

# 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 term of the number of non-zero 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., 2016, 158(1-2): 363-381]. However, due to the non-convexity and non-compactness 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.** the generalized trust region subproblem, semidefinite programming, linear time complexity, approximation algorithms

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

**1. Introduction.** We consider in this paper the following generalized trust region subproblem (GTRS),

$$\begin{aligned} \text{(GTRS)} \quad & \min f(x) := x^T A x + 2a^T x \\ & \text{s.t. } h(x) := x^T B x + 2b^T x + d \leq 0, \end{aligned}$$

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 attentions in the literature. The GTRS also admits its own applications such as time of arrival problems [10] and subproblems of consensus ADMM 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 be then 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

---

\*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

(GTRS) under the SD condition. Jiang et al. [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, while Hazan and Koren [9] has 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 comes from the non-convex constraint of the GTRS; while this challenging point is not present in the TRS as the constraint in the TRS is convex. More specifically, the non-convexity 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 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<sup>1</sup> 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  $c$ ,

$$(1.1) \quad \begin{aligned} x^T A x + 2a^T x &\leq c \\ x^T B x + 2b^T x + d &\leq 0, \end{aligned}$$

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 continuous on the compact feasible region [9]. However, a linear-time estimation of the lower and upper bounds are nontrivial in the GTRS. We propose 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.,

$$(1.2) \quad \begin{aligned} x^T A x + 2a^T x &\leq c - \epsilon \\ x^T B x + 2b^T x + d &\leq -\frac{\epsilon}{K}. \end{aligned}$$

---

<sup>1</sup>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 only happen when  $\lambda^*$  is an extreme eigenvalue of the matrix pencil  $(A, B)$ ; otherwise we are in the easy case.

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  $\{X : X \succeq 0, \text{tr}(X) \leq 1\}$ , where  $\text{tr}(X)$  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. We then illustrate the sub-routines 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.

**Notations** The notations  $A \succ 0$  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 also 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  $a$ . Notations  $\lambda_{\max}(A)$  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. 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.

**2.1. Preliminary.** Besides the linear-time algorithm for the TRS, Hazan and Koren [9] also demonstrated that their algorithm can be extended to the GTRS when the quadratic form in the constraint is 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  $x^T C x \geq \lambda_{\max}(C) - \epsilon$  and a scalar  $\lambda = x^T C x$  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 to the smallest eigenvalue. In fact, the negativity of the approximate largest eigenvalue for  $-C$  gives the smallest approximate eigenvalue of  $C$ .

**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  $x^T C x \leq \lambda_{\min}(C) + \epsilon$  and a scalar  $\lambda = x^T C x$  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]. 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 Assumptions 2.3 always holds. For ease of notation, we also define  $\tilde{\xi} = \min\{\xi, 1\}$  and

$$(2.1) \quad \phi = \rho_A + \rho_B + \|a\| + \|b\| + |d| + 1.$$

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  $g_i(x) = x^T Q_i x + 2p_i^T x + q_i$ , 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<sub>1</sub>) *There is no  $x \in \mathbb{R}^n$  such that  $g_1(x) \leq 0$  and  $g_2(x) < 0$ .*

(S<sub>2</sub>) *There exists a nonnegative multiplier  $\lambda \geq 0$  such that  $g_1(x) + \lambda g_2(x) \geq 0$ ,  $\forall x \in \mathbb{R}^n$ .*

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,*

$$(2.2) \quad \begin{aligned} x^T A x + 2a^T x &\leq c - 2\epsilon, \\ x^T B x + 2b^T x + d &\leq 0. \end{aligned}$$

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

$$\begin{aligned} 0 &\leq x^T A x + 2a^T x - c + \epsilon + \lambda(x^T B x + 2b^T x + d + \epsilon/K) \\ &= x^T A x + 2a^T x - c + 2\epsilon + \lambda(x^T B x + 2b^T x + d) - \epsilon, \end{aligned}$$

for all  $x \in \mathbb{R}^n$ , where  $\varepsilon = (1 - \lambda/K)\epsilon > 0$ .

Due to  $K > \lambda \geq 0$ , for all  $x \in \mathbb{R}^n$ , we have

$$\begin{aligned} & x^T Ax + 2a^T x - c + 2\epsilon + \lambda(x^T Bx + 2b^T x + d - \varepsilon/K) \\ & \geq x^T Ax + 2a^T x - c + 2\epsilon + \lambda(x^T Bx + 2b^T x + d) - \varepsilon \geq 0. \end{aligned}$$

Hence for all  $\lambda \geq 0$ , we have

$$x^T Ax + 2a^T x - c + 2\epsilon + \lambda(x^T Bx + 2b^T x + d - \varepsilon/K) \geq 0, \quad \forall x.$$

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

$$\begin{aligned} x^T Ax + 2a^T x & \leq c - 2\epsilon, \\ x^T Bx + 2b^T x + d - \varepsilon/K & < 0 \end{aligned}$$

is infeasible. This further implies that the system (2.2) is infeasible.  $\square$

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.,  $\lambda_{\max}(A) \leq \rho_A$ , 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  $\mu\lambda_{\max}(A) + (1 - \mu)\lambda_{\min}B \succ 0$  due to  $\mu A + (1 - \mu)B \succ 0$ . This, together with  $\|A\|_2 \leq \rho_A$  and  $\lambda_{\min}(B) \leq -\xi$ , gives rise to  $0 < -\lambda_{\max}(A)/\lambda_{\min}(B) \leq \rho_A/\xi$  and thus the estimated  $K$  can be restricted to be upper bounded by some positive constant (e.g.,  $\frac{\rho_A}{\xi} + 1$ ). Hence we set  $K = \frac{\rho_A}{\xi} + 1$  in our algorithm for simplicity.

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

**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 in the following subsections. We first identify an interval  $(\mu_1, \mu_2]$  where the targeted  $\mu$  is located, if exists. In Algorithm 3.1, we initialize  $(\mu_1, \mu_2]$  as  $(0, 1]$ . In each step, we invoke the eigenvalue oracle  $(\lambda, x) = \text{APPROXMINEV}(\mu A + (1 - \mu)B, \xi/4, \delta/T)$  to find an approximate smallest eigenvalue of the midpoint of this interval, where  $x$  is a unit vector such that  $\lambda = x^T Ax$  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  $x^T Ax > x^T Bx$ , or  $(\mu, \mu_2]$  if  $x^T Ax \leq x^T Bx$ . The intuition of this step is that for all  $\nu \in (\mu_1, \mu]$ , we have  $\lambda_{\min}(\nu A + (1 - \nu)B) \leq x^T(\nu A + (1 - \nu)B)x \leq x^T(\mu A + (1 - \mu)B)x = \lambda$  when  $x^T Ax > x^T Bx$ . This means that the target  $\mu$ , if 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.** *Suppose that Assumption 2.3 holds.  $\text{PSDPENCIL}(A, B, \xi, \phi, \delta)$  takes at most  $\log_2 \left( \frac{8\phi}{\xi} \right)$  iterations of  $\text{APPROXMINEV}$  and returns  $(\mu, \lambda, x)$  such that  $\lambda = x^T(\mu A + (1 - \mu)B)x \geq 3\xi/4$  and  $\lambda_{\min}(\mu A + (1 - \mu)B) \geq \lambda - \xi/4$ . The output is correct with probability of at least  $1 - \delta$  and the total runtime is*

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

---

**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  $\|A\|_2 \leq \rho_A$  and  $\|B\|_2 \leq \rho_B$ ,  $\phi$  in (2.1) and  $\xi, \delta > 0$

**Output:**  $\mu > 0$  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  $\lambda_{\min}(\mu A + \mu B) \leq \lambda \leq \lambda_{\min}(\mu A + \mu B) + \xi/4$ ; output is correct with probability of at least  $1 - \delta$

```

1: function PSDPENCIL( $A, B, \xi, \phi, \delta$ )
2:   initialize  $T = \log_2 \frac{8\phi}{\xi}$ ,  $\mu_1 = 0$ ,  $\mu_2 = 1$ 
3:   for  $i = 1 : T$  do
4:      $\mu = (\mu_1 + \mu_2)/2$ 
5:     invoke  $(\lambda, x) \leftarrow \text{APPROXMINEV}(\mu A + (1 - \mu)B, \xi/4, \delta/T)$ 
6:     if  $\lambda \geq 3\xi/4$  then
7:       return  $(\mu, \lambda, x)$ 
8:     else if  $x^T A x > x^T B x$  then
9:       update  $\mu_1 \leftarrow \mu$  ▷ update  $x_1 \leftarrow x$  for analysis in Theorem 3.1
10:    else
11:      update  $\mu_2 \leftarrow \mu$  ▷ update  $x_2 \leftarrow x$  for analysis in Theorem 3.1
12:    end if
13:  end for
14:  return “Assumption 2.3 fails.”
15: end function

```

---

*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  $x^T A x$  and  $x^T B x$ ) is dominated by APPROXMINEV. Since  $\|\mu A + (1 - \mu)B\|_2 \leq \mu \|A\|_2 + (1 - \mu) \|B\|_2 \leq \phi$ , 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  $O\left(\frac{N\sqrt{\phi}}{\sqrt{\xi}} \log \left(\frac{n}{\delta} \log \frac{\phi}{\xi}\right)\right)$ . 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  $(\mu, \lambda, x)$  for some iteration  $i \leq T$ , then the returned  $\mu$  is the one as required, i.e.,  $\lambda_{\min}(\mu A + (1 - \mu)B) \geq \lambda - \xi/4$ . Now it suffices to prove that under Assumption 2.3, the algorithm must terminate in some iteration  $i \leq T$ .

Recall that Assumption 2.3 states that  $\exists \mu_0 > 0$  such that  $\mu_0 A + (1 - \mu_0)B \succeq \xi I$ . Since  $\|A\|_2 \leq \rho_A$  and  $\|B\|_2 \leq \rho_B$ , 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  $\phi \geq \rho_A + \rho_B$ . 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  $x_1^T C(\mu)x_1 = \mu x_1^T A x_1 + (1 - \mu)x_1^T B x_1 \leq \mu_1 x_1^T A x_1 + (1 - \mu_1)x_1^T B x_1 < 3\xi/4$  for  $\mu \leq \mu_1$  because  $x_1^T A x_1 > x_1^T B x_1$ . Furthermore,  $\lambda_{\min}(C(\mu)) \leq x_1^T C(\mu)x_1 < 3\xi/4$  for  $\mu \leq \mu_1$ . Similarly, we have  $\lambda_{\min}(C(\mu)) < 3\xi/4$  for  $\mu \geq \mu_2$ . So if the algorithm terminates in line 14, we have  $\mu_2 - \mu_1 \leq \frac{\xi}{8\phi}$  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  $\{\mu : \lambda_{\min}(\mu A + (1 - \mu)B) \geq 3\xi/4\}$  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 that each output of APPROXMINEV is correct with probability of at least  $1 - \delta/T$  and that the total iteration number is at most  $T$ .  $\square$

**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).

---

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

---

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

**Output:**  $\mu_0 > 0$  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  $\lambda_{\min}(\mu_0 A + \mu_0 B) \leq \lambda \leq \lambda_{\min}(\mu_0 A + \mu_0 B) + \xi/4$  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  $(\mu_0, \lambda, x) \leftarrow \text{PSDPENCIL}(A, B, \xi, \phi, \delta/2)$ 
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  $(\varpi, y) \leftarrow \text{APPROXMINEV}(B, \xi/2, \delta/2)$ 
6:   find a root  $\alpha$  to (3.2)
7:   set  $u = f(\alpha y)$  ▷ initial upper bound
8:   return  $(\mu_0, \lambda, l, u)$ 
9: end function

```

---

An upper bound for problem (GTRS) is given by  $u = f(\alpha y)$ , where  $\alpha$  is the positive solution of the following quadratic inequality and  $y$  is the unit eigenvector returned by  $(\varpi, y) \leftarrow \text{APPROXMINEV}(B, \xi/2, \delta/2)$ ,

$$(3.1) \quad (\alpha y)^T B (\alpha y) + 2b^T(\alpha y) + d \leq 0.$$

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

$$(3.2) \quad \varpi \alpha^2 + 2b^T y \alpha + d = 0.$$

The root of the above equation is  $\alpha = \frac{-2b^T y \pm \sqrt{4(b^T y)^2 + 4\varpi d}}{-2\varpi}$  with  $|\alpha| \leq \frac{4\|b\| + \sqrt{4|\varpi d|}}{2|\varpi|} \leq \frac{4\|b\|}{\xi} + \frac{\sqrt{2|d|}}{\sqrt{\xi}}$ . Due to  $y^T B y \leq \varpi$ , either choice of  $\alpha$  is feasible for (3.1). This yields



a feasible solution  $\alpha y$  for (GTRS). Then we have  $|f(\alpha y)| = |\alpha^2 y^T A y + 2\alpha a^T y| \leq \alpha^2 \|A\|_2 + 2|\alpha| \|a\| \leq \left( \frac{4\|b\|}{\xi} + \frac{\sqrt{2|d|}}{\sqrt{\xi}} \right)^2 \rho_A + 2 \left( \frac{4\|b\|}{\xi} + \frac{\sqrt{2|d|}}{\sqrt{\xi}} \right) \|a\| = O(\frac{\phi^3}{\xi^2})$ , recalling that  $\tilde{\xi} = \min\{\xi, 1\}$ . The vector 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  $O(\frac{N\sqrt{\phi}}{\sqrt{\xi}} \log \frac{n}{\delta})$ . The output is correct with probability of at least  $1 - \delta/2$ .

Next we illustrate that a lower bound can be found by a feasible solution of the Lagrangian dual problem. Note that  $(\mu_0, \lambda, x) \leftarrow \text{PSDPENCIL}(A, B, \xi, \phi, \delta/2)$  from line 2 in Algorithm 3.2. From Theorem 3.1 and Algorithm 3.2,  $\lambda$  is a  $\xi/4$  approximate smallest eigenvalue satisfying  $\lambda \geq 3\xi/4$  under Assumption 2.3 and the optput is correct with probability of at least  $1 - \delta/2$ . Since the smallest eigenvalue of  $\mu_0 A + (1 - \mu_0)B$  satisfies  $\lambda_{\min}(\mu_0 A + (1 - \mu_0)B) \geq \lambda - \xi/4 \geq \xi/2$ , we have  $\mu_0 \rho_A + (1 - \mu_0)(-\xi) \geq \xi/2$  and thus  $1 \geq \mu_0 \geq \frac{3\xi}{2(\rho_A + \xi)}$ . The Lagrangian dual problem of (GTRS) is

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

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,

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

Note that  $1 \geq \mu_0 \geq \frac{3\xi}{2(\rho_A + \xi)}$  implies  $0 \leq \nu_0 \leq 2\rho_A/3\xi$  and further  $\|p\| \leq \|a\| + 2\rho_A\|b\|/3\xi$ . Note also that the main time cost for computing the lower bound is in calling PSDPENCIL, which runs in time, by Theorem 3.1,

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

To summarise, in this subsection, we have demonstrated that

$$u - l \leq O\left(\frac{\phi^3}{\xi^2}\right) + 2\rho_A|d|/3\xi + \frac{\mu_0(\|a\| + \|b\|(1/\mu_0 - 1))^2}{2\xi},$$

i.e.,

$$(3.3) \quad u - l \leq O\left(\frac{\phi^3}{\xi^2}\right),$$



and the total runtime of  $\text{BOUNDS}(A, B, \xi, a, b, \phi, \delta)$  is

$$(3.4) \quad O\left(\frac{N\sqrt{\phi}}{\sqrt{\xi}} \log \frac{n}{\delta} + \frac{N\sqrt{\phi}}{\sqrt{\xi}} \log\left(\frac{n}{\delta} \log \frac{1}{\xi}\right) \log \frac{1}{\xi}\right) = O\left(\frac{N\sqrt{\phi}}{\sqrt{\xi}} \log\left(\frac{n}{\delta} \log \frac{1}{\xi}\right) \log \frac{1}{\xi}\right).$$

As we call  $\text{PSDPENCIL}(A, B, \xi, \phi, \delta/2)$  and  $\text{APPROXMINEV}(B, \xi/2, \delta/2)$  once each, the output of  $\text{BOUNDS}(A, B, \xi, a, b, \phi, \delta)$  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  $\text{BOUNDS}(A, B, \xi, a, b, d, \phi, \delta)$  computes lower and upper bounds for (GTRS) satisfying*

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

and the total runtime is,

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

**3.3. Identify feasibility of quadratic systems.** This subsection shows that line 6 of Algorithm 4.1 returns an upper bound of the Euclidean norm of the optimal solution and using this bound, the subroutine FEAS can help us to 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:

$$(3.5) \quad D_i \bullet X \geq \epsilon, \quad i = 1, 2, \quad X \in \mathcal{K},$$

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

**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 (3.5) 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

$$(3.6) \quad P_i \bullet X \geq \epsilon, \quad i = 1, 2, \quad X_{11} = 1, \quad X \succeq 0,$$

where  $P_1 = \begin{pmatrix} c & -a^T \\ -a & -A \end{pmatrix}$ ,  $P_2 = K \begin{pmatrix} -d & -b^T \\ -b & -B \end{pmatrix}$ . 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 3,  $\text{BOUNDS}(A, B, \xi, a, b, \phi, \delta/2)$ , of Algorithm 4.1,  $R$  be defined by (4.1), 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,  $R \leq O(\phi^{3/2}/\xi^{3/2})$ .*

*Proof.* Let the notations be the same as in Section 3.2. Hence the output is correct with probability of at least  $1 - \delta/2$ . 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  $f_1(x^*) - u + \nu_0 f_2(x^*) \leq 0$ , where  $u$  is an upper bound of (GTRS) derived from line 3 or 11 of Algorithm 4.1, and thus

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

Let  $\mathcal{X} = \{x : x^T(A + \nu_0 B)x + 2(a + \nu_0 b)^T x - u + \nu_0 d \leq 0\}$ . Then  $\mathcal{X} \subset \{x : \frac{\bar{\lambda}}{\mu_0} x^T x + 2(a + \nu_0 b)^T x - u + \nu_0 d \leq 0\}$ . This further implies

$$\mathcal{X} \subset \left\{ x : \|x\| \leq \frac{\mu_0 \|a + \nu_0 b\|}{\bar{\lambda}} + \sqrt{\left\| \frac{\mu_0}{\bar{\lambda}} (a + \nu_0 b) \right\|^2 + \frac{\mu_0}{\bar{\lambda}} (u - \nu_0 d)} \right\}.$$

From the arguments in Section 3.2, we have  $3\xi/2(\rho_A + \xi) \leq \mu_0 \leq 1$ . Hence, together with  $\bar{\lambda} \geq \xi/2$ ,  $\mu_0 \nu_0 = 1 - \mu_0 \leq 1$  and  $u \leq O(\phi^3/\xi^2)$ , it holds that  $R \leq O(\phi^{3/2}/\xi^{3/2})$ .  $\square$

By defining  $Y^* = \begin{pmatrix} 1 \\ x^* \end{pmatrix} \begin{pmatrix} 1 \\ x^* \end{pmatrix}^T / S$ , we have

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

i.e.,  $Y^* \in \mathcal{K}$ , where  $S = R^2 + 1$ . This motivates us to solve the following SDP system instead of (3.6),

$$(3.7) \quad \frac{1}{\kappa} P_i \bullet Y \geq \frac{\epsilon}{\kappa S}, \quad i = 1, 2, \quad Y \in \mathcal{K},$$

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  $\|P_i/\kappa\|_2 \leq 1, i = 1, 2$  and the optimal solution, if exists,  $Y^* \in \mathcal{K}$ . Therefore, the SDP feasibility problem (3.7) 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 (3.7) is infeasible, which is further equivalent to the infeasibility of (3.6), or returns  $Y \in \mathcal{K}$  such that  $\frac{1}{\kappa} P_i \bullet Y \geq \frac{\epsilon}{2\kappa S}, i = 1, 2$ , which is further equivalent to that  $X = SY$  satisfies  $P_i \bullet X \geq \epsilon/2, i = 1, 2, X \succeq 0$ . When RELAXSOLVE( $\frac{P_1}{\kappa}, \frac{P_2}{\kappa}, \frac{\epsilon}{\kappa S}, \delta$ ) returns  $Y \in \mathcal{K}$  such that  $\frac{1}{\kappa} P_i \bullet Y \geq \frac{\epsilon}{2\kappa S}$ , 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  $z^T P_i z \geq \epsilon/2r, i = 1, 2$  with  $r = \text{rank}(Y) \leq 2$  as there are only two inequalities in (3.7).

LEMMA 3.5 ([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  $M \bullet X \geq a$ , SZROTATION outputs a decomposition  $X = \sum_{i=1}^r y_i y_i^T$  such that  $y_i^T M y_i \geq a/r$  for all  $i = 1, \dots, r$ . The procedure runs in time  $O(Nr)$ , where  $N \geq n$  is the number of non-zero entries in  $M$ .*

THEOREM 3.6. *Given the linear-time SDP solver RELAXSOLVE, parameters  $\epsilon, \delta > 0$ , and  $Q_1$  and  $Q_2$  defined in (3.8), 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*

---

**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  $\|A\|_2 \leq \rho_A$  and  $\|B\|_2 \leq \rho_B$ ,  
 $a, b, c, d, \epsilon, \delta, \mu_A, \mu_B, K, R > 0$

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

```

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

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

5:   invoke RELAXSOLVE( $Q_1, Q_2, \epsilon/(\kappa S), \delta$ )
6:   if RELAXSOLVE returns “infeasible” then
7:     return “infeasible”
8:   else {RELAXSOLVE returns  $Y$  such that  $Q_i \bullet Y \geq \frac{\epsilon}{2\kappa S}, i = 1, 2,$ }
9:     invoke SZROTATION( $Q_1, SY$ ) that return  $X = \sum_{i=1}^r z_i z_i^T$  as output
10:    if  $r = 1$  then
11:       $z = z_1$ 
12:    else ▷ r=2
13:      find a vector  $z \in \{z_1, z_2, \dots, z_r\}$  for which  $z^T Q_2 z \geq \epsilon/2r$  and let  $\tilde{z} =$ 
 $z(2 : n + 1)$ 
14:    end if
15:    if  $z(1) \neq 0$  then
16:       $x = \tilde{z}/z(1)$ 
17:    else
18:      set  $\alpha = \min\{\frac{\kappa\epsilon}{2r(|2b^T \tilde{z}| + |d|)}, \frac{\kappa\epsilon}{2Kr(|2a^T \tilde{z}| + |c|)}, 1\}$ ;  $x = \tilde{z}/\alpha$ 
19:    end if
20:    return  $x$ 
21:  end if
22: end function

```

---

*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 sub-algorithms RELAXSOLVE and SZROTATION with  $O\left(\frac{N}{\sqrt{\epsilon'}} \log\left(\frac{n}{\delta} \log \frac{1}{\epsilon'}\right) \log \frac{1}{\epsilon'}\right)$  and  $O(Nr)$ , respectively, where  $\epsilon' = \epsilon/(\kappa S)$  and  $r \leq 2$  is the rank of  $X$  returned by RELAXSOLVE. Since  $O(Nr)$  is dominated by  $O\left(\frac{N}{\sqrt{\epsilon'}} \log\left(\frac{n}{\delta} \log \frac{1}{\epsilon'}\right) \log \frac{1}{\epsilon'}\right)$ , 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  $\kappa = \max\{\mu_A, K\mu_B\} \leq O(\phi^3/\tilde{\xi}^2)$  and  $S = R^2 + 1 \leq O(\phi^3/\tilde{\xi}^3)$ . The

bound for  $\kappa$  follows that  $\mu_A = \rho_A + 2\|a\| + |c| \leq 2\phi + |u| + |l| \leq O(\phi^3/\tilde{\xi}^2)$ ,  $K = O(\phi/\xi)$  and  $\mu_B = \rho_B + 2\|b\| + |d| = O(\phi)$ .

*Correctness:* If RELAXSOLVE returns “infeasible”, the SDP relaxation (3.7) is infeasible and thus (3.6) is infeasible. This implies the infeasibility of (1.2) since (3.6) 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  $Q_i \bullet Y \geq \frac{\epsilon}{2\kappa S}$ ,  $i = 1, 2$ . Then  $X = SY$  satisfies  $P_i \bullet X \geq \epsilon/2\kappa$ ,  $i = 1, 2$ ,  $X \succeq 0$ . 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  $\text{rank}(Y) \leq 2$  and thus  $\text{rank}(X) \leq 2$ . It follows from Lemma 3.5 that invoking SZROTATION( $Q_1, X$ ) yields a solution  $X = \sum_{i=1}^r z_i z_i^T$  such that  $z_i^T Q_1 z_i \geq \epsilon/2r\kappa$ ,  $i = 1, 2$ . Together with  $Q_2 \bullet X \geq \epsilon/2\kappa$ , we conclude that at least one of  $z_i$  satisfies  $z_i^T Q_2 z_i \geq \epsilon/2r\kappa$ . So SZROTATION( $Q_1, X$ ) indeed finds a vector  $z \in \{z_1, z_2\}$  such that  $z^T Q_j z \geq \epsilon/2r\kappa$ , ( $j = 1, 2$ ), which further implies that

$$\begin{aligned} \tilde{z}^T A \tilde{z} + 2z(1)a^T \tilde{z} &\leq cz(1)^2 - \epsilon/2r\kappa, \\ \tilde{z}^T B \tilde{z} + 2z(1)b^T \tilde{z} + z(1)^2 d &\leq -\epsilon/2rK\kappa, \end{aligned}$$

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

If  $z(1) \neq 0$ , 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

$$\begin{aligned} x^T A x + 2a^T x - c &\leq -\epsilon/2r\kappa z(1)^2, \\ x^T B x + 2b^T x + d &\leq -\epsilon/2rK\kappa z(1)^2. \end{aligned}$$

Then we have

$$\begin{aligned} x^T A x + 2a^T x &\leq c, \\ x^T B x + 2b^T x + d &\leq 0 \end{aligned}$$

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

$$\begin{aligned} \tilde{z}^T A \tilde{z} &\leq -\epsilon/2r\kappa, \\ \tilde{z}^T B \tilde{z} &\leq -\epsilon/2rK\kappa. \end{aligned}$$

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

$$\begin{aligned} x^T A x &= \frac{\tilde{z}^T A \tilde{z}}{\alpha^2} \\ &\leq -\frac{\epsilon}{2r\kappa\alpha^2} \\ &\leq -\frac{|2a^T \tilde{z}| + |c|}{\alpha} \quad (\text{due to } -1/\alpha \leq -\frac{2r\kappa(|2a^T \tilde{z}| + |c|)}{\epsilon}) \\ &\leq -|2a^T \tilde{z}/\alpha| - |c|. \quad (\text{due to } \alpha \leq 1) \end{aligned}$$

Thus,  $x^T A x + 2a^T x - c \leq x^T A x + 2|a^T \tilde{z}/\alpha| + |c| \leq 0$ . Similarly, we have

$$x^T B x + 2b^T x + d \leq 0.$$

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

**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  $f(\tilde{x}) \leq v^* + \epsilon$ , where  $v^*$  is an optimal value of (GTRS).

---

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

---

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

**Output:** an  $\epsilon$  optimal solution; 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  $K = \frac{\rho_A}{\xi} + 1$  and  $\phi = \rho_A + \rho_B + \|a\| + \|b\| + |d| + 1$ 
3:   invoke  $(\mu_0, \lambda, l, u) \leftarrow \text{BOUNDS}(A, B, \xi, a, b, \phi, \delta/2)$ 
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

```

$$(4.1) \quad R = \frac{\mu_0 \|a + \nu_0 b\|}{\bar{\lambda}} + \sqrt{\left( \left\| \frac{\mu_0}{\bar{\lambda}} (a + \nu_0 b) \right\|^2 + \frac{\mu_0}{\bar{\lambda}} (u - \nu_0 d) \right)}$$

```

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  $u = \min\{u, x^T A x + 2a^T x\}$ ,  $c = (l + u)/2$  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.** *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  $f(\tilde{x}) \leq v^* + \epsilon$ , where  $v^*$  is an optimal value of (GTRS). The total runtime is*

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

*Proof. Correctness:* Section 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  $u - l \leq O(\phi^3/\xi^2)$  from (3.3).

From Theorem 3.6, we know that the subroutine FEAS either returns a feasible solution for (1.1) (yielding a new upper bound  $u = \min\{c, x^T A x + 2a^T x\}$ ) or declares the infeasibility of (1.2) (yielding a new lower bound  $l = c - 2\epsilon'$  by Lemma 2.5). 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, the length of  $u_T - l_T$  satisfies

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

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

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

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

*Runtime:* The main runtime of Algorithm 4.1 is in subroutines BOUNDS and FEAS. The equation (3.4) in Section 3.2 shows that the main operations in BOUNDS( $A, B, \xi, a, b, \phi, \delta/2$ ) run in time

$$(4.2) \quad O\left(\frac{N\sqrt{\phi}}{\sqrt{\xi}} \log\left(\frac{n}{\delta} \log \frac{1}{\xi}\right) \log \frac{1}{\xi}\right).$$

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 Subsection 3.3, the total time of lines 5-14 is

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

which is equivalent to,

$$(4.3) \quad O\left(\frac{N\phi^3}{\sqrt{\epsilon\xi^5}} \log\left(\frac{n}{\delta} \log \frac{\phi}{\epsilon\xi}\right) \log \frac{\phi}{\epsilon\xi} \log \frac{\phi}{\epsilon\xi}\right),$$

by noting  $u - l \leq O(\phi^3/\xi^2)$ ,  $\delta' = \delta/2T$ ,  $T = \log_2((u-l)/\epsilon')$  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  $1 - \delta$ . Combining the runtime of (4.2) and (4.3), we conclude that the total runtime is

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

□

It is interesting to compare our linear-time algorithm to existing algorithms in the literature. The algorithms in [1, 14] and our algorithm all require the regularity condition  $A + \lambda B \succ 0$ . However, both [1] and [14] do not give a way to compute such a  $\lambda$  in linear time; While our subroutine PSDPENCIL gives a linear-time algorithm for such computation. Another drawback of the algorithms in [1] and [14] is that they both require exact computation of an extreme eigenpair and it is unknown how much the solutions of (GTRS) will be perturbed if an inexact eigenpair was used. Though in

practice, [14] and [1] use an inexact eigenpair, their theoretical guarantee is missing. Besides, the algorithm in [14] requires  $O(\frac{L\|x_0-x^*\|^2}{\epsilon})$  iterations to solve a minimax reformulation of 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 total complexity is then  $O(\frac{L\|x_0-x^*\|^2}{\epsilon})$  (this may be roughly considered as  $O(\frac{\phi^4}{\epsilon\xi^3})$  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  $\frac{N\phi^3}{\sqrt{\epsilon\xi^5}}$  as shown in Theorem 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 is not given there. The paper [15] considers a case that two matrices are not simultaneously diagonalizable, a case that is numerically unstable, which is not considered in our setting. 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 final result is not provided in [2, 15, 21]. In summary, our methods represent the first provable linear-time algorithm for the GTRS.

**5. Conclusion.** In this paper, we have presented the first linear-time algorithm to approximately solve the generalize trust region subproblem, which extends the recent result in [9] for the trust region subproblem. Our algorithm avoids diagonalization or factorization of matrices as that in [9]. Our algorithm also has the same time complexity as in the linear-time algorithm for TRS in [9] as well as in generalized eigenvector computation [8]. Similar to [9], our algorithm avoids the “hard case” by using an approximate linear-time SDP solver. 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 some 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.

**Acknowledgements.** This research was partially supported by Shanghai Sailing Program 18YF1401700, Natural Science Foundation of China (NSFC) 11801087 and Hong Kong Research Grants Council under Grants 14213716 and 14202017. 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*, Mathematical Programming, (2017), pp. 1–38.
- [2] A. BEN-TAL AND D. DEN HERTOG, *Hidden conic quadratic representation of some nonconvex quadratic optimization problems*, Mathematical Programming, 143 (2014), pp. 1–29.
- [3] A. BEN-TAL AND M. TEBoulLE, *Hidden convexity in some nonconvex quadratically constrained quadratic programming*, Mathematical Programming, 72 (1996), pp. 51–63.
- [4] S. BURER AND K. M. ANSTREICHER, *Second-order-cone constraints for extended trust-region subproblems*, SIAM Journal on Optimization, 23 (2013), pp. 432–451.
- [5] S. BURER AND B. YANG, *The trust region subproblem with non-intersecting linear constraints*, Mathematical Programming, 149 (2015), pp. 253–264.
- [6] A. R. CONN, N. I. GOULD, AND P. L. TOINT, *Trust Region Methods*, vol. 1, Society for Industrial and Applied Mathematics (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*, Journal of Global Optimization,



- 54 (2012), pp. 275–293.
- [8] R. GE, C. JIN, P. NETRAPALLI, A. SIDFORD, ET AL., *Efficient algorithms for large-scale generalized eigenvector computation and canonical correlation analysis*, in International Conference on Machine Learning, 2016, pp. 2741–2750.
- [9] E. HAZAN AND T. KOREN, *A linear-time algorithm for trust region problems*, Mathematical Programming, 158 (2016), pp. 363–381.
- [10] H. HMAM, *Quadratic Optimization with One Quadratic Equality Constraint.*, tech. report, Warfare and Radar Division DSTO Defence Science and Technology Organisation, Australia, Report DSTO-TR-2416, 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 Journal on Optimization, 27 (2017), pp. 1485–1512.
- [12] K. HUANG AND N. D. SIDIROPOULOS, *Consensus-admm for general quadratically constrained quadratic programming*, IEEE Transactions on Signal Processing, 64, pp. 5297–5310.
- [13] R. JIANG AND D. LI, *Simultaneous Diagonalization of Matrices and Its Applications in Quadratically Constrained Quadratic Programming*, SIAM Journal on Optimization, 26 (2016), pp. 1649–1668.
- [14] R. JIANG AND D. LI, *Novel reformulations and efficient algorithm for the generalized trust region subproblem*, accepted by SIAM Journal on Optimization, (2019).
- [15] R. JIANG, D. LI, AND B. WU, *SOCp reformulation for the generalized trust region subproblem via a canonical form of two symmetric matrices*, Mathematical Programming, 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 Journal on Matrix Analysis and Applications, 13 (1992), pp. 1094–1122.
- [17] J. M. MARTÍNEZ, *Local minimizers of quadratic functions on Euclidean balls and spheres*, SIAM Journal on Optimization, 4 (1994), pp. 159–176.
- [18] J. J. MORÉ, *Generalizations of the trust region problem*, Optimization Methods and Software, 2 (1993), pp. 189–209.
- [19] J. J. MORÉ AND D. C. SORESENSEN, *Computing a trust region step*, SIAM Journal on Scientific and Statistical Computing, 4 (1983), pp. 553–572.
- [20] I. PÓLIK AND T. TERLAKY, *A survey of the S-lemma*, SIAM Review, 49 (2007), pp. 371–418.
- [21] T. K. PONG AND H. WOLKOWICZ, *The generalized trust region subproblem*, Computational Optimization and Applications, 58 (2014), pp. 273–322.
- [22] F. RENDL AND H. WOLKOWICZ, *A semidefinite framework for trust region subproblems with applications to large scale minimization*, Mathematical Programming, 77 (1997), pp. 273–299.
- [23] M. SALAH AND A. TAATI, *An efficient algorithm for solving the generalized trust region subproblem*, Computational and Applied Mathematics, 37 (2018), pp. 395–413.
- [24] R. J. STERN AND H. WOLKOWICZ, *Indefinite trust region subproblems and nonsymmetric eigenvalue perturbations*, SIAM Journal on Optimization, 5 (1995), pp. 286–313.
- [25] J. F. STURM AND S. ZHANG, *On cones of nonnegative quadratic functions*, Mathematics of Operations Research, 28 (2003), pp. 246–267.
- [26] J. WANG AND Y. XIA, *A linear-time algorithm for the trust region subproblem based on hidden convexity*, Optimization Letters, (2016), pp. 1–8.
- [27] V. A. YAKUBOVICH, *S-procedure in nonlinear control theory*, Vestnik Leningrad University, 1 (1971), pp. 62–77.
- [28] B. YANG AND S. BURER, *A Two-Variable Approach to the Two-Trust-Region Subproblem*, tech. report, Tech. Report, Department of Mathematics, University of Iowa, 2013.
- [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, ed., Princeton University Press, 1992, pp. 19–31.
- [30] Y. YE AND S. ZHANG, *New results on quadratic minimization*, SIAM Journal on Optimization, 14 (2003), pp. 245–267.
- [31] H. ZHANG, A. R. CONN, AND K. SCHEINBERG, *A derivative-free algorithm for least-squares minimization*, SIAM Journal on Optimization, 20 (2010), pp. 3555–3576.