A Semidefinite Framework for Trust Region Subproblems with Applications to Large Scale Minimization *

Franz Rendl † Henry Wolkowicz ‡

February 26, 2002

Revision of CORR Report 94-32.

Abstract

Primal-dual pairs of semidefinite programs provide a general framework for the theory and algorithms for the trust region subproblem (TRS). This latter problem consists in minimizing a general quadratic function subject to a convex quadratic constraint and, therefore, it is a generalization of the minimum eigenvalue problem. The importance of (TRS) is due to the fact that it provides the step in trust region minimization algorithms. The semidefinite framework is studied as an interesting instance of semidefinite programming as well as a tool for viewing known algorithms and deriving new algorithms for (TRS). In particular, a dual simplex type method is studied that solves (TRS) as a parametric eigenvalue problem. This method uses the Lanczos algorithm for the smallest eigenvalue as a black box. Therefore, the essential cost of the algorithm is the matrix-vector multiplication and, thus, sparsity can be exploited. A primal simplex type method provides steps for the so-called hard case. Extensive numerical tests for

*This is an abbreviated revision of the University of Waterloo research report CORR 94-32.

†Technische Universität Graz, Institut für Mathematik, Kopernikusgasse 24, A-8010 Graz, Austria. Research support by Christian Doppler Laboratorium für Diskrete Optimierung (rendl@fmatbds01.tu-graz.ac.at).

‡University of Waterloo, Department of Combinatorics and Optimization, Waterloo, Ontario N2L 3G1, Canada (E-mail: hwolkowi@orion.uwaterloo.ca and URL: http://orion.uwaterloo.ca/~hwolkowi). Research support by the National Science and Engineering Research Council Canada.

large sparse problems are discussed. These tests show that the cost of the algorithm is $1+\alpha(n)$ times the cost of finding a minimum eigenvalue using the Lanczos algorithm, where $0<\alpha(n)<1$ is a fraction which decreases as the dimension increases.

Key words: trust region subproblems, parametric programming, semidefinite programming, min-max eigenvalue problems, large scale minimization.

AMS 1991 Subject Classification: Primary: 49M37, 90C30; Secondary: 90C06, 65K10.

1 Introduction

The trust region subproblem, (TRS), consists in minimizing a quadratic (possibly nonconvex) function subject to a quadratic (or norm) constraint. We present an efficient algorithm for this problem that can exploit sparsity. The algorithm is based on a parametric eigenvalue problem within a semidefinite programming, SDP, framework. We include two pairs of primal-dual SDPs. These programs provide a transparent framework for our algorithm as well as for current algorithms for (TRS).

Let $q(x) := x^tAx - 2a^tx$, where $A = A^t$ is a symmetric real $n \times n$ matrix and $a \in \Re^n$. And let $s > 0$. Computation of the step between iterates, in trust region algorithms for minimization, requires solution of the *trust region subproblem*

$$
\begin{array}{llll}
(TRS) & \mu^* := & \min & q(x) \qquad (1.1)\\
 & & \text{subject to} & x^tx = s^2 \quad (\leq s^2). \qquad (1.2)
\end{array}
$$

(For simplicity of exposition, (TRS) refers to the equality constrained case, $= s^2$. Numerical tests are provided for the inequality case, $\leq s^2$. We include theoretical details for the inequality case when they are substantially different.)

It is well known that a vector x yields the global minimum of (TRS) if and only if there exists $\lambda \in \Re$ such that the following relations hold:

$$(A-\lambda I)x = a; \quad \text{(stationarity)} \tag{1.3}$$

$$x^tx = s^2; \quad \text{(feasibility)} \tag{1.4}$$

$$A - \lambda I \succeq 0; \quad \text{(strengthened second order)} \tag{1.5}$$

where $\succeq 0$ denotes positive semidefiniteness. Moreover, if $A - \lambda I$ is positive definite, then the optimizer x is unique. More recently, it has been shown that strong Lagrangian duality holds, see Theorem 2.1.

Currently, most methods for solving (TRS) are based on applying Newton's method to the *secular equation* in λ, which is essentially (1.4) after eliminating x using (1.3). The Newton method is *safeguarded* to maintain positive definiteness in order to satisfy (1.5). Each iteration usually requires Cholesky factorizations of $A - \lambda I$, both for solving (1.3) and for safeguarding (1.5). This can be too expensive for large scale optimization if sparsity is lost.

The general (TRS) lies somewhere between the pure quadratic (eigenvalue case, $a = 0$) and the pure linear (norm, $A = 0$) case. Both of these cases can be solved quickly and easily. Define

$$g(s) := \min\{x^t A x - 2a^t x : x^t x = s^2\}. \tag{1.6}$$

Thus g describes the optimal solution of problem (1.1), (1.2) depending on the norm s of x. Further define

$$G(\sigma) := \min\{y^t A y - 2\sigma a^t y : y^t y = 1\}. \tag{1.7}$$

Setting $x := sy$ we have $\|x\| = s \iff \|y\| = 1$. Therefore, for $s > 0$, we get $g(s) = s^2 G(\frac{1}{s})$. It is clear that $G(\sigma)$ and therefore $g(s)$ describe, in limiting behaviour, the purely quadratic problem ($\sigma \to 0$) as well as the purely linear objective function ($\sigma \to \infty$). We will show below that $g(s)$ is closely related to a parametric (or perturbed) eigenvalue problem.

1.1 Background

Trust region subproblems appear in the context of nonlinear least squares in work by Levenberg 1944, [20] and in work by Marquadt 1963, [24]. These authors worked on the case where A is positive definite. Applications to general minimization appears in work by Goldfeld, Quandt and Trotter 1966 [11]. Early theoretical results on (TRS) appear in Forsythe and Golub 1965 [8]. In particular, they study properties of the secular function, which is essential in algorithmic derivations. Hebden 1973 [14] proposed an algorithm which exploits the structure of the secular function. He made use of earlier work on the structure by Reinsch 1967 and 1971 [35, 36]. Gay 1981 [10] improved on this algorithm and handled the hard case, i.e. the case where the optimal Lagrange multiplier is equal to the smallest eigenvalue of A.

Other algorithms at this time were proposed in [26, 40]. A more efficient treatment of the hard case was the central point of the seminal work by Moré and Sorensen in 1983, [29]. Their algorithm has remained as the standard for (TRS). It typically yields an approximate optimal solution in under ten iterations of a Newton type method. In particular, the algorithm is particularly efficient in the hard case and typically takes only 2-3 iterations, see also [27]. Other algorithms are presented in e.g. [9, 12] and more recently, using DC (difference of convex functions) optimization, in [46].

The (TRS) has appeared elsewhere in the literature under different guises. It is equivalent to the problem of ridge regression in estimation problems, e.g. [16]; and it is also equivalent to the problem of regularization for ill-posed problems, e.g. [47].

Recently, there has been a revival of interest with new duality results, relations to eigenvalue perturbations, and extensions to nonconvex constraint functions, see e.g. [7, 42, 43, 28, 3, 2]. The (TRS) problem has been shown to be solvable in polynomial time, see [52, 18]. The polynomiality is derived using detailed estimates in [49]. In [43], it is shown that strong duality holds for the Lagrangian dual of (TRS); thus, (TRS) is equivalent to a concave maximization problem and, therefore, it is a tractable polynomial time problem, by the results for general convex programs presented in [30]. Applications to solving NP-hard problems are given in [13, 53]. A recent study of the subgradients and stability of $g(s)$ is presented in [44]. A surprising result that there is at most one local-nonglobal optimum for (TRS) is presented in [25].

This paper uses a parametric eigenvalue problem to solve (TRS). Previous characterizations of solutions of (TRS) as a parametric eigenvalue problem appear in [43, Theorem 3.2]. In addition, independent work on a parametric eigenvalue approach similar to our work is presented by Sorensen [41], and is continued by Santos and Sorensen [38]. (Comparisons with our work is given in the concluding Section 5.1.)

Also, recently, there has been a lot of interest in problems with multiple quadratic constraints. The two trust region problem arose in sequential quadratic programming techniques for constrained nonlinear minimization, [5]. Other applications to constrained optimization appear in e.g. [6]. Multiple quadratic constraints arise in combinatorial optimization and their Lagrangian relaxations can be shown to be equivalent to semidefinite relaxations, e.g. [39, 33, 34, 32, 19]. In [32], (TRS) is the tool that is used to prove the equivalence of several bounds for quadratic 0,1 optimization. These problems and relaxations also appear in systems control e.g. [4].

Thus, (TRS) can be seen to be an important stepping stone between convex programs on one hand, which yield necessary and sufficient optimality conditions, and NP-hard problems on the other hand, such as multiple quadratic constrained problems and, equivalently, hard combinatorial optimization problems. Moreover, the primal-dual pair of SDP programs we study illustrate many of the important geometric and algebraic properties of general SDP programs. This pair of SDP programs provide the optimal solution for (TRS); this is in contrast to general SDP programs which usually arise as relaxations and provide bounds for the underlying original problem.

1.2 Outline

The main result in this paper is an algorithm that solves (TRS) using matrix-vector multiplication and no explicit solution of a system of equations, see Section 4. (The reader who is only interested in the algorithm can skip directly to this section.) The algorithm is based on maximizing (unconstrained) a real valued concave function, $k(t)$, based on a parametric eigenvalue problem, i.e.

$$k(t) = (s^2+1)\lambda_1(D(t)) - t, \text{ where } D(t) = \begin{bmatrix} t & -a^t \\ -a & A \end{bmatrix}.$$

We also provide a general framework for (TRS) based on two primal-dual pairs of SDPs. These SDP frameworks can be used to derive various algorithms. In addition, the SDPs are of interest in themselves since they illustrate many interesting properties of general semidefinite programming.

The paper is organized as follows. We begin with a nonlinear primal-dual pair of SDPs that solve (TRS). This pair can be used to describe the steps of the state of the art Moré and Sorensen algorithm, see Remark 2.4. We then present a linear SDP primal-dual pair. We show that the steps from before can be done here without Cholesky factorizations; thus we can exploit structure and sparsity. Both pairs of SDPs are equivalent to (TRS). The primal-dual SDPs with linear constraints provide the framework for our algorithm and are essential for dealing with the hard case.

Section 3 provides detailed analysis of the functions that arise in the above SDP primal-dual pairs. A parametric eigenvalue problem, equivalent to (TRS), is derived in Section 3.1. Theorem 3.7 provides the relationships that form the basis of our algorithm. Section 3.2 discusses the hard case inside the SDP framework; while Section 3.3 describes the various functions associated with (TRS).

Section 4 provides the detailed outline of the algorithm. The algorithm is based on the parametric eigenvalue problem but also uses the special structure that arises from the SDP programs. Results of computational tests are presented in Section 5. Test problems were chosen randomly but taken from several classes in order to exhibit all the different possible cases that can occur for (TRS). The algorithm never failed to find an optimum and the average number of iterations was approximately 4.5. The main work in each iteration is one application of the Lanczos algorithm. However, after the first iteration, subsequent iterations involve applying Lanczos after perturbing the first diagonal element of a matrix. Therefore the Lanczos algorithm finds the minimum eigenvalue very quickly using the eigenvector from the previous iteration. For large dimensions, the amount of work for the whole algorithm was 1.8 times that of the first iteration. We solved tens of thousands of problems with dimensions varying from 100 to 2000.

2 Duality and Semidefinite Programming

Semidefinite programming, SDP, is an extension of linear programming, where the nonnegativity (elementwise) is replaced by a positive semidefiniteness condition, denoted $X \succeq 0$. SDP has recently appeared in many applications, see e.g. [22, 23, 48, 1, 32]. In particular, a hidden semidefinite condition arises in many quadratic programming problems, since a quadratic function is bounded only if the Hessian is positive semidefinite. The (TRS) has a quadratic objective and a quadratic constraint; we shall see that SDP is the hidden key to deriving algorithms for this problem.

We now present two programs which are dual to (TRS) and which exhibit strong duality, see (2.1) and (2.7). Moreover, these programs maximize a concave function and so they show that (TRS) is, implicitly, a convex minimization problem, and so is a tractable polynomial time problem, by the work in [30]. Each of these dual programs leads to a pair of primal-dual SDPs. The first pair of primal-dual SDP's can be used to derive the algorithm for (TRS) in [29]; while the second pair is used to derive our algorithm.

2.1 A Nonlinear Primal-Dual Pair of SDPs

First recall that

$$L(x,\lambda) := x^tAx - 2a^tx - \lambda(||x||^2 - s^2)$$

denotes the *Lagrangian* of (TRS). In [43] it is shown that there is no duality gap for Lagrangian duality for (TRS).

Theorem 2.1 *(i) Strong duality holds for (TRS), i.e.*

$$\mu^* = \min_x \max_\lambda L(x,\lambda) = \max_\lambda \min_x L(x,\lambda).$$

Moreover, attainment holds for x and uniquely for λ.
(ii) A dual problem for (TRS), without a duality gap, is

$$(D) \qquad \mu^* = \max_{A-\lambda I \succeq 0} h(\lambda), \tag{2.1}$$

where $h(\lambda) = \lambda s^2 - a^t(A-\lambda I)^\dagger a$, and $\cdot^\dagger$ denotes the Moore-Penrose generalized inverse.

□

A proof is given in [43]. Alternate proofs for strong duality appear in [21, Problem 3] and also in [46].

Example 2.2 *(**Hard Case**) Let*

$$A = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \ a = (1 \ 0)^t, \ s = 1.$$

The constraint in the dual problem is equivalent to $\lambda \leq -1$, which also implies that the optimal λ for the dual is $\lambda = -1$. The optimal value for the dual is $\mu^* = -1.5$. However, $x_\lambda = (A-\lambda I)^\dagger a = (.5 \ 0)^t$. Substituting into the objective function we see that $q(x_\lambda) = -.75$. Moreover, x_λ lies inside the disk and not on the boundary, i.e. complementary slackness fails. This illustrates one of the weaknesses of Lagrangian duality, i.e. the attainment point in the Lagrangian is not necessarily the optimum point for (TRS), even though there is a zero duality gap.

Slater's constraint qualification holds for the dual program (2.1). We can therefore derive its dual program.

Corollary 2.3 *The dual program to (2.1) is*

$$(DD) \qquad \begin{array}{lc} \mu^* = \min & h(\lambda) + \operatorname{trace} X(A-\lambda I) \\ \textit{subject to} & s^2 - ||(A-\lambda I)^\dagger a||^2 - \operatorname{trace} X = 0 \\ & X \succeq 0. \end{array} \tag{2.2}$$

Remark 2.4 *The above primal-dual pair of SDPs can be used to derive the algorithm in [29, equation (3.6)]. Though the function h is not used explicitly, the algorithm actually is trying to solve (D). In fact, it applies Newton's method to a modified form of the stationarity condition $h'(\lambda) = ||x_\lambda|| - s = 0$, i.e. to $1/||x_\lambda|| - 1/s = 0$, where $x_\lambda = (A - \lambda I)^\dagger a$. The latter equation is almost linear which improves the performance of Newton's method. Backtracking is done in order to safeguard the positive semidefiniteness constraint in (D). Note that the duality gap at feasible points, between (D) and (DD), is given by the complementary slackness equation*

$$\textit{duality gap} \ = \operatorname{trace} X(A - \lambda I). \tag{2.3}$$

Now, let $x_\lambda = (A - \lambda I)^\dagger a$ and $||x_\lambda + z||^2 = s^2$. Then, [29, equation (3.6)] in our notation is

$$\begin{aligned} q(x_\lambda + z) &= (\lambda s^2 - x_\lambda^t (A - \lambda I) x_\lambda) + z^t (A - \lambda I) z \\ &= h(\lambda) + z^t (A - \lambda I) z. \end{aligned} \tag{2.4}$$

The first equality is the key in [29] for handling the hard case. For fixed λ, a linpack routine for estimating the smallest singular value of $A - \lambda I$ is used to find a z (a direction of negative curvature) to make $z^t(A - \lambda I)z$ small. In our framework, we set $X = zz^t$. This is equivalent to making the duality gap term, or complementary slackness term, small.

Thus, the algorithm in [29] can be described as applying Newton's method with backtracking to (D); while the hard case step in [29] can be described as: find X to reduce the duality gap between (D) and (DD) while maintaining feasibility in (DD). (Alternatively, backtracking can be avoided by applying an interior-point method directly to (D), see [37].)

Remark 2.5 *The above provides a pair of primal-dual SDP programs where the objective functions are not linear. In the inequality constrained case, $||x|| \leq s$, we must have the Lagrange multiplier nonpositive, $\lambda \leq 0$. This is an added constraint in (D). Correspondingly, the equality constraint in (DD) is changed to ≥ 0.*

2.2 A Linear Primal-Dual Pair of SDPs

The above nonlinear pair of SDPs provides an explanation for current algorithms for (TRS). We now exploit the strong Lagrangian duality in Theorem 2.1, (in particular attainment for y_0 defined below) and show that (TRS) can be reformulated as an unconstrained concave maximization problem. This latter problem can be formulated as a primal-dual linear SDP. The special structure of these formulations are exploited in our algorithm.

Homogenization yields

$$
\begin{aligned}
\mu^* &= \min_{||x||=s,\ y_0^2=1} x^tAx - 2y_0a^tx \\
&= \max_t \min_{||x||=s,\ y_0^2=1} x^tAx - 2y_0a^tx + ty_0^2 - t \\
&\geq \max_t \min_{||x||^2+y_0^2=s^2+1} x^tAx - 2y_0a^tx + ty_0^2 - t \\
&\geq \max_{t,\lambda} \min_{x,y_0} x^tAx - 2y_0a^tx + ty_0^2 - t + \lambda(||x||^2 + y_0^2 - s^2 - 1) \\
&= \max_{r,\lambda} \min_{x,y_0} x^tAx - 2y_0a^tx + ry_0^2 - r + \lambda(||x||^2 - s^2) \\
&= \max_{\lambda} \left(\max_r \min_{x,y_0} x^tAx - 2y_0a^tx + ry_0^2 - r + \lambda(||x||^2 - s^2) \right) \\
&= \max_{\lambda} \min_{x,y_0^2=1} x^tAx - 2y_0a^tx + \lambda(||x||^2 - s^2) \\
&= \mu^*,
\end{aligned}
$$

where we have equated r with $t+\lambda$. We have used Theorem 2.1 for the last two equalities. The final equality also uses the symmetry of the function.

Define

$$k(t) = (s^2+1)\lambda_1(D(t)) - t, \tag{2.5}$$

where, for $t \in \Re$, the symmetric $(n+1)\times(n+1)$ matrix

$$D(t) = \begin{bmatrix} t & -a^t \\ -a & A \end{bmatrix}, \tag{2.6}$$

and λ_1 denotes the smallest eigenvalue. Then the third expression in the above chain implies that an unconstrained dual problem to (TRS) is

$$\max_t k(t). \tag{2.7}$$

Thus, the nonconvex constrained problem (TRS) is transformed to an unconstrained maximization of a concave function from $\Re$ to $\Re$. Note, in the case that $\lambda_1(D(t))$ is simple, the derivative satisfies

$$k'(t) = (s^2+1)y_0^2 - 1, \tag{2.8}$$

where $\begin{pmatrix} y_0 \\ x \end{pmatrix}$ is the normalized eigenvector for $\lambda_1(D(t))$, i.e. a stationary point $k'(t) = 0$ is equivalent to the feasibility condition for (TRS) for sx, i.e.

$s^2 - ||sx||^2 = 0$. In fact, we shall see that this still holds for *some* normalized eigenvector in the case that $\lambda_1(D(t))$ is not simple. (As mentioned above, in the case of an inequality constraint, $||x|| \leq s$, we have to include the extra constraint $\lambda_1(D(t)) \leq 0$.)

In fact, this dual problem is a max-min eigenvalue problem with very simple structure. (We can include $-t$ in the eigenvalue expression by subtracting $\frac{t}{s^2+1}I$ from $D(t)$.) Therefore, (TRS) can be solved by known max-min eigenvalue techniques, see e.g. [31, 15]. These techniques are known to have asymptotic quadratic convergence. However, these techniques are for general parameter problems and do not exploit the very special structure of our problem. Moreover, we need fast global convergence rather than fast asymptotic convergence.

In addition, as mentioned in Section 1.2, the fact that (TRS) can be transformed into an unconstrained concave maximization problem implies that it can be solved in polynomial time.

We can change the dual problem (2.7) into a SDP by adding the variable λ.

$$(DSDP) \qquad \begin{array}{rl} \mu^* = \max & (s^2+1)\lambda - t \\ \text{subject to} & D(t) \succeq \lambda I. \end{array}$$

(The constraint $\lambda \leq 0$ is included in the $||x|| \leq s$ case.) This is still the dual to (TRS); there is no duality gap and strong duality holds. Each feasible dual solution provides a lower bound on the optimal value μ^*. Therefore, given any value t and a lower bound to the smallest eigenvalue, we conclude that

$$\lambda_1(D(t)) \geq \lambda \Longrightarrow \mu^* \geq (s^2+1)\lambda - t. \tag{2.9}$$

Moreover, Slater's constraint qualification holds for (DSDP) so that we can take the Lagrangian dual of this dual to get a semidefinite equivalent for (TRS), see e.g. [51].

$$(PSDP) \qquad \begin{array}{rc} \mu^* = \min & \operatorname{trace} D(0)X \\ \text{subject to} & \operatorname{trace} X = s^2+1 \\ & X_{11} = 1 \\ & X \succeq 0. \end{array}$$

(In the inequality constrained case, the additional constraint $\lambda \leq 0$ in (DSDP) corresponds to $\operatorname{trace} X \leq s^2 + 1$ in (PSDP). This further emphasizes the fact that the dual variable λ is the shadow price corresponding to the trace constraint or norm constraint; while t is the shadow price for the constraint on X_{11}.) This SDP can be obtained directly as an SDP relaxation of (TRS) if we consider X in the form

$$X = \begin{pmatrix} y_0 \\ x \end{pmatrix} \begin{pmatrix} y_0 & x^t \end{pmatrix} = \begin{bmatrix} y_0^2 & y_0 x^t \\ y_0 x & x x^t \end{bmatrix}, \text{ with } y_0^2 = 1, \tag{2.10}$$

where x is feasible for (TRS). This program provides us with the means of obtaining upper bounds for (TRS). Moreover, the dual pair of SDPs yields the duality gap at feasible solutions $(\lambda, t), X$, i.e.

$$\text{duality gap} = \operatorname{trace} D(0)X - (s^2+1)\lambda + t = \operatorname{trace}\,(D(t) - \lambda I)X, \tag{2.11}$$

i.e. the value of the complementary slackness relation. In the case of the inequality constraint in (TRS), we need to add $\lambda\,(\operatorname{trace} X - (s^2+1))$ to the duality gap. So we get $\operatorname{trace} D(t)X - (s^2+1)\lambda$. The relationship between optimal solutions of the SDP programs and (TRS) is as follows.

Theorem 2.6 *Suppose that* (λ^*, t^*), *and* $X^* = \begin{bmatrix} 1 & y^{*t} \\ y^* & \bar{X}^* \end{bmatrix}$ *are optimal for the primal-dual pair (DSDP) and (PSDP), respectively. Then:*

1. $\mu^* = \max_t k(t) = k(t^*) = \max_{A-\lambda I \succeq 0} h(\lambda) = h(\lambda^*) = \lambda^* s^2 - a^t y^*$, *and* $\|y^*\| \le s$.

2. *Let the matrix* X^* *be factored as* $X^* = TT^t$, *where* $T = \begin{bmatrix} w^t \\ \bar{T} \end{bmatrix}$ *is* $(n+1) \times r$ *and full column rank. Then* $y^* = \bar{T}w$ *and, for every* $0 \neq v \in \Re^r$ *such that* $Tv = \begin{pmatrix} 0 \\ \eta_v \end{pmatrix}$ *we have that* $\eta_v \in \mathcal{N}(A - \lambda^* I)$ *and the vectors* $x^* = y^* \pm \alpha \eta_v$, *where*

$$\alpha = \frac{s^2 - \|y^*\|^2}{y^{*t}\eta_v + sign(y^{*t}\eta_v)\sqrt{(y^{*t}\eta_v)^2 + (s^2 - \|y^*\|^2)}}$$

   solve (TRS), with unique optimal Lagrange multiplier $\lambda^* = \lambda_1(D(t^*))$. *Moreover,* $X^* = \begin{pmatrix} 1 \\ x^* \end{pmatrix} (1\ x^*)$ *is optimal for (PSDP) and* $\begin{pmatrix} 1 \\ x^* \end{pmatrix}$ *is an eigenvector for* $\lambda^* = \lambda_1(D(t^*))$.

Proof. First note that the optimal value of each of the pair of dual SDP programs equals the optimal value of (TRS) by construction and strong duality. Feasibility of X^* for (PSDP) implies that $||y^*|| \leq s$. Moreover, the relationships with the functions k and h have already been shown. Now let $Z^* = D(t^*) - \lambda^* I$. Then, complementary slackness, trace $Z^*X^* = 0$, implies

$$0 = Z^*X^* = \begin{bmatrix} t^* - \lambda^* & -a^t \\ -a & A - \lambda^* I \end{bmatrix} \begin{bmatrix} 1 & y^{*t} \\ y^* & \bar{X}^* \end{bmatrix}.$$

The complementary slackness and feasibility further yield:

$$t^* = a^t y^* + \lambda^*; \tag{2.12}$$

$$\bar{X}^* a = (t^* - \lambda^*) y^*; \tag{2.13}$$

$$(A - \lambda^* I) y^* = a; \tag{2.14}$$

$$(A - \lambda^* I) \bar{X}^* = a y^{*t}; \tag{2.15}$$

$$A - \lambda^* I \succeq 0. \tag{2.16}$$

From (2.14) we get stationarity for y^*, while (2.16) gives the second order optimality condition. Therefore, to prove optimality for (TRS) we need only verify feasibility (and complementary slackness in the inequality constrained case).

Now, in the case that the smallest eigenvalue is simple, complementary slackness, $Z^*X^* = 0$, implies that X^* is rank-one; feasibility for (PSDP) yields the factorization similar to (2.10). Therefore optimality for (TRS) follows by noting that feasibility of X in (PSDP) implies feasibility for the solution $x^* = y^*$. The Lagrange multiplier value comes from the eigenvalue-eigenvector equation for t in (DSDP). Uniqueness for λ follows from uniqueness of the Lagrange multiplier in (TRS). (Or from the strict concavity of the function $h(\lambda)$.)

In the case that the smallest eigenvalue is not simple, the above argument still holds except for the simple representation of x^*, since X^* is not necessarily rank-one. We use the full rank factorization of X^*. First note that $0 = \operatorname{trace} Z^*TT^t = \operatorname{trace} T^tZ^*T$ implies that the semidefinite matrix $T^tZ^*T = 0$ and, further, that $Z^*T = 0$. Therefore each column of T is an eigenvector for $\lambda_1 D(t^*)$. The result now follows by noting that y^* is a stationary point and so x^* is also a stationary point and, moreover, the formula for α guarantees that $||x^*|| = s$, see e.g. [29, Page 558].

□

The relationship between the optimal solutions of the SDP pair is given in the following corollary. As in ordinary linear programming, the connection is through the complementary slackness conditions. Possible optimal points for (DSDP) are just extreme points, i.e. points where $\lambda_1 D(t) = \lambda$. However, the possible optimal points for (PSDP) are not as simple to characterize and depend on being able to solve an inverse eigenvalue problem.

Corollary 2.7 *Suppose that (λ^*, t^*), are optimal for (DSDP). Then*

$$\lambda^* = \lambda_1(D(t^*)) \text{ and } \mu^* = (s^2+1)\lambda^* - t^*.$$

Let T be $(n+1) \times r$ and full column rank such that $(D(t^*) - \lambda^* I)T = 0$ and $r \geq 2$ if the multiplicity of the eigenvalue λ^* is ≥ 2. Then $T = \begin{pmatrix} w \\ \bar{T} \end{pmatrix}$ can be normalized so that $||w|| = 1$. If v is chosen as in Theorem 2.6, and $z = T(w+z)$, then $X^* = zz^t$ solves (PSDP).

Conversely, suppose that $X^* = \begin{bmatrix} 1 & y^{*t} \\ y^* & \bar{X} \end{bmatrix}$ solves (PSDP) and $X^* = TT^t$ with T full column rank. Then the linear equations

$$\mu^* = (s^2+1)\lambda^* - t^*, \quad (D(t^*) - \lambda^* I)T = 0,$$

can be solved for t^*, λ^*, which are then the optimal solutions for (DSDP).

□

3 Eigenvalue Functions

In this section we study the various functions involved in the SDP programs derived above. These functions have special structure that we exploit in our algorithm. In particular, we present 6 functions: the first 3 describe the functions for the nonlinear primal-dual SDP pair; while the next 3 describe the corresponding functions for the linear pair.

3.1 Parametric Eigenvalue Problem

We first take a closer look at $\lambda_1(D(t))$, the smallest eigenvalue of $D(t)$, as t varies. At first, the following assumption will be made to guarantee that $\lambda_1(D(t))$ has multiplicity 1 for all $t \in \Re$. For simplicity of notation, we let A be diagonal with diagonal elements α_i. Let i be the multiplicity of $\lambda_1(A)$, i.e.

$$\alpha_1 = \ldots = \alpha_i < \alpha_{i+1} \leq \ldots \leq \alpha_n.$$

$$\textbf{Assumption:}\ \exists j \in \{1,\ldots,i\} : a_j \neq 0. \tag{3.1}$$

Assumption (3.1) can be rephrased for general symmetric A in geometric terms as follows, see e.g. [29].

> Condition (3.1) is violated if and only if $a^t x = 0$ for all x in the eigenspace corresponding to $\lambda_1(A)$.

We will discuss questions related to the case when (3.1) does not hold in the following Section 3.2. If (3.1) holds, we may without loss of generality assume $a_1 \neq 0$. The assumption (3.1) is called *the easy case* in the literature. If the assumption fails, then *the hard case* holds.

An application of Schur's determinant theorem on Schur complements yields the following.

Proposition 3.1 *(see [50])*

$$\det(D(t) - \lambda I) = (t - \lambda) \prod_{k=1}^{n} (\alpha_k - \lambda) - \sum_{k=1}^{n} a_k^2 \prod_{j \neq k}^{n} (\alpha_j - \lambda).$$

□

Let $J := \{i : a_i \neq 0\}$. Note that $1 \in J$. Then we get

$$\det(D(t) - \lambda I) = [t - \lambda - \sum_{j \in J} \frac{a_j^2}{\alpha_j - \lambda}] \prod_{k=1}^{n} (\alpha_k - \lambda). \tag{3.2}$$

For $\lambda \notin \{\alpha_j : j \in J\}$ define

$$d(\lambda) := \lambda + \sum_{j \in J} \frac{a_j^2}{\alpha_j - \lambda}.$$

Note that

$$d(\lambda) = \lambda + a^t (A - \lambda I)^{-1} a, \text{ for } \lambda < \alpha_1.$$

This function is similar to the secular functions used in current divide-and-conquer methods and trust region algorithms.

Lemma 3.2 *If (3.1) holds, then for all $t \in \Re$ we have*

$$\lambda_1(D(t)) \textit{ is simple and } \lambda_1(D(t)) < \alpha_1.$$

Proof. Since the eigenvalues of A interlace those of $D(t)$ for all $t \in \Re$, see e.g. [17, Pg 185], we trivially have $\lambda_1(D(t)) \leq \alpha_1$. By (3.2) we have to show that $t - d(\lambda) = 0$ has a unique zero $\lambda^* < \alpha_1$ for all $t \in \Re$. Now (3.1) implies that $d(\lambda)$ has a pole at $\lambda = \alpha_1$ and

$$\lim_{\lambda \to \alpha_1, \lambda < \alpha_1} d(\lambda) = +\infty.$$

Moreover

$$\lim_{\lambda \to -\infty} d(\lambda) = -\infty.$$

Since $d(\lambda)$ is continuous for $\lambda < \alpha_1$, we see that $d(\lambda) = t$ has at least one solution $\lambda < \alpha_1$ for all t. By the interlacing argument there can be at most one solution $\lambda < \alpha_1$. (Note also that d is strictly monotonically increasing and convex on $(-\infty, \alpha_1)$.)

$\square$

Corollary 3.3 *Suppose that $\lambda_1(D(t))$ is not simple. Then the hard case holds for (TRS).*

$\square$

For a given value t, we have seen how to move to the boundary of the feasible set of (DSDP), i.e. we use $\lambda = \lambda_1(D(t))$. The following shows how to move to the boundary given a value for λ.

Corollary 3.4 *Suppose that $\lambda < \alpha_1$ is given. Then*

$$D(d(\lambda)) - \lambda I \succeq 0 \text{ and singular.}$$

$\square$

We will now investigate the eigenvector corresponding to $\lambda_1(D(t))$. Recall that the eigenvector can be used to obtain a feasible solution for (PSDP), see Theorem 2.6.

Lemma 3.5 *Suppose (3.1) holds and let $t \in \Re$ be fixed. Let $y(t)$ be a normalized eigenvector corresponding to $\lambda_1(D(t))$ and denote its first component by $y_0(t)$. Then $y_0(t) \neq 0$.*

Proof. Suppose $y_0(t) = 0$. Then $\lambda_1(D(t))$ is an eigenvalue of A, which contradicts Lemma 3.2.

$\square$

In view of Lemma 3.5 we may further normalize the eigenvector $y(t)$ so that it is norm 1 and $y_0(t) > 0$ $\forall t$.

Lemma 3.6 *If (3.1) holds and the eigenvector $y(t)$ is normalized, then $y_0(t) : \Re \mapsto (0,1)$ is strictly monotonically decreasing.*

Proof. First note that for $t \to -\infty$ we have $\lambda_1(D(t)) - t \to 0$ and thus $y(t) \to e_1$. (Throughout e_i denotes the ith canonical unit vector in the appropriately dimensioned space.) On the other hand if $t \to +\infty$ we have $\lambda_1(D(t)) \uparrow \alpha_1$ and therefore $y_0(t) \downarrow 0^+$. These are immediate consequences of the properties of $d(\lambda)$. We show now that $y_0(t)$ is monotonically decreasing. Let $t = t_0$ be fixed and suppose $y(t_0), u_2(t_0), \ldots, u_{n+1}(t_0)$ is an orthonormal basis of eigenvectors of $D(t_0)$. Since $\lambda_1(D(t_0))$ is simple, as shown in Lemma 3.2, we get, see e.g. Theorem 4.2, Chapter 6 from [45]:

$$\begin{array}{rcl} y(t) & = & y(t_0) \quad + \sum_{j>1} \frac{t-t_0}{\lambda_1(t_0)-\lambda_j(t_0)} u_j(t_0) u_j^t(t_0) e_1 e_1^t y(t_0) \\ & & \quad\quad +\text{terms of higher order in } t - t_0, \end{array} \tag{3.3}$$

for t sufficiently close to t_0. Therefore we get

$$y_0(t) = y_0(t_0) + (t - t_0) \sum_{j>1} \frac{(e_1^t u_j(t_0))^2}{\lambda_1(t_0) - \lambda_j(t_0)} y_0(t_0) + o(t - t_0).$$

Now $y_0(t_0) > 0$, $\lambda_1(t_0) - \lambda_j(t_0) < 0$, thus for $t > t_0$ but close to t_0 we have $y_0(t) < y_0(t_0)$.

□

We now formulate and prove the main result of this section, relating $\lambda_1(D(t))$ to the function $g(s)$, defined by (1.6).

Theorem 3.7 *Let A and a be as above. Let $t \in \Re$ and suppose $y(t) := (y_0(t), z(t)^t)^t$ is a normalized eigenvector of $D(t)$ corresponding to $\lambda_1(D(t))$. Suppose (3.1) holds. Then $y_0(t) \neq 0$ and*

$$v := \frac{1}{y_0(t)} z(t)$$

is the unique optimal solution of

$$\min\{v^t A v - 2a^t v : v^t v = \frac{1 - y_0(t)^2}{y_0(t)^2}\}. \tag{3.4}$$

Conversely suppose $v \in \Re^n$ and $\lambda \in \Re$ satisfy

$$(A - \lambda I)v = a; \quad A - \lambda I \textit{ positive definite and } v^t v = \frac{1 - y_0^2}{y_0^2},$$

thereby defining $y_0 > 0$. Then $y := y_0(1, v^t)^t$ is an eigenvector of $D(t)$ for $t := a^t v + \lambda$ and $\lambda_1(D(t)) = \lambda$.

Proof. Fix $t \in \Re$. Suppose

$$[D(t) - \lambda_1 I]y = 0$$

and

$$||y|| = 1.$$

Expanding we get

$$(t - \lambda_1)y_0 - a^t z = 0, \tag{3.5}$$

$$-y_0 a + Az - \lambda_1 z = 0,$$

$$y_0^2 + z^t z = 1.$$

By Lemma 3.5, we may assume $y_0 > 0$. Set $v := \frac{1}{y_0} z$. Then

$$Av - \lambda_1 v = a, \tag{3.6}$$

$$v^t v = \frac{1 - y_0^2}{y_0^2},$$

$$A - \lambda_1 I \text{ is positive definite.}$$

The last relation follows from Lemma 3.2. Thus v is the unique minimizer of (3.4). To see the other direction, suppose

$$Av - \lambda v = a,$$

$$v^t v = \frac{1 - y_0^2}{y_0^2},$$

and $A - \lambda I$ positive definite. Setting $y := (y_0, y_0 v^t)^t$ and $t := a^t v + \lambda$ we get

$$(D(t) - \lambda I)y = 0.$$

Therefore y is an eigenvector of $D(t)$ corresponding to λ. Note also that $y^t y = y_0^2(1 + v^t v) = 1$. Since $A - \lambda I$ is positive definite, we get $\lambda < \lambda_1(A)$, therefore $\lambda = \lambda_1(D(t))$.

□

It is interesting to view Theorem 3.7 using the SDP framework and our dual pair of SDP programs, see Theorem 2.6. Since we have assumed the easy case, we see that normalizing the eigenvector, $||y||^2 = s^2 + 1$, corresponds to satisfying the trace constraint in (PSDP). If we set $z = \lambda_1(D(t))$, then (z, t), and $X = y(t)y(t)^t$ solve the perturbed dual pair of

SDP programs where (PSDP) has the perturbed constraint $X_{11} = y_0(t)^2$, and (DSDP) has the appropriate perturbed objective function.

Theorem 3.7 can be used to solve the minimization problem (1.1), (1.2) using the dual simplex scenario. Suppose A, a and s are given. We have to find $t \in \Re$ such that

$$\tau(t) := \sqrt{\frac{1 - y_0^2(t)}{y_0^2(t)}} = s \text{ or } y_0^2 = \frac{1}{s^2 + 1},$$

where $y_0(t)$ denotes the first component of a normalized eigenvector $y(t) = (y_0(t), z(t)^t)$ corresponding to $\lambda_1(D(t))$. Then we know from the theorem that

$$x := \frac{1}{y_0(t)} z(t)$$

is the unique solution of our minimization problem with corresponding Lagrange multiplier $\lambda_1(D(t))$. Note that due to the monotonicity of $y_0^2(t)$, the function $\tau(t)$ is also strictly monotone. Therefore the correct value of t can be approximated by standard search procedures. We discuss computational issues like finding an interval $[t^l, t^u]$ that contains the desired value t in a subsequent section. The main point here is to note that the present theory solves (1.1), (1.2) by successive calculations of the smallest eigenvalue of $D(t)$. No factorization or solution of a linear system is required, and the possible sparsity of A can be fully exploited. This is because the norm of the current approximate solution is calculated using the relation with y_0 in (3.4), rather than directly. Moreover, no explicit safeguarding for positive semidefiniteness of the Hessian $A - \lambda I$ is needed.

3.2 The Hard Case

We now discuss the case where condition (3.1) does not hold. Several authors call this case the hard case, because numerical problems can be expected in this situation. We will show now that this case can also be handled by our approach. In fact, the hard case corresponds to multiplicity greater than one for the smallest eigenvalue of $D(t^*)$. Therefore, the only added complexity comes from the fact that the Lanczos algorithm is slow when the smallest eigenvalue is not simple. We get an additional speedup in the hard case by using a primal simplex step.

Suppose a is orthogonal to the eigenspace corresponding to α_1. After having chosen the coordinate system using the orthonormal eigenvectors of

A, we get that A is diagonal and

$$a_1 = \ldots = a_i = 0.$$

To avoid trivialities, we also assume $a \neq 0$. From our discussion in Section 1 it is clear that if $s > 0$ is sufficiently small, then the optimal solution to the trust region problem is still unique, because in this case the quadratic part of the objective function is substantially smaller than the linear term. Therefore an optimal x will be "almost" parallel to a. As s gets larger, uniqueness of the optimal solution may be lost. The key to understanding the behaviour of the optimal solution lies again in the smallest eigenvalue $\lambda_1(D(t))$ and the corresponding eigenspace. We recall the definition of $d(\lambda)$.

$$d(\lambda) = \lambda + \sum_{j \in J} \frac{a_j^2}{\alpha_j - \lambda} \quad \forall \lambda \neq \alpha_j : j \in J. \tag{3.7}$$

Note that d does not have a pole at α_1. Let

$$t_0 := d(\alpha_1).$$

We denote the multiplicity of the eigenvalue α_1 of A by i. Clearly, $i \geq 1$.

Lemma 3.8 $\lambda_1(D(t_0)) = \alpha_1$ *with multiplicity* $i + 1$. *Moreover, there exists an eigenvector* y *with first component different from 0.*

Proof. To show that $\lambda_1(D(t_0)) = \alpha_1$ we first note that $d'(\lambda) \geq 1$. Therefore $\lambda < \alpha_1$ implies $d(\lambda) < d(\alpha_1)$. Suppose $\lambda < \alpha_1$. Then the characteristic polynomial, see (3.2), satisfies

$$\det(D(t_0) - \lambda I) = [d(\alpha_1) - d(\lambda)] \prod_{k=1}^{n} (\alpha_k - \lambda) > 0,$$

because each factor on the right hand side is positive. Therefore, by interlacing, $\lambda_1(D(t_0)) = \alpha_1$. The multiplicity of α_1 is at least $i+1$, by the definition of t_0. Since A is a principal submatrix of $D(t_0)$, and the multiplicity of α_1 in A is precisely i, it follows from the interlacing theorem that the multiplicity of α_1 in $D(t_0)$ can be at most i. This proves the first part of the lemma.

To see the second part, we note that there are i pairwise orthogonal eigenvectors $e_2, \ldots, e_{i+1}$ for α_1. Thus there exists another eigenvector y in their orthogonal complement, so

$$y = (y_0, 0, \ldots, 0, y_{i+1}, \ldots, y_n)^t.$$

Now the assumption $y_0 = 0$ implies that the columns of $D(t_0)$ corresponding to $y_{i+1}, \ldots, y_n$ are linearly dependent, contradicting the assumption that $\alpha_i < \alpha_{i+1} \leq \ldots \alpha_n$. The proof also shows that if we assume $\|y\| = 1$, then y_0^2 is uniquely determined.

$\square$

We now distinguish between two cases for $y_0 = y_0(t_0)$.

Case 1. $(1 - y_0^2)/y_0^2 > s^2$.
In this case we can proceed similarly as in the previous section, because the following observation shows that $t^* < t_0$.

Lemma 3.9 *If $t < t_0$ then $\lambda_1(D(t))$ is simple and less than α_1. The corresponding eigenvectors have first component nonzero. Moreover, there exists some $t < t_0$ such that*

$$\frac{1 - y_0(t)^2}{y_0(t)^2} = s^2.$$

$\square$

We omit the proof, because the argument is essentially the same as in the previous section.

Case 2. $(1 - y_0^2)/y_0^2 \leq s^2$.
In this case let $y = (y_0, z^t)^t$ denote the normalized eigenvector for $D(t_0)$ having $y_0 \neq 0$. Define $u := \frac{1}{y_0} z$ and select v from the eigenspace of α_1 from A of norm 1. By construction, $v \perp u$. Now it is a simple matter to verify that

$$x := u + \sqrt{(s^2 - \frac{1 - y_o^2}{y_0^2})}\, v$$

satisfies the optimality conditions with $\lambda = \alpha_1$. Thus $t^* = t_0$.

In our algorithm we treat the hard case, and the "almost" hard case, similar to the approach in [29], i.e. we find a vector which allows us to move to the correct radius while improving the objective function. In [29], this is done using the Cholesky factorization and a linpack routine which estimates the smallest singular value and vector, see Remark 2.4. We proceed using the framework of our dual pair of SDP programs. As seen above, the possibility of the hard case is indicated by $(1 - y_0^2)/y_0^2 < s^2$, or equivalently, by

$$y_0^2 > 1/(1 + s^2). \tag{3.8}$$

Now if (3.8) holds, then the matrix

$$X = \frac{1}{y_0^2} \begin{pmatrix} y_0 \\ x \end{pmatrix} (y_0 \; x) \tag{3.9}$$

is feasible for (PSDP) with $\operatorname{trace} X < 1 + s^2$. We can now check the duality gap using the feasible pair $(z,t), X$. If we are within our tolerances for the norm of x and for the duality gap, then we stop.

3.3 Six Useful Functions

In the above semidefinite framework, we can treat the pair of SDP programs as linear programs and try and apply techniques from linear programming. There were also several other functions that arose from this framework. These can be divided into two groups. We now list these functions and their properties. The first group of three functions form the basis for the current algorithms for (TRS) and require the Cholesky factorization to find derivatives for Newton's method and for safeguarding positive definiteness. (See Remark 2.4.) The corresponding functions for our algorithm appear as the second group of three functions.

1. **The Quadratic Dual**
   The following three functions arise from the quadratic dual pair (D) and (DD). The properties we present are described in [43].

   (a) $h(\lambda) = \lambda s^2 - a^t (A - \lambda I)^\dagger a$
   Recall that $\mu^* = \max_{A - \lambda I \succeq 0} h(\lambda)$. This function is strictly concave on the open interval $(-\infty, \lambda_1(A))$ and it diverges to $-\infty$ as λ decreases to $-\infty$. In the easy case, it also diverges to $-\infty$ as λ increases to $\lambda_1(A)$. Therefore the maximum occurs in the open interval $(\lambda_1(A), -\infty)$.
   However, in the hard case, the maximum of $h(\lambda)$ may be at the boundary point $\lambda_1(A)$.

   (b) $h'(\lambda) = s^2 - a^t ((A - \lambda I)^\dagger)^2 a$
   The derivative of h is equivalent to the feasibility condition of (TRS), Solving $h'(\lambda) = 0$, subject to the semidefinite condition, is clearly equivalent to solving (TRS). This function is concave on the same region that h is.
   In the hard case, the derivative can be > 0 at the optimum. Moreover, the (one-sided) derivative formula does not necessarily

hold with the Moore-Penrose generalized inverse, but may require some other generalized inverse.

(c) $\phi(\lambda) = \frac{1}{s} - \frac{1}{\|(A-\lambda I)^\dagger a\|}$

We can solve the equivalent, square rooted, reciprocal equation, subject to the semidefinite condition. The function ϕ is almost linear and is convex on the region that h is concave. This is the function currently used to get fast algorithms for (TRS).

In the hard case, Newton's method predicts points that can lie on the wrong side of the smallest eigenvalue of A.

2. **The SDP Dual**

The following three functions arise from the dual pair (PSDP) and (DSDP). The properties we present can be derived from the relations with the above three functions.

(a)
$$k(t) = (s^2+1)\lambda_1(D(t)) - t \tag{3.10}$$

Recall that $\mu^* = \max_t k(t)$. This function is strictly concave in the easy case, since the maximum eigenvalue is a concave function. It is coercive since it diverges to $-\infty$ as $|t|$ goes to ∞.

In the hard case, the function is not differentiable at points where the multiplicity of the eigenvalue changes. It is linear and equal to a constant plus $(-t)$ when t becomes sufficiently large. The linearity follows from interlacing, i.e. when $D(t)$ has a multiple smallest eigenvalue, then interlacing implies that the eigenvalue becomes and stays equal to the smallest eigenvalue of A, for larger values of t.

(b)
$$k'(t) = (s^2+1)y_0^2(t) - 1 \tag{3.11}$$

The derivative of k is equivalent to the feasibility condition of (TRS) under a normalization condition, where y_0 is the normalized first component of the eigenvector for $\lambda_1(D(t))$.

The function is convex, nonincreasing, and can have a jump discontinuity at the point where the hard case is detected, i.e. when y_0 becomes 0.

(c)
$$\psi(t) := \sqrt{s^2+1} - \frac{1}{y_0(t)} \tag{3.12}$$

We can solve the equivalent, square rooted, reciprocal equation. There are advantages for solving the equation $\psi(t) = 0$. This equation is almost linear, nonincreasing, and concave.
In the hard case, this function is not defined when $y_0 = 0$. Therefore, it may not provide useful information in the hard case, for $t > t^*$.

4 The Dual Simplex Algorithm

In this section we describe our algorithm to solve the *inequality constrained* (TRS). We have shown (see Remark 2.4) that the nonlinear primal-dual SDP pair can be used to describe current algorithms, e.g. [29], for (TRS). Our algorithm is based on replacing the nonlinear primal-dual SDP pair by the linear primal-dual pair, (DSDP) and (PSDP), and taking similar steps, though without backtracking. In addition, it is based on solving $\max_t k(t)$ (see (3.10)) or $k'(t) = 0$ (see (3.11)), for $t \in \Re$. We exploit the structure of $\psi(t)$ (see (3.12)) and $k(t)$. The main computational step is the eigenvalue and eigenvector computation for $D(t)$ (see (2.6)) for various values of t. We assume that a subroutine is available, that calculates with 'sufficient accuracy' the smallest eigenvalue along with a normalized eigenvector of a symmetric matrix. We do not assume any knowledge about whether the hard case occurs or not. Our algorithm is set up to handle this situation automatically. In fact, the algorithm takes advantage of the structure of the hard case, or near hard case, and usually takes fewer iterations.

The method we are going to describe parallels that of the dual simplex algorithm for linear programming. Therefore we call this a dual simplex method. We first outline the algorithm with notation familiar from linear programming.

1. Start with the variable t and evaluate the basic variable (extreme point) (t, λ) of the (dual) feasible set; we use the smallest eigenvalue $\lambda = \lambda_1(D(t))$.

2. Using complementary slackness, find an approximate solution to the primal problem (PSDP); we use a normalized eigenvector y for λ found above, and $X = (s^2 + 1)yy^t$. Therefore the first constraint in (PSDP), the trace constraint, is satisfied.

3. With the previous values from X_{11} (equivalently from y_0), we use various forms of inverse interpolation to predict a value for t such that

the second constraint in (PSDP), $X_{11} = 1$ (equivalently $\psi(t) = 0$), is closer to being satisfied and/or the duality gap is decreased. (Note that this means we are using the shadow price interpretation of t.) An important ingredient of this is taking a primal step, i.e. moving to the boundary of the primal feasible set and reducing the duality gap.

We now describe in detail the main steps of the algorithm.

4.1 Initialization

We first derive intervals that contain the optimal value μ^* and the optimal parameter value t^*.

Lemma 4.1 *Let A, a and s be given. Let t^*, λ^* be optimal for (DSDP). Then*

$$\lambda_1(A) - \frac{1}{s}\|a\| \leq t^* \leq \lambda_1(A) + s\|a\|. \tag{4.1}$$

Proof. From (2.12),

$$t^* - \lambda^* = a^t y^*.$$

But $X^* \succeq 0$ in Theorem 2.6 implies that $\|y^*\|^2 \leq \operatorname{trace} \bar{X}^* = s^2$, i.e.

$$|t^* - \lambda^*| \leq \|a\| s.$$

Since $\lambda_1(D(t^*)) = \lambda^* \leq \lambda_1(A)$, the upper bound is proved.

If the hard case holds, then $t^* \geq \lambda_1(D(t^*)) = \lambda^* = \lambda_1(A)$, and the lower bound is proved.

If the hard case does not hold, then λ^* is a simple eigenvalue of $D(t^*)$. Let a normalized eigenvector be $(y_0(t^*), z(t^*)^t)^t$. From Lemma 3.5, we know that $(y_0(t^*) \neq 0$. Set

$$v^* := \frac{1}{y_0(t^*)} z(t^*).$$

To derive the lower bound we consider two cases.
Case 1: $t^* \geq \lambda_1(A)$:
In this case the lemma is proved.
Case 2: $t^* < \lambda_1(A)$:
We get, using (3.5), (3.6) and the fact that $t \geq \lambda_1(D(t))$, that

$$t^* - \lambda_1^* = a^t v^* = (Av^* - \lambda_1^* v^*)^t v^* \geq (\lambda_1(A) - \lambda_1^*) s^2 \geq (\lambda_1(A) - t^*) s^2.$$

Let $A = PDP^t$ be the spectral decomposition of A with $PP^t = I$, and D containing the eigenvalues of A. Set $b := P^t a$. Then we can use $t = d(\lambda)$, see Lemma 3.2, to get an upper bound on $t^* - \lambda_1^*$. From (3.2)

$$t^* - \lambda_1^* = \sum_j \frac{b_j^2}{\lambda_j(A) - \lambda_1^*} = \sum_j \frac{b_j^2}{\lambda_j(A) - t^* + \sum_k \frac{b_k^2}{\lambda_k(A) - \lambda_1^*}}$$

$$\leq \sum_j \frac{b_j^2}{\lambda_j(A) - t^*} \quad \leq \quad \frac{||a||^2}{\lambda_1(A) - t^*},$$

since $\lambda_1(A) - t^* > 0$ and $\|b\| = \|a\|$. From the above sets of inequalities we have

$$(\lambda_1(A) - t^*)s^2 \leq \frac{||a||^2}{\lambda_1(A) - t^*}.$$

Therefore we conclude

$$|\lambda_1(A) - t^*| \leq \frac{||a||}{s}.$$

After noting again that $\lambda_1(A) - t^* > 0$, the proof is complete.

$\square$

Note that we only need upper and lower bounds on the smallest eigenvalue of A to apply this lemma. Upper and lower bounds on the optimal value can similarly be found, i.e. if v is a normalized eigenvector for $\lambda_1(A)$, then

$$s^2\lambda_1(A) - 2||a||s \leq \mu^* \leq s^2\lambda_1(A) + 2s|a^t v| \leq s^2\lambda_1(A) + 2s||a||. \qquad (4.2)$$

Algorithm Part 1: Initialization

Input: A, a, s is the data for (TRS); $feastol, dgaptol, zerotol$ are respectively, tolerances for feasibility, duality gap, and accuracy in determining t.
Find: the smallest eigenvalue and corresponding eigenvector for A
- use e.g. Lanczos algorithm with restart.
Find the bounds for intervals of uncertainty:
$t_0^l \leq t^* \leq t_0^u$ *(initial interval of uncertainty for t^*)*
$\mu^l \leq \mu^* \leq \mu^u$ *(initial interval of uncertainty for μ^*, duality gap)*
Set:
$t_c = t_0^l$ *(c denotes current estimate of t^*)*
$t_c^l = t_0^l, t_c^u = t_0^u$
$t_p^l = t_0^l, t_p^u = t_0^u$ *(p denotes previous estimate of t^*)*

4.2 Stopping Criteria

We stop if we have reached a solution which is almost optimal and almost feasible, or if the interval of uncertainty for t cannot be made any smaller, i.e. this interval has reached machine accuracy. More precisely, suppose we are given: feasibility and duality gap tolerances, $feastol$ and $dgaptol$, respectively; machine accuracy $macheps$; and the current duality gap $\mu^u - \mu^l$, where $\mu^u \geq \mu^* \geq \mu^l$. We stop when the current estimate of the optimum, denoted v, satisfies

$$||v|| \leq s(1 + feastol)$$

and the current duality gap $\mu^u - \mu^l$ satisfies

$$\mu^u - \mu^l \ \leq (\mu^u + 1) dgaptol;$$

or, we stop when the current interval of uncertainty for t^* satisfies

$$t^u - t^l \leq (|t^u| + 1) macheps.$$

Note that the relationships between duality gap and complementary slackness show that almost feasibility and a small duality gap imply almost stationarity, see (2.11) and (2.14).

The output from the program yields intervals of uncertainty for t^*, λ^*, μ^* and an almost feasible point v with objective value in the interval of uncertainty for μ^*.

4.3 Generating an Improved Iterate

The new trial point t_+ is generated in several different ways. We would like to use the properties of y_0, the first component of a normalized eigenvector of $D(t)$, see Section 3.1. The values of y_0 are used to predict a new value for t. (This compares to the dual simplex method where shadow prices are used to find new values.) We solve the stationarity condition $k'(t) = 0$ (see (3.11)) by solving the modified equation $\psi(t) = 0$ (see(3.12)). Ideally, we would like to use inverse interpolation on y_0 of the form

$$t(y_0) = c_0 + \frac{c_1}{1 - y_0} + \frac{c_2}{y_0} \tag{4.3}$$

(possibly with $c_1 = -c_2$, if only two points are available). This takes care of the fact that for $y_0 \to 0$, we have $t \to +\infty$, and $y_0 \to 1$ implies $t \to -\infty$, see the above Section 3.1. Using the information from the previous iterates, we first compute the c_i and set

$$t_+ = t(y^*) = t(\frac{1}{\sqrt{s^2+1}}).$$

The inverse interpolation works well in the easy case when we interpolate with points on the *same side* of t^*. Recall that ψ is concave, nonincreasing, and almost linear. However, in the hard case, or near hard case, the function $y_0(t)$ can be wildly different on opposite sides of t^*. This can result in very poor interpolation. To overcome this problem, the algorithm performs a primal step in the update part, i.e. a move to the boundary of the (PSDP) feasible set.

Since our interpolation scheme does not take into account the monotonicity of y_0, it may be the case that the new trial point is outside the interval of interest. In this case we exploit the concavity of $k(t)$: we use the intersection point of the tangent lines to $k(t)$ at two points on opposite sides of t^*, call this $(tt, k(tt)$; the value $k(tt)$ provides an upper bound on the optimal value μ^*, while tt provides a new interpolation point.

Remark 4.2 *In [29], the Newton method is applied to $\phi(\lambda) = 0$, to obtain a new approximation for λ^*. This iteration converges monotonically if the current estimate is on the* good side *of λ^*, i.e. less than λ^* but bigger than $-\lambda_1(A)$. However, the new approximation can be very poor if the current estimate is on the other side, the* bad side, *of λ^*; this can result in several steps of backtracking to maintain positive definiteness. For our algorithm, the inverse interpolation for t^* replaces Newton's method. Inverse linear interpolation would result in the same monotonic convergence behaviour, though backtracking is never needed.*

Algorithm Part 2: Find a new estimate t_+

$t_+ = \frac{t_c^l + t_c^u}{2}$ *(midpoint is default)*
if $t_0^l < t_c^l$ *and* $t_c^u < t_0^u$ *(points on both sides of t^* found already)*
    Find the intersection point (tt, ktt) of the two tangent lines
        to the graph of $k(t)$ at the points t_c^l and t_c^u
    set $t_+ = tt$
    update the upper bound μ^u using ktt
endif
find estimate tt for t^* using inverse interpolation

if *tt is inside current interval of uncertainty*

set $t_+ = tt$

endif

4.4 Updating Information

After having determined the new trial point t_+, we calculate $\lambda_1(D(t_+))$ together with a normalized eigenvector. There exist several publicly available software packages that are designed specifically to find extreme eigenvalues of symmetric matrices. With a "Block-Lanczos" approach, the computational effort boils down to essentially a sequence of multiplications of the form "matrix times vector". Since we change only one element of $D(t)$ in the course of the iterations, we can make efficient use of previously calculated eigenvectors to start new iterations.

We take advantage of the special structure of our problem in several ways.

First, we test the sign of $\lambda = \lambda_1(D(t_+))$. If the sign is positive and $t_+ \leq t^*$, then we cannot get the correct negative sign for larger values of t, i.e. the constraint is redundant.

Next, if the sign of λ is correct, we check if $t_+ < t^*$ (bad side), then we take a primal step to the boundary in (PSDP) while improving the value of the objective function. (This is equivalent to the negative curvature direction described in Remark 2.4; however, we use an existing eigenvector from a previous iteration for the direction.) If, on the other hand, $t_+ > t^*$ (good side), then we use a steepest descent step to the boundary in (PSDP) while again improving the value of the objective function.

Algorithm Part 3: Updating

Find smallest eigenvalue λ *and*

corresponding normalized eigenvector for $D(t_+)$*.*

(Use Lanczos algorithm with previous eigenvector information.)

if $\lambda > 0$ *(wrong sign for Lagrange multiplier)*

if $t_+ \leq t^*$ *then* **STOP**, $x^* = A^\dagger a,\ \mu^* = -a^t x^*$

else *(correct sign for Lagrange multiplier)*

Update lower bound μ^l *using the value of the dual function* $k(t_+)$

endif

if $t_+ < t^*$

Perform a PSDP (primal step to boundary)
step and update best feasible data and μ^u
$t_p^l \leftarrow t_c^l, \; t_c^l \leftarrow t_+$

else

Take a steepest descent step to update μ^u
$t_p^u \leftarrow t_c^u, \; t_c^u \leftarrow t_+$

endif

5 Conclusion

5.1 Summary

We have presented a primal-dual semidefinite framework for trust region subproblems. This framework can be used to generate and analyze many known and new algorithms. In particular, we have presented an algorithm for large sparse trust region subproblems. The algorithm maintains dual feasibility and complementary slackness while iterating to obtain primal feasibility and reduce the duality gap. Each iteration of the algorithm requires the calculation of the smallest eigenvalue, and corresponding eigenvector, of the parametric matrix $D(t)$. This is done using a Lanczos routine. Therefore sparsity of the data can be fully exploited and the work of the algorithm is based on matrix vector multiplications. Moreover, only one element of the matrix is perturbed at each iteration and so consecutive Lanczos steps become cheaper.

Independent work, on a similar algorithm as presented herein, was done by Sorensen in [41]. The basic idea of inverse interpolation for a parametric eigenvalue problem lies behind both algorithms. The parametric eigenvalue connection is made precise in Theorem 3.7. The main differences in the two algorithms is that we exploit the semidefinite framework and the structure of the function $k(t)$. The structure of this function allows for very efficient handling of the hard case. In fact, the primal feasibility step reduces the duality gap and is essentially equivalent to the so-called negative curvature step taken in [29]. In addition, our approach does not use the optimality conditions to ensure that the objective value is sufficiently accurate, but rather it reduces the duality gap directly as do the current family of primal-dual interior-point algorithms for linear programs.

5.2 Numerics

We have run tens of thousands of tests for the inequality constrained case using randomly generated problems of various dimensions. There were various forms of the hard case used. These involved different multiplicities for the smallest eigenvalue of A. In particular, some of the problems had very high multiplicity, up to $n/3$. The problems were generated to have special properties: general problems using random A, a, s; general problems but with the hard case; and the random matrix A is chosen positive definite for both easy and hard case problems.

We had a relative duality gap tolerance of 10^{-5} and a feasibility tolerance of 10^{-5}. However, decreasing the desired tolerances did not increase the number of iterations significantly, thus indicating that we had very fast asymptotic convergence.

Our initial results used a simple linear interpolation. They showed an average number of approximately 4.5 iterations with essentially zero problems having more than 10 iterations. (More recent tests used the improved interpolation scheme (4.3). This gave a 25% reduction in the iteration count in the easy case.) The algorithm has yet to fail on a problem. Moreover, the number of iterations was, surprisingly, independent of the dimension of the problem. The time spent in the Lanczos step increased with the dimension, as expected. However, Lanczos efficiency increased after the first iteration, since we gain information on the starting approximation to the eigenvector. We observed that the total work for the algorithm in the easy case was approximately 1.8 times the work for the first call to the Lanczos algorithm. Since the trust region problem is a generalization of the minimum eigenvalue problem, we cannot expect to do much better than this.

The tests were done on a SUN SPARC station 1 using MATLAB with a fortran interface for the Lanczos algorithm. Average results for several dimensions follow. The hard case examples had multiplicity of the smallest eigenvalue ranging from 1 to 6. A multiple of the identity was added when a positive definite matrix was needed. The work is the amount of work times one Lanczos step.

dim	probs	density	iters	cpu sec	work	comment
1540-1565	30	.01	4.4453	54.1876	1.73	
1540-1558	30	.013	4.1148	56.1322	1.75	pos def Hessian
1500-1565	2	.01	4.2951	57.0429	4.8589	hard case
1800-1865	12	.05	4.2453	28.1692	4.2589	hard case

The codes and a this research report are available by anonymous ftp (to orion.uwaterloo.ca, in the directory pub/henry/software/trustreg.d) and through WWW using URL http://orion.uwaterloo.ca/~hwolkowi.

References

[1] F. ALIZADEH. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, 5:13–51, 1995.

[2] J. PENG andY. YUAN. Generalizations of the trust region subproblem. Technical report, Institute of Computational mathmatics and Scientific Computing, Academic Sinica, Beijing 100080 China, 1995.

[3] A. BEN-TAL and M. TEBOULLE. Hidden convexity in some nonconvex quadratically constrained quadratic programming. Technical report, Israel Institute of Technology, Haifa, Israel, 1993.

[4] S. BOYD, L. El GHAOUI, E. FERON, and V. BALAKRISHNAN. *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA, June 1994.

[5] M.R. CELIS, J.E. DENNIS Jr., and R.A. TAPIA. A trust region strategy for nonlinear equality constrained optimization. In *Proceedings of the SIAM Conference on Numerical Optimization, Boulder, CO*, 1984. Also available as Technical Report TR84-1, Department of Mathematical Sciences, Rice University, Houston, TX.

[6] T.F. COLEMAN and C. HEMPEL. Computing a trust region step for a penalty function. *SIAM J. Sci. STAT. Comput.*, 11:180–201, 1990.

[7] O.E. FLIPPO and B. JANSEN. Duality and sensitivity in nonconvex quadratic optimization over a ellipsoid. Technical Report 93-15, Technical University of Delft, Delft, The Netherlands, 1993.

[8] G.E. FORSYTHE and G.H. GOLUB. On the stationary values of a second-degree polynomial on the unit sphere. *SIAM J. Applied Mathematics*, 13:1050–1068, 1965.

[9] W. GANDER, G.H. GOLUB, and U. von MATT. A constrained eigenvalue problem. *Linear Algebra and its Applications*, 114/115:815–839, 1989.

[10] D.M. GAY. Computing optimal locally constrained steps. *SIAM J. Sci. Statist. Comput.*, 2:186–197, 1981.

[11] S.M. GOLDFELDT, R.E. QUANDT, and H.F. TROTTER. Maximization by quadratic hill-climbing. *Econometrica*, 34:541–551, 1966.

[12] G. GOLUB and U. von MATT. Quadratically constrained least squares and quadratic problems. *Numer. Math.*, 59:561–580, 1991.

[13] C.-G. HAN, P.M. PARDALOS, and Y.YE. Computational aspects of an interior point algorithm for quadratic programming problems with box constraints. In T.F. COLEMAN and Y. LI, editors, *Large Scale Numerical Optimization*. SIAM, 1990.

[14] M.D. HEBDEN. An algorithm for minimization using exact second derivatives. Technical Report TP515, Atomic Energy Research Establishment, Harwell, England, 1973.

[15] C. HELMBERG, F. RENDL, R. J. VANDERBEI, and H. WOLKOWICZ. An interior point method for semidefinite programming. *SIAM Journal on Optimization*, pages 342–361, 1996. URL: ftp://orion.uwaterloo.ca/pub/henry/reports/sdp.ps.gz.

[16] A.E. HOERL and R.W. KENNARD. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970.

[17] R. HORN and C. JOHNSON. *Matrix Analysis*. Cambridge University Press, New York, 1985.

[18] N. KARMARKAR. An interior-point approach to NP-complete problems. Lecture, 3rd SIAM Conference on Optimization, Boston, 1989.

[19] SERGE KRUK. Semidefinite programming applied to nonlinear programming. Master's thesis, University of Waterloo, 1996.

[20] K. LEVENBERG. A method for the solution of certain problems in least squares. *Quart. Appl. Math.*, 2:164–168, 1944.

[21] A. LEWIS. Take-home final exam. Course CO663 in Convex Analysis, University of Waterloo, 1994.

[22] L. LOVÁSZ. On the Shannon capacity of a graph. *IEEE Transactions on Information Theory*, 25:1–7, 1979.

[23] L. LOVÁSZ and A. SCHRIJVER. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization*, 1(2):166–190, 1991.

[24] D. MARQUARDT. An algorithm for least-squares estimation of non-linear parameters. *SIAM J. Appl. Math.*, 11:431–441, 1963.

[25] J.M. MARTINEZ. Local minimizers of quadratic functions on Euclidean balls and spheres. *SIAM Journal on Optimization*, 4(1):159–176, 1994.

[26] J.J. MORÉ. The Levenberg-Marquardt algorithm: implementation and theory. In G.A. Watson, editor, *Lecture Notes in Mathematics #630, Numerical Analysis*, pages 105–116. Springer-Verlag, Berlin, Heidelberg and New York, 1977.

[27] J.J. MORÉ. Recent developments in algorithms and software for trust region methods. In Bachem, Grotschel, and Korte, editors, *Mathematical Programming, the State of the Art*, pages 268–285. 1983.

[28] J.J. MORÉ. Generalizations of the trust region problem. Technical Report MCS-P349-0193, Argonne National Labs, Argonne, IL, 1993.

[29] J.J. MORÉ and D.C. SORENSEN. Computing a trust region step. *SIAM J. Sci. Statist. Comput.*, 4:553–572, 1983.

[30] Y. E. NESTEROV and A. S. NEMIROVSKY. *Interior Point Polynomial Algorithms in Convex Programming : Theory and Algorithms.* SIAM Publications. SIAM, Philadelphia, USA, 1994.

[31] M.L. OVERTON. Large-scale optimization of eigenvalues. *SIAM J. Optimization*, 2:88–120, 1992.

[32] S. POLJAK, F. RENDL, and H. WOLKOWICZ. A recipe for semidefinite relaxation for (0,1)-quadratic programming. *Journal of Global Optimization*, 7:51–73, 1995.

[33] M.V. RAMANA. *An algorithmic analysis of multiquadratic and semidefinite programming problems.* PhD thesis, Johns Hopkins University, Baltimore, Md, 1993.

[34] M.V. RAMANA and A.J. GOLDMAN. Some geometric results in semidefinite programming. Technical report, Johns Hopkins University, Baltimore, Md, 1994.

[35] C. REINSCH. Smoothing by spline functions. *Numer. Math.*, 10:177–183, 1967.

[36] C. REINSCH. Smoothing by spline functions II. *Numer. Math.*, 16:451–454, 1971.

[37] F. RENDL, R. J. VANDERBEI, and H. WOLKOWICZ. Max-min eigenvalue problems, primal-dual interior point algorithms, and trust region subproblems. *Optimization Methods and Software*, 5:1–16, 1995.

[38] S.A. SANTOS and D.C. SORENSEN. A new matrix-free algorithm for the large-scale trust-region subproblem. Technical Report TR95-20, Rice University, Houston, TX, 1995.

[39] N.Z. SHOR. Quadratic optimization problems. *Soviet J. of Computer and Systems Science*, 25(6):1–11, 1987.

[40] D.C. SORENSEN. Newton's method with a model trust region modification. *SIAM J. Numer. Anal.*, 19:409–426, 1982.

[41] D.C. SORENSEN. Minimization of a large scale quadratic function subject to an ellipsoidal constraint. Technical Report TR94-27, Rice University, Houston, TX, 1994.

[42] R. STERN and H. WOLKOWICZ. Trust regions and nonsymmetric eigenvalue perturbations. *SIAM J. Matrix Analysis and Applications*, 15(3):755–778, 1994.

[43] R. STERN and H. WOLKOWICZ. Indefinite trust region subproblems and nonsymmetric eigenvalue perturbations. *SIAM J. Optimization*, 5(2):286–313, 1995.

[44] R.J. STERN and J.J. YE. Variational analysis of an extended eigenvalue problem. *Linear Algebra and Its Applications*, 1992. to appear.

[45] G.W. STEWART III. *Introduction to Matrix Computations.* Academic Press, New York, NY, 1973.

[46] P.D. TAO and L. T. H. AN. D.C. (difference of convex functions) optimization algorithms (DCA) for globally minimizing nonconvex quadratic forms on Euclidean balls and spheres. Technical report, LMI, INSA, Rouen, Mont Saint Aignan Cedex, France, 1995.

[47] A.N. TIKHONOV and V.Y. ARSENIN. *Solutions of Ill-Posed Problems.* V.H. Winston & Sons, John Wiley & Sons, Washington D.C., 1977. Translation editor Fritz John.

[48] L. VANDENBERGHE and S. BOYD. Semidefinite programming. *SIAM Review*, 38:49–95, 1996.

[49] S.A. VAVASIS and R. ZIPPEL. Proving polynomial-time for sphere-constrained quadratic programming. Technical report, Department of Computer Science, Cornell University, Ithaca, NY, 1990.

[50] J.H. WILKINSON. *The Algebraic Eigenvalue Problem.* Oxford University Press, London, 1965.

[51] H. WOLKOWICZ. Some applications of optimization in matrix theory. *Linear Algebra and its Applications*, 40:101–118, 1981.

[52] Y. YE. A new complexity result on minimization of a quadratic function with a sphere constraint. In *Recent Advances in Global Optimization*, pages 19–31. Princeton University Press, 1992.

[53] Y. YE. On affine scaling algorithms for nonconvex quadratic programming. *Mathematical Programming*, 56:285–300, 1992.