\tilde{z} + z(1)^2d &\le -\epsilon/(2rK), \end{aligned}$$

where $\tilde{z} = z(2 : n+1)$.

If $z(1) \ne 0$, by dividing $z(1)^2$ on both sides of the two inequalities in the above system and letting $x = \tilde{z}/z(1)$, we have

$$\begin{aligned} x^TAx + 2a^Tx - c &\le -\epsilon/(2rz(1)^2), \\ x^TBx + 2b^Tx + d &\le -\epsilon/(2rKz(1)^2). \end{aligned}$$

Then we have

$$\begin{aligned} x^TAx + 2a^Tx &\le c, \\ x^TBx + 2b^Tx + d &\le 0 \end{aligned}$$

as required. Else we have $z(1) = 0$. Note that

$$\begin{aligned} &\tilde{z}^TA\tilde{z} \le -\epsilon/(2r), \\ &\tilde{z}^TB\tilde{z} \le -\epsilon/(2rK). \end{aligned}$$

By setting $\alpha = \min\{\frac{\epsilon}{2r(|2a^T\tilde{z}|+|c|)}, \frac{\epsilon}{2rK(|2b^T\tilde{z}|+|d|)}, 1\} \le 1$ and $x = \tilde{z}/\alpha$, we have

$$\begin{aligned} x^TAx &= \frac{\tilde{z}^TA\tilde{z}}{\alpha^2} \\ &\le -\frac{\epsilon}{2r\alpha^2} \\ &\le -\frac{|2a^T\tilde{z}| + |c|}{\alpha} \qquad \left(\text{due to } -1/\alpha \le -\frac{2r(|2a^T\tilde{z}| + |c|)}{\epsilon}\right) \\ &\le -|2a^Tx| - |c| \qquad (\text{due to } x = \tilde{z}/\alpha \text{ and } \alpha \le 1). \end{aligned}$$

Thus, $x^TAx + 2a^Tx - c \le x^TAx + 2|a^Tx| + |c| \le 0$. Similarly, we have

$$x^TBx + 2b^Tx + d \le 0.$$

Hence $x$ is indeed a solution to system (1.1). ∎

**4. Main algorithm.** In our main algorithm, Algorithm 4.1, we first invoke a subroutine Bounds (defined in section 3.2) to compute an initial estimation for lower and upper bounds, $l$ and $u$, for problem (GTRS) and then use bisection techniques, by invoking Feas (defined in section 3.3) for at most $O(\log(\frac{u-l}{\epsilon}))$ iterations, to obtain a feasible solution $\tilde{x}$ with $f(\tilde{x}) \le v^* + \epsilon$, where $v^*$ is the optimal value of (GTRS).

**Algorithm 4.1.** Find an $\epsilon$ optimal solution for (GTRS).

**Input:** symmetric $A, B \in \mathbb{R}^n$ with $\|A\|_2 \le \rho_A$ and $\|B\|_2 \le \rho_B$, $a, b \in \mathbb{R}^n$, $d \in \mathbb{R}$, $\epsilon, \delta > 0$, and $\xi > 0$

**Output:** an $\epsilon$ optimal solution $x$; output is correct with probability of at least $1-\delta$

1: **function** GTRS$(A, B, a, b, d, \epsilon, \rho_A, \rho_B, \xi, \delta)$
2: let $K = \rho_A/\xi + 1$ and $\phi = \rho_A + \rho_B + \|a\| + \|b\| + |d| + 1$
3: invoke $(\mu_0, \lambda, l, u) \leftarrow$Bounds$(A, B, \xi, a, b, \phi, \delta/2)$
4: define $\nu_0 = 1/\mu_0 - 1$, $\bar{\lambda} = \lambda - \xi/4$, $c = \frac{l+u}{2}$, $\epsilon' = \frac{\epsilon}{7}$, $T = \log_2(\frac{u-l}{\epsilon'})$, $\delta' = \frac{\delta}{2T}$, $\mu_A = \rho_A + 2\|a\| + |c|$ and $\mu_B = \rho_B + 2\|b\| + |d|$
5: **for** $t = 1 : T$ **do**
6: set $R$ as (3.7)
7: invoke Feas$(A, B, a, b, c, d, \epsilon', \xi, \delta', \mu_A, \mu_B, K, R)$
8: **if** Feas$(A, B, a, b, c, d, \epsilon', \xi, \delta', \mu_A, \mu_B, K, R)$ returns "infeasible" **then**
9: set $l = c - 2\epsilon'$; $c = (l+u)/2$
10: **else** Feas$(A, B, a, b, c, d, \epsilon', \xi, \delta', \mu_A, \mu_B, K, R)$ returns a feasible solution $x$ to (1.1)
11: set $u = \min\{u, x^TAx + 2a^Tx\}$, $c = (l+u)/2$ and $\mu_A = \rho_A + 2\|a\| + |c|$
12: **end if**
13: **end for**
14: **return** $x$
15: **end function**

We are now ready to present our main result, which shows us the correctness and linear runtime of Algorithm 4.1.

Theorem 4.1 (restatement of Theorem 2.6). *Let $\epsilon > 0$ and $0 < \delta < 1$. Under Assumption* 2.3, *with probability of at least $1-\delta$, Algorithm* 4.1 *returns an $\epsilon$ optimal solution $\tilde{x}$ to* (GTRS)*, i.e., a feasible solution $\tilde{x}$ with $f(\tilde{x}) \le v^* + \epsilon$, where $v^*$ is an optimal value of* (GTRS)*. The total runtime is*

$$O\left(\frac{N\phi^3}{\sqrt{\epsilon\tilde{\xi}^5}} \log\left(\frac{n}{\delta}\log\frac{\phi}{\epsilon\tilde{\xi}}\right)\log\frac{\phi}{\epsilon\tilde{\xi}}\log\frac{\phi}{\epsilon\tilde{\xi}}\right). \tag{4.1}$$

*Proof. Correctness.* Theorem 3.2 shows that the lower and upper bounds $l$ and $u$ can be estimated, with probability of at least $1-\delta/2$, by Bounds$(A, B, \xi, a, b, \phi, \delta/2)$ and $u - l \le O(\phi^3/\tilde{\xi}^2)$.

From Theorem 3.6, we know that the subroutine Feas either returns a feasible solution for (1.1) (yielding a new upper bound $u = \min\{c, x^TAx + 2a^Tx\}$) or declares the infeasibility of (2.2) (yielding a new lower bound $l = c - 2\epsilon'$). Now consider the loop in lines 5–13 of Algorithm 4.1. Let $l_p$ and $u_p$ denote the values of $l$ and $u$ in the end of the $p$th iteration in the "for" loop (particularly, let $l_0$ and $u_0$ be the initial values of $l$ and $u$) and then the length of $u-l$ is at most $\frac{u_p - l_p}{2} + 2\epsilon'$ at the end of the current iteration. At the end of the main loop of Algorithm 4.1 (for simplicity, assume $T \ge 1$), the length of $u_T - l_T$ satisfies

$$u_T - l_T \leq \frac{u_0 - l_0}{2^T} + \left(2 + 1 + \cdots + \frac{1}{2^{T-2}}\right)\epsilon' \leq \epsilon' + 4\epsilon' = 5\epsilon'.$$

From Lemma 2.5, we have $l_T - 2\epsilon' \leq f(x^*) \leq f(x) = u_T$. Thus $u_T - f(x^*) \leq u_T - (l_T - 2\epsilon') \leq (5+2)\epsilon' = 7\epsilon' = \epsilon$. So $f(x) \leq f(x^*) + \epsilon$. That is, after

$$O\left(\log \frac{u-l}{\epsilon}\right) = O\left(\log \frac{\phi}{\tilde{\xi}\epsilon}\right)$$

iterations of binary search, we obtain an $\epsilon$ optimal solution.

*Runtime.* The main runtime of Algorithm 4.1 is in subroutines BOUNDS and FEAS. Equation (3.6) shows that the main operations in BOUNDS$(A, B, \xi, a, b, \phi, \delta/2)$ run in time

$$O\left(\frac{N\sqrt{\phi}}{\sqrt{\xi}} \log\left(\frac{n}{\delta} \log \frac{\phi}{\xi}\right) \log \frac{\phi}{\xi}\right). \tag{4.2}$$

The result returned by BOUNDS$(A, B, \xi, a, b, \phi, \delta/2)$ is correct with probability of at least $1 - \delta/2$.

Note that Algorithm 4.1 invokes FEAS $O(\log \frac{u-l}{\epsilon'})$ times. Then from Theorem 3.6 in section 3.3, the total time of lines 5–13 is

$$O\left(\frac{N\phi^3}{\sqrt{\epsilon'\tilde{\xi}^5}} \log\left(\frac{n}{\delta'} \log \frac{\phi}{\epsilon'\tilde{\xi}}\right) \log \frac{\phi}{\epsilon'\tilde{\xi}} \log \frac{u-l}{\epsilon'}\right), \tag{4.3}$$

which is equivalent to

$$O\left(\frac{N\phi^3}{\sqrt{\epsilon\tilde{\xi}^5}} \log\left(\frac{n}{\delta} \log \frac{\phi}{\epsilon\tilde{\xi}}\right) \log \frac{\phi}{\epsilon\tilde{\xi}} \log \frac{\phi}{\epsilon\tilde{\xi}}\right), \tag{4.4}$$

by noting $u - l \leq O(\phi^3/\tilde{\xi}^2)$, $\delta' = \delta/(2T)$, $T = \log_2((u-l)/\epsilon')$, and $\epsilon' = \epsilon/7$. The output is correct with probability of, by noting that $\delta' = \frac{\delta}{2T}$, at least $1 - T \times \frac{\delta}{2T} = 1 - \frac{\delta}{2}$.

Hence the output of the whole algorithm is correct with probability of at least $1 - \delta$. As the runtime (4.4) dominates (4.2), we conclude that the total runtime is

$$O\left(\frac{N\phi^3}{\sqrt{\epsilon\tilde{\xi}^5}} \log\left(\frac{n}{\delta} \log \frac{\phi}{\epsilon\tilde{\xi}}\right) \log \frac{\phi}{\epsilon\tilde{\xi}} \log \frac{\phi}{\epsilon\tilde{\xi}}\right).$$

$\square$

*Remark* 4.2. We have to point out a weakness of our algorithm. We are currently unable to implement the algorithm efficiently to beat existing algorithms, especially the algorithms in our recent paper [14]. The paper [14] presents an efficient algorithm, whose practical runtime is mainly consumed by an extreme eigenpair computation when the dimension $n$ ranges from 10,000 to 40,000. Provided a known parameter $\lambda \geq 0$ such that $A + \lambda B \succ 0$, the runtime for the steepest descent method for the minimax reformulation of GTRS in [14] is even less than that of an extreme eigenpair computation. The numerical performance of [14] is much better than its theoretical guarantees. Our result in this paper focuses on theoretical guarantees for approximately solving the GTRS. Algorithm 4.1 involves repetitive calls, though the number of calls is theoretically of an order $O(\log(\cdot))$, of approximately computing an eigenpair corresponding to the smallest eigenvalue of a symmetric matrix, which is heavy when compared to one eigenpair computation in [14]. Hence up to now, our algorithm in this paper cannot numerically outperform that in [14] even under some careful implementation. We would like to study how to implement our algorithm (with suitable modifications) efficiently as a future direction.

**4.1. Comparisons.** It is interesting to compare the complexity bounds of our linear-time algorithm to existing algorithms in the literature. The eigenvalue-based algorithm in section 3 of [1] and the method in [14] all require the regularity condition $A+\lambda B \succ 0$ (this case is named "definite feasible QCQP" in [1]); our algorithm in this paper requires a similar condition, a slightly stronger condition $\mu A + (1-\mu)B \succ \xi I$ (Assumptiion 2.3). However, both [1] and [14] do not give a way to compute such a $\lambda$ in linear time. Moreover, given such a $\lambda$, the overall complexity of the approaches in [1] and [14] depends on the complexity of computing an exact extreme generalized eigenpair, which is unknown, and the impact of approximate computation of the generalized eigenvalue pair on the solution accuracy has not been investigated in these papers. Though in practice, [1] and [14] use an inexact eigenpair, their theoretical guarantee is missing. On the other hand, our subroutine PsdPencil gives a linear-time algorithm for computing a $\lambda \geq 0$ such that $A + \lambda B \succ 0$ approximately holds. Besides, the algorithm in [14] requires $O(\frac{L\|x_0-x^*\|^2}{\epsilon})$ iterations to solve a minimax reformulation of the GTRS to achieve an $\epsilon$ optimal solution, where $L$ is the Lipschitz constant for the gradients of functions in the minimax reformulation. Each iteration needs several matrix vector products and thus the complexity in each iteration is $O(N)$. So the complexity of the algorithm for solving the reformulation in [14] is then $O(\frac{LN\|x_0-x^*\|^2}{\epsilon})$ (this may be roughly considered as $O(\frac{N\phi^4}{\epsilon\tilde{\xi}^3})$ due to $L \leq \phi$ and $\|x_0 - x^*\|^2 \leq (2R)^2 \leq O(\phi^3/\xi^3)$), which is worse than our complexity that is proportional to $\frac{N\phi^3}{\sqrt{\epsilon\tilde{\xi}^5}}$ in (4.1). Next let us give comparisons with the results in [2, 15, 21]. The method in [2] involves a process in simultaneously diagonalizing two matrices, whose computation cost is not given there. The paper [15] does not present complexity results for solving the GTRS and focuses on a numerically unstable case that $A$ and $B$ are not SD, which is different from our setting where $\mu A + (1-\mu)B \succeq \xi I$, implying $A$ and $B$ are SD. The algorithm in [21] solves an extreme generalized eigenpair of a parameterized matrix pencil for the GTRS at each step, whose iteration complexity is unknown. And how an inexact computation of reformulation or extreme eigenpairs will influence the solution accuracy is not provided in [2, 15, 21]. In summary, our methods represent the first provable linear-time algorithm for the GTRS.

As the GTRS is a generalization of the TRS and the generalized eigenvalue problem, it would be interesting to compare the complexity bounds between our algorithm and the algorithm in [9]. The complexity in [9] for the TRS with a generalized norm constraint $x^T Bx \leq 1$, where $B$ is a positive definite matrix, is

$$O\left(\frac{N\sqrt{\kappa}}{\epsilon}\right)\log\left(\frac{\kappa}{\epsilon}\right)\log\left(\frac{n}{\delta}\log\frac{\kappa}{\epsilon}\right)\log\frac{u-l}{\epsilon}, \tag{4.5}$$

where $u$ and $l$ are upper and lower bounds for the TRS. Note that the total complexity here has an additional $\log\frac{u-l}{\epsilon}$ compared to the results in Theorem 3 in [9] since the complexity in Theorem 3 in [9] is only for one iteration of the binary search. Recall that from (4.3) in the proof of Theorem 4.1, our complexity is

$$O\left(\frac{N\phi^3}{\sqrt{\epsilon\tilde{\xi}^5}}\log\left(\frac{n}{\delta}\log\frac{\phi}{\epsilon\tilde{\xi}}\right)\log\frac{\phi}{\epsilon\tilde{\xi}}\log\frac{u-l}{\epsilon}\right). \tag{4.6}$$

A comparison between (4.5) and (4.6) shows that the two algorithms have the same linear dependence of $N$ (the nonzero input) and a similar dependence of $\epsilon$, though in general the parameters $\phi, \tilde{\xi}$, and $\kappa$ in the bounds are incomparable.

**5. Conclusion.** In this paper, we have presented the first linear-time algorithm to approximately solve the generalized trust region subproblem, which extends the recent result in [9] for the trust region subproblem. The nonconvexity and noncompactness of the feasible region of the GTRS make such generalization nontrivial. By using several subroutines to obtain an initial estimation of lower and upper bounds and a compact region that contains the optimal solution, together with the linear-time SDP solver developed in [9], we developed Algorithm 4.1 to find an $\epsilon$ optimal solution of the GTRS in linear time. A byproduct of this paper is to provide a linear-time algorithm, PsdPencil, to detect a $\lambda$ such that $A + \lambda B \succ 0$ under mild conditions, which may be of independent interest for readers. Our future research will focus on extending the current algorithm to some variants of the GTRS with additional linear constraints or an additional unit ball constraint.

**Acknowledgment.** The authors would like to thank the two anonymous referees for the invaluable comments that improve the quality of the paper significantly.

## REFERENCES

[1] S. Adachi and Y. Nakatsukasa, *Eigenvalue-based algorithm and analysis for nonconvex QCQP with one constraint*, Math. Program., 173 (2019), pp. 79–116.
[2] A. Ben-Tal and D. den Hertog, *Hidden conic quadratic representation of some nonconvex quadratic optimization problems*, Math. Program., 143 (2014), pp. 1–29.
[3] A. Ben-Tal and M. Teboulle, *Hidden convexity in some nonconvex quadratically constrained quadratic programming*, Math. Program., 72 (1996), pp. 51–63.
[4] S. Burer and K. M. Anstreicher, *Second-order-cone constraints for extended trust-region subproblems*, SIAM J. Optim., 23 (2013), pp. 432–451.
[5] S. Burer and B. Yang, *The trust region subproblem with non-intersecting linear constraints*, Math. Program., 149 (2015), pp. 253–264.
[6] A. R. Conn, N. I. Gould, and P. L. Toint, *Trust Region Methods*, MPS-SIAM Ser. Optim. 1, SIAM, Philadelphia, 2000.
[7] 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.
[8] R. Ge, C. Jin, S. Kakade, P. Netrapalli, and A. Sidford, *Efficient algorithms for large-scale generalized eigenvector computation and canonical correlation analysis*, in Proceedings of the International Conference on Machine Learning, 2016, pp. 2741–2750.
[9] E. Hazan and T. Koren, *A linear-time algorithm for trust region problems*, Math. Program., 158 (2016), pp. 363–381.
[10] H. Hmam, *Quadratic Optimization with One Quadratic Equality Constraint.*, Tech. report, DSTO-TR-2416, Warfare and Radar Division DSTO Defence Science and Technology Organisation, Australia, 2010.
[11] N. Ho-Nguyen and F. Kilinc-Karzan, *A second-order cone based approach for solving the trust-region subproblem and its variants*, SIAM J. Optim., 27 (2017), pp. 1485–1512.
[12] K. Huang and N. D. Sidiropoulos, *Consensus-ADMM for general quadratically constrained quadratic programming*, IEEE Trans. Signal Process., 64, pp. 5297–5310.
[13] R. Jiang and D. Li, *Simultaneous diagonalization of matrices and its applications in quadratically constrained quadratic programming*, SIAM J. Optim., 26 (2016), pp. 1649–1668.
[14] R. Jiang and D. Li, *Novel reformulations and efficient algorithm for the generalized trust region subproblem*, SIAM J. Optim., 29 (2019), pp. 1603–1633.

[15] 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.
[16] J. Kuczyński and H. Woźniakowski, *Estimating the largest eigenvalue by the power and lanczos algorithms with a random start*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1094–1122.
[17] J. M. Martínez, *Local minimizers of quadratic functions on Euclidean balls and spheres*, SIAM J. Optim., 4 (1994), pp. 159–176.
[18] J. J. Moré, *Generalizations of the trust region problem*, Optim. Methods Softw., 2 (1993), pp. 189–209.
[19] J. J. Moré and D. C. Sorensen, *Computing a trust region step*, SIAM J. Sci. Stat. Comput., 4 (1983), pp. 553–572.
[20] I. Pólik and T. Terlaky, *A survey of the S-lemma*, SIAM Rev., 49 (2007), pp. 371–418.
[21] T. K. Pong and H. Wolkowicz, *The generalized trust region subproblem*, Comput. Optim. Appl., 58 (2014), pp. 273–322.
[22] 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.
[23] M. Salahi and A. Taati, *An efficient algorithm for solving the generalized trust region subproblem*, Comput. Appl. Math., 37 (2018), pp. 395–413.
[24] R. J. Stern and H. Wolkowicz, *Indefinite trust region subproblems and nonsymmetric eigenvalue perturbations*, SIAM J. Optim., 5 (1995), pp. 286–313.
[25] J. F. Sturm and S. Zhang, *On cones of nonnegative quadratic functions*, Math. Oper. Res., 28 (2003), pp. 246–267.
[26] 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.
[27] V. A. Yakubovich, *S-procedure in nonlinear control theory*, Vestn. Leningrad Univ., 1 (1971), pp. 62–77.
[28] B. Yang and S. Burer, *A two-variable approach to the two-trust-region subproblem*, SIAM J. Optim., 26 (2016), pp. 661–680.
[29] Y. Ye, *A new complexity result on minimization of a quadratic function with a sphere constraint*, in Recent Advances in Global Optimization, C. Floudas and P. Pardalos, eds., Princeton University Press, Princeton, NJ, 1992, pp. 19–31.
[30] Y. Ye and S. Zhang, *New results on quadratic minimization*, SIAM J. Optim., 14 (2003), pp. 245–267.
[31] H. Zhang, A. R. Conn, and K. Scheinberg, *A derivative-free algorithm for least-squares minimization*, SIAM J. Optim., 20 (2010), pp. 3555–3576.