An Augmented Primal-Dual Method for Linear Conic Programs

Florian Jarre
Institut für Mathematik
Universität Düsseldorf
Universitätsstraße 1
D–40225 Düsseldorf, Germany
e-mail: jarre@opt.uni-duesseldorf.de
and
Franz Rendl
Institut für Mathematik
Universität Klagenfurt
Universitätsstraße 65-67
A–9020 Klagenfurt, Austria
e-mail: franz.rendl@uni-klu.ac.at

April 11, 2007

Abstract. We propose a new iterative approach for solving linear programs over convex cones. Assuming that Slaters condition is satisfied, the conic problem is transformed to the minimization of a convex differentiable function. This "agumented primal-dual function" or "apd-function" is restricted to an affine set in the primal-dual space. The evaluation of the function and its derivative is cheap if the projection of a given point onto the cone can be computed cheaply, and if the projection of a given point onto the affine subspace defining the primal problem can be computed cheaply. For the special case of a semidefinite program, a certain regularization of the apd-function is analyzed. Numerical examples minimizing the apd-function with a conjugate gradient method illustrate the potential of the approach.

Key words. Conic program, linear convergence, augmented primal-dual function.

1 Introduction

We present a new method for solving convex conic programs. The method is based on minimizing a convex differentiable "augmented primal-dual function" (apd-function) that is related to the augmented Lagrangian but less problem dependent and does not require any penalty parameter. If Slaters condition is satisfied, the problem of solving the conic program is equivalent to minimizing the apd-function. The evaluation of function value and gradient of the apd-function requires two conic projections and four projections on an affine subspace. If these projections are cheap it is possible to minimize the apd-function by any descent method such as a conjugate gradient type method or a limited memory BFGS method. In our numerical examples we report results obtained with a conjugate gradient approach.

When applying this algorithm to the apd-function of a linear program in standard form with a system matrix $A \in \mathbb{R}^{m\times n}$, a factorization of the matrix AA^T can be computed in a preprocessing phase. Given this factorization the cost per iteration is of order $O(mn)$ arithmetic operations. When minimizing the apd-function of a linear program by Newton's method – at a cost of order $O(n^3)$ arithmetic operations per iteration – this algorithm converges in a finite number of iterations. We therefore anticipate that a conjugate gradient or BFGS approach will converge rapidly as well.

When applied to a semidefinite program having a unique and strictly complementary solution the algorithm is sublinearly convergent. We therefore derive a simple modification of the apd-function for which Newtons method is locally quadratically convergent.

Our approach is related to projection methods such as considered for example in [1]. New elements of this paper are a transformation of a conic problem into an affine-convex pair with cheap projections onto the affine set, a conjugate gradient acceleration, a regularization for the case of semidefinite programs, and promising numerical examples.

Several methods have been proposed in the literature to overcome the limits of interior point methods for solving large-scale semidefinite programs. We recall the spectral bundle method [4] which uses eigenvalue optimization. The low-rank factorization approach of Burer and Monteiro [2] treats semidefinite programs using nonlinear optimization techniques. The currently strongest computational results are reported in the papers by Toh [11] and by Kocvara and Stingl [5]. Toh uses an iterative solver for the augmented KKT system, and Kocvara and Stingl apply an iterative solver to a modified barrier problem. The approach presented in the current paper is closely related to the 'boundary point method' from [8] and the regularization approaches in [6].

2 Linear conic programs

We consider linear conic programs of the form

$$(P) \qquad \text{minimize } \langle c, x\rangle \mid x \in K \cap (\mathcal{L} + b),$$

where K is a closed convex cone in a finite dimensional Euclidean space E, $\mathcal{L}$ is a linear subspace, and $b, c \in E$ are given data. We always assume that the dimension of E is denoted by n. Our practical applications refer to the cases where K is the positive orthant in $E = \mathbb{R}^n$ and where K is the cone of symmetric positive semidefinite matrices in $E = \mathcal{S}^l = \{X \in \mathbb{R}^{l\times l} \mid X = X^T\}$. Here, $n = l(l+1)/2$.

We always assume that K has a nonempty interior (no hidden equality constraints) and that K is pointed (in LP-notation this assumption means there are no free variables).

Often the set $\mathcal{L}$ is given in the form

$$\mathcal{L} = \{\tilde{x} \mid A\tilde{x} = 0\} \qquad \text{and} \qquad \mathcal{L} + b = \{x \mid Ax = Ab\}, \tag{1}$$

where A is a matrix or some other representation of a linear operator. In particular, our analysis yields an algorithm applicable to linear programs of the form

$$\text{minimize } c^T x \mid Ax = \bar{b}, \quad x \geq 0,$$

where $\bar{b} := Ab$.

We use the dual program as introduced in [7],

$$(D) \qquad \text{minimize } \langle b, s\rangle \mid s \in K^D \cap (\mathcal{L}^\perp + c),$$

where $\mathcal{L}^\perp$ is the orthogonal complement of $\mathcal{L}$ and

$$K^D = \{s \in E \mid \langle s, x\rangle \geq 0 \ \forall x \in K\}$$

is the dual cone of K.

It is easily verified that weak duality holds, namely

$$\langle b, c\rangle \leq \langle c, x\rangle + \langle b, s\rangle$$

for all x, s that are feasible for (P) and (D). When (P) satisfies Slaters condition and (D) has a feasible solution then strong duality holds, see [7]. In this case, a point x is optimal for (P) if, and only if, there exists a point s feasible for (D) with

$$\langle b, c\rangle = \langle c, x\rangle + \langle b, s\rangle. \tag{2}$$

We denote such x and s by x^{opt} and s^{opt}.

3 Decomposing the conic program

The linear constraints of (P) and (D) (including (2)) are satisfied for all points in the affine space

$$\boldsymbol{A} := (\mathcal{L} + b) \times (\mathcal{L}^\perp + c) \cap \{(x, s) \mid \ \langle c, x\rangle + \langle b, s\rangle = \langle b, c\rangle\} \subset E \times E,$$

and the conic constraints are satisfied for all points in the cone

$$\boldsymbol{C} := K \times K^D \subset E \times E.$$

By the assumption on K, it follows that $\boldsymbol{C}$ is full dimensional, $\dim(\boldsymbol{C}) = 2n$. We assume that $\boldsymbol{A}$ is of dimension $n-1$. (In the case that $b \in \mathcal{L}$ and $c \in \mathcal{L}^\perp$ the set $\boldsymbol{A}$ is of dimension n. As we do not provide solutions in the relative interior of the solution set, this case is trivial with optimal solution $z^{opt} = 0$.)

Solving (P) hence is equivalent to finding $z := (x; s) \in \boldsymbol{A} \cap \boldsymbol{C}$ where $\boldsymbol{A}$ is an affine subspace and $\boldsymbol{C}$ a convex cone. Moreover, as we will see, projections onto $\boldsymbol{A}$ and $\boldsymbol{C}$ are easily computable for the case of linear or semidefinite programming.

For a closed set $\mathcal{C}$ and a vector $\bar{z}$ we denote the distance of $\bar{z}$ to $\mathcal{C}$ by

$$d(\bar{z}, \mathcal{C}) := \min\{\|z - \bar{z}\|_2 \mid z \in \mathcal{C}\}.$$

Thus, solving (P) is also equivalent to finding z such that

$$\phi(z) := \frac{1}{2}(d(z, \boldsymbol{A})^2 + d(z, \boldsymbol{C})^2) = 0,$$

i.e. such that the differentiable convex function ϕ is minimized. (Differentiability of ϕ is shown in Remark 1 below.)

When (P) is a linear program in standard form, the function $d(z, \boldsymbol{C})^2$ is of the form $\sum_i ((z_i)^+)^2$ where $(z_i)^+ := \max\{0, z_i\}$. Thus, ϕ is closely related to the augmented Lagrangian function. We therefore call ϕ an augmented primal-dual function. It differs from the augmented Lagrangian in that the representation of the linear subspace $\mathcal{L}$ (i.e. the matrix A when $\mathcal{L}$ is of the form (1)) does not enter the representation of ϕ. In other words, ϕ is less “data dependent” than the augmented Lagrangian – and it depends on a larger number of unknowns. As we will see, however, the dependence on a large number of unknowns does not imply that computations with ϕ are numerically expensive.

Remark 1 *For a closed convex set $\mathcal{C}$ let*

$$\Pi_{\mathcal{C}} \textit{ be the orthogonal projection}$$

(with respect to the Euclidean norm) onto $\mathcal{C}$. Then, given an algorithm for the evaluation of $\Pi_{\mathcal{C}}$, the distance $d(z, \mathcal{C}) = \|z - \Pi_{\mathcal{C}}(z)\|_2$ is easily computed. Moreover, a steepest descent step of step length 1 starting at z for minimizing the differentiable function $\phi_{\mathcal{C}}$ with

$$\phi_{\mathcal{C}}(z) := \frac{1}{2} d(z, \mathcal{C})^2$$

will lead to the nearest point minimizing d (i.e. to $\Pi_{\mathcal{C}}(z)$). Unfortunately, this property is lost when minimizing the sum $\phi(z) = \frac{1}{2}(d(z, \boldsymbol{A})^2 + d(z, \boldsymbol{C})^2)$ by the steepest descent method.

Proof. For completeness we provide an elementary proof of Remark 1. We show that $\phi_{\mathcal{C}}(z)$ is a differentiable function and $\nabla \phi_{\mathcal{C}}(z) = z - \Pi_{\mathcal{C}}(z)$. Let $\hat{z} := \Pi_{\mathcal{C}}(z)$. Let Δz be arbitrary. We show that

$$\phi_{\mathcal{C}}(z + \lambda \Delta z) = \phi_{\mathcal{C}}(z) + \lambda \Delta z^T (z - \hat{z}) + o(|\lambda|).$$

First note that

$$2\phi_{\mathcal{C}}(z + \lambda \Delta z) \le \|\hat{z} - (z + \lambda \Delta z)\|_2^2 = \|\hat{z} - z\|_2^2 - 2\lambda(\hat{z} - z)^T \Delta z + O(\lambda^2).$$

On the other hand, let $\hat{z}(\lambda) := \Pi_{\mathcal{C}}(z + \lambda \Delta z)$. As $\hat{z}(\lambda) \in \mathcal{C}$ it follows $(\hat{z}(\lambda) - \hat{z})^T (z - \hat{z}) \le 0$, and $\|\hat{z}(\lambda) - \hat{z}\|_2 \le \|\lambda \Delta z\|_2$. It follows

$$\begin{aligned} 2\phi_{\mathcal{C}}(z + \lambda \Delta z) &= \|\hat{z}(\lambda) - (z + \lambda \Delta z)\|_2^2 = \|(\hat{z}(\lambda) - \hat{z}) + (\hat{z} - z) - \lambda \Delta z\|_2^2 \\ &\ge \|\hat{z} - z\|_2^2 - 2\lambda(\hat{z} - z)^T \Delta z - O(\lambda^2). \end{aligned}$$

This completes the proof of Remark 1. □

Note that the projections onto $\boldsymbol{A}$ and $\boldsymbol{C}$ – and thus the function ϕ – are easy to compute for the case of linear and semidefinite programming:

For the case of linear programming, $\boldsymbol{C}$ is the positive orthant in $\mathbb{R}^{2n}$, and the projection onto $\boldsymbol{C}$ can be performed in $O(n)$ arithmetic operations. In the case of semidefinite programming, $\boldsymbol{C}$ is the cartesian product of the semidefinite cone with itself, and the projection onto $\boldsymbol{C}$ can be computed by performing the eigenvalue decompositions of two symmetric matrices (order l^3 operations).

In Section 6 it is shown that also the projection onto $\boldsymbol{A}$ can be done efficiently for these two examples. If $\mathcal{L}$ is given as in (1) with $Ab \in \mathbb{R}^m$ and the Cholesky factorization of AA^T is computed in a preprocessing step before starting the algorithm then the projection can be evaluated in $O(mn)$ operations.

4 Solving (P) and (D)

As shown in the previous section, solving (P) and (D) is reduced to finding a point in the intersection of the two convex sets $\boldsymbol{A}$ and $\boldsymbol{C}$, both of which are explicitely given. In this section we assume that the intersection of $\boldsymbol{A}$ and $\boldsymbol{C}$ is nonempty.

4.1 Minimizing the distance between $\boldsymbol{A}$ and $\boldsymbol{C}$

Standard projection methods solve the problem of finding a point in $\boldsymbol{A} \cap \boldsymbol{C}$ by the following *simple algorithm:*

Algorithm 1 (Alternating projections)
Initialization: Let $z^0 \in \boldsymbol{A}$ be given. Set $k = 0$.

1. *Set $\hat{z}^k := \Pi_{\boldsymbol{C}}(z^k)$.*
2. *Set $z^{k+1} := \Pi_{\boldsymbol{A}}(\hat{z}^k)$. Set $k = k+1$. Go to Step 1.*

By Remark 1, one iteration of Algorithm 1 can be interpreted as one steepest descent step for minimizing $\frac{1}{2}d(\ .\ , \boldsymbol{C})^2$ followed by a steepest descent step for minimizing $\frac{1}{2}d(\ .\ , \boldsymbol{A})^2$. In general, such method converges very slowly. We therefore consider an acceleration minimizing the sum of both functions by a conjugate gradient scheme:

4.2 Minimizing ϕ

The first simple approach for minimizing ϕ is a conjugate gradient type method with Polak-Ribière type update or Fletcher-Reeves type update of the search direction. For descent methods it is important to understand the behavior of the second derivative of the objective function.

For linear and semidefinite programming, the function ϕ is twice differentiable almost everywhere. (It is differentiable everywhere.) For linear programming the eigenvalues of the Hessian H of ϕ at any point z such that $H(z)$ exists are at most 2 (as each of the Hessians of $\frac{1}{2}d(\ .\ , \boldsymbol{A})^2$ and $\frac{1}{2}d(\ .\ , \boldsymbol{C})^2$ only has the eigenvalues zero and one.) The eigenvalues of H are nonnegative, but unfortunately, they may be zero or arbitrarily close to zero. This makes the application of descent methods for minimizing ϕ difficult. Before continuing our analysis of the function ϕ we reduce the number of degrees of freedom by restricting ϕ to a lower dimensional subspace:

Note that $\boldsymbol{A}$ is an affine subspace. We restrict ϕ to $\boldsymbol{A}$ and define the function $\tilde{\phi}$ by

$$\tilde{\phi}(\tilde{z}) := \phi(\tilde{z}) = \frac{1}{2}\|d(\tilde{z}, \boldsymbol{C})\|_2^2 \qquad \text{for } \tilde{z} \in \boldsymbol{A}. \tag{3}$$

We stress that $\tilde{\phi}$ is not defined outside $\boldsymbol{A}$. To emphasize this fact we also denote the variable by $\tilde{z}$ rather than just z.

Remark 2 *The function $\tilde{\phi}$ is differentiable, and for $\tilde{z} \in \boldsymbol{A}$ its gradient is given by*

$$\nabla\tilde{\phi}(\tilde{z}) = \tilde{z} - \Pi_{\boldsymbol{A}}(\Pi_{\boldsymbol{C}}(\tilde{z})).$$

Proof. Let $\boldsymbol{A} = \boldsymbol{z} + \boldsymbol{L}$ where $\boldsymbol{z}$ is a fixed vector and $\boldsymbol{L}$ a linear subspace. Note that

$$\tilde{z} - \Pi_{\boldsymbol{A}}(\Pi_{\boldsymbol{C}}(\tilde{z})) = \Pi_{\boldsymbol{L}}(\tilde{z} - \Pi_{\boldsymbol{C}}(\tilde{z})).$$

By Remark 1 it therefore suffices to recall the following more general (and well known) statement:

If $\varphi : E \times E \to \mathbb{R}$ is a differentiable function, then the gradient of the restriction $\tilde{\varphi}$ of φ to $\boldsymbol{A}$ is given by

$$\nabla\tilde{\varphi}(\tilde{z}) = \Pi_{\boldsymbol{L}}(\nabla\varphi(\tilde{z})).$$

The gradient of $\tilde{\varphi}$ at $\tilde{z} \in \boldsymbol{A}$ is a vector $\tilde{w} \in \boldsymbol{L}$ such that

$$\tilde{\varphi}(\tilde{z} + \Delta\tilde{z}) = \tilde{\varphi}(\tilde{z}) + \tilde{w}^T\Delta\tilde{z} + o(\Delta\tilde{z})$$

for all sufficiently small $\Delta\tilde{z} \in \boldsymbol{L}$. The vector $w := \Pi_{\boldsymbol{L}}(\nabla\varphi(\tilde{z}))$ certainly lies in $\boldsymbol{L}$. For $\Delta\tilde{z} \in \boldsymbol{L}$ it follows from symmetry of $\Pi_{\boldsymbol{L}}$ that

$$\begin{aligned}
\tilde{\varphi}(\tilde{z}) + w^T\Delta\tilde{z} &= \varphi(\tilde{z}) + (\Pi_{\boldsymbol{L}}(\nabla\varphi(\tilde{z})))^T\Delta\tilde{z} \\
&= \varphi(\tilde{z}) + (\nabla\varphi(\tilde{z}))^T\Pi_{\boldsymbol{L}}\Delta\tilde{z} \\
&= \varphi(\tilde{z}) + (\nabla\varphi(\tilde{z}))^T\Delta\tilde{z} \\
&= \varphi(\tilde{z} + \Delta\tilde{z}) + o(\Delta\tilde{z}) \\
&= \tilde{\varphi}(\tilde{z} + \Delta\tilde{z}) + o(\Delta\tilde{z}).
\end{aligned}$$

This completes the proof. □

A steepest descent step with line search for minimizing $\tilde{\phi}$ starting at a point $\tilde{z} = z^k \in \boldsymbol{A}$ is the same as the computation of z^{k+1} with Algorithm 1 followed by an extrapolation along the line $z^k + \lambda(z^{k+1} - z^k)$. We briefly list a conjugate gradient acceleration of the steepest descent approach:

Algorithm 2 (cg-method for minimizing $\tilde{\phi}$)
Let $\tilde{z}^0 \in \boldsymbol{A}$ be given. Let $\Delta\tilde{z}^0 := -\nabla\tilde{\phi}(\tilde{z}^0)$. Set $k = 0$.

1. *Let $\lambda_k :=$ argmin$\{\tilde{\phi}(\tilde{z}^k + \lambda\Delta\tilde{z}^k) \mid \lambda > 0\}$.*
2. *Set $\tilde{z}^{k+1} := \tilde{z}^k + \lambda_k\Delta\tilde{z}^k$.*
3. *Compute $\Delta\tilde{z}^{k+1}$ from $\Delta\tilde{z}^k$ and $\nabla\tilde{\phi}(\tilde{z}^{k+1})$ using an update formula such as Polak-Ribière.*
4. *If k is a multiple of $(n-1)$ set $\Delta\tilde{z}^{k+1} := -\nabla\tilde{\phi}(\tilde{z}^{k+1})$ (restart).*
5. *Set $k := k + 1$. Go to Step 1.*

Remark 3 *The concept of Algorithm 2 is in some sense 'complementary' to the boundary point method of [8]. The latter algorithm generates iterates within the primal-dual cone approaching the set of linear constraints, while the iterates in Algorithm 2 always satisfy the linear constraints and approach the primal-dual cone.*

Remark 4 *When $\boldsymbol{C}$ is polyhedral, and (P), (D) have a unique optimal solution z^{opt}, then the Hessian of $\tilde{\phi}$ is piecewise linear and positive definite near z^{opt} (since z^{opt} is necessarily strictly complementary!) and thus, Newton's method for minimizing $\tilde{\phi}$ converges in a finite number of iterations.*

Now consider the case where $\boldsymbol{C}$ is not polyhedral. Below we give a very simple example with a unique, strictly complementary optimal solution z^{opt} of (P) and (D) such that there are directions $z^{opt} + \lambda\Delta\tilde{z}$ through z^{opt} along which the intersection of $\boldsymbol{A}$ and $\boldsymbol{C}$ is "tangential" (the function $\tilde{\phi}$ in (3) growing in the order of λ^4) and other directions along which the intersection of $\boldsymbol{A}$ and $\boldsymbol{C}$ "transversal" ($\tilde{\phi}$ growing in the order of λ^2). This implies that the condition number of the Hessian of $\tilde{\phi}$ near z^{opt} is unbounded and the conjugate gradient method is likely to converge *sublinearly*! For the case of semidefinite programs we therefore derive an acceleration for this situation.

5 Application to semidefinite programs

In this section we use the following notation common for semidefinite programs: The space of real symmetric $l \times l$-matrices is denoted by $\mathcal{S}^l$. The dimension of $\mathcal{S}^l$ is $n := l(l+1)/2$. The notation $X \succeq 0$ (X $\succ 0$) is used to indicate that $X \in \mathcal{S}^l$ is positive semidefinite (positive definite). The standard scalar product on the space of $l \times l$-matrices is given by

$$\langle C, X\rangle := C \bullet X := \operatorname{trace}(C^T X) = \sum_{i,j=1}^{l} C_{i,j} X_{i,j}.$$

For given matrices $A^{(i)} \in \mathcal{S}^l$, $i = 1, 2, \ldots, m$, we define a linear map $\mathcal{A} : \mathcal{S}^l \to \mathbb{R}^m$ by

$$\mathcal{A}(X) := \begin{bmatrix} A^{(1)} \bullet X \\ \vdots \\ A^{(m)} \bullet X \end{bmatrix}, \quad X \in \mathcal{S}^l.$$

The adjoint operator $\mathcal{A}^* : \mathbb{R}^m \to \mathcal{S}^l$ is given by

$$\mathcal{A}^*(y) = \sum_{i=1}^{m} y_i A^{(i)}, \quad y \in \mathbb{R}^m.$$

With these definitions, the standard pair of primal and dual linear semidefinite programs can now be stated as follows:

(P) $\qquad$ minimize $\; C \bullet X \;$ subject to $\; \mathcal{A}(X) = \bar{b}, \quad X \succeq 0$

and

(D') $\qquad$ maximize $\; \bar{b}^T y \;$ subject to $\; \mathcal{A}^*(y) + S = C, \quad S \succeq 0.$

The dual program is equivalent (in the sense that the optimal solutions coincide) to

(D) $\qquad$ minimize $\; B \bullet S \;$ subject to $\; S \in \mathcal{L}^{\perp} + C, \quad S \succeq 0,$

where $B \in \mathcal{S}^l$ is such that $\mathcal{A}(B) = \bar{b}$ and $\mathcal{L} = \{X \in \mathcal{S}^l \mid \mathcal{A}(X) = 0\}$.

Assumption 1 *Throughout this section we assume that the matrices $A^{(i)}$ are linearly independent and that (P) and (D) are strictly feasible and that there is a unique and strictly complementary solution $Z^{opt} = (X^{opt}, S^{opt})$ of (P) and (D) satisfying $X^{opt} + S^{opt} \succ 0$.*

Simple example: We give a simple example of a pair of semidefinite programs (P) and (D) satisfying Assumption 1 such that the Hessian of $\tilde{\phi}$ (see (3)) has an unbounded condition number for $\bar{Z}$ near Z^{opt}. (The Hessian is not defined at Z^{opt}.) Let $m = 1$ and the data of (P) and (D) be given by

$$C := \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \qquad B := \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad \text{and} \quad A^{(1)} := \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}.$$

The primal-dual optimal solution $Z^{opt} = (X^{opt}, S^{opt}) = (B, C)$ is unique and strictly complementary. The space $\boldsymbol{L} := \mathcal{L} \times \mathcal{L}^{\perp} \cap \{(\Delta X, \Delta S) \mid C \bullet \Delta X + B \bullet \Delta S = 0\}$ is given by

$$\boldsymbol{L} = \left\{ \Delta Z = (\Delta X, \Delta S) = \left(\begin{pmatrix} 2a & -a \\ -a & b \end{pmatrix}, \begin{pmatrix} -b & -b \\ -b & 0 \end{pmatrix} \right) \mid \; a, b \in \mathbb{R} \right\}.$$

By construction, $Z^{opt} + \Delta Z \in \boldsymbol{A}$ for $\Delta Z \in \boldsymbol{L}$, and for small $|a|, |b|$ it is easily verified that

$$d(Z^{opt} + \Delta Z, \boldsymbol{C}) = O(|b|) \quad \text{if } a = 0, \qquad d(Z^{opt} + \Delta Z, \boldsymbol{C}) = O(a^2) \quad \text{if } b = 0.$$

Thus, the second directional derivative of $\tilde{\phi}$ is zero Z^{opt} along the direction $b = 0$ and positive along the direction $a = 0$. Minimizing $\tilde{\phi}$ by some conjugate gradient scheme will result in a very slow algorithm.

Discussion: Of course, the above example is not surprising. We have given a convex characterization of the optimal solution of a convex program as the intersection of two convex sets $\boldsymbol{A}$ and $\boldsymbol{C}$, each of which is easily computable. We do not have the property that this characterization is well conditioned under "reasonable assumptions". So far, a computable characterization of the optimal solution of a convex program with both properties – convexity and well conditionedness – is unknown. (The KKT conditions are well conditioned under suitable assumptions but the complementarity part of the KKT conditions is non convex.) This lack of a convex *and* well conditioned characterization of the optimal solution is responsible for the fact that most polynomial-time methods for convex programs use some homotopy approach to compute an optimal solution.

5.1 A local acceleration

We propose an acceleration that can be applied locally near the optimal solution $Z^{opt} = (X^{opt}, S^{opt})$ of (P) and (D), e.g. when the minimization of $\tilde{\phi}$ is turning slow.

Let $\hat{f}(Z) = \hat{f}(X, S) := \|XS - SX\|_F^2$. The non convex function $\hat{f}$ is minimized at Z^{opt}. It is differentiable and the derivative

$$\nabla_Z \hat{f}(Z) = 2 \begin{pmatrix} 2XSX - X^2S - SX^2 \\ 2SXS - S^2X - XS^2 \end{pmatrix}$$

can be computed in order n^3 operations. More precisely, by exploiting the fact that $XS = (SX)^T$, it can be evaluated with three matrix-matrix multiplications: two for evaluating $2XSX - XXS - SXX = X(SX - XS) + (X(SX - XS))^T$, and one more for the second block of $\nabla \hat{f}(Z)$. Also the derivative of the restriction of $\hat{f}$ to $\boldsymbol{A}$ can be computed as in the proof of Remark 2.

We therefore propose to solve (P) and (D) in two stages, the first one minimizing $\tilde{\phi}$ for $\tilde{Z} \in \boldsymbol{A}$, and when convergence of this stage is slow, starting a second stage minimizing $\tilde{\phi} + \hat{f}$ for $\tilde{Z} \in \boldsymbol{A}$. For both stages we may use a nonlinear cg-method as in Algorithm 2. The cg-method is n-step locally quadratically convergent if the objective function is three times differentiable near Z^{opt} and if the Hessian at Z^{opt} is positive definite. In Lemma 1 below, we show a slightly weaker statement.

Note: In the following we will only consider points in $\boldsymbol{A}$. For compactness of notation we omitt the additional identifcation $\tilde{Z}$ to indicate that $\tilde{Z} \in \boldsymbol{A}$ and shortly write $Z \in \boldsymbol{A}$. The restriction of $\phi + \hat{f}$ to $\boldsymbol{A}$ will be denoted by Ψ,

$$\Psi(Z) := \phi(Z) + \hat{f}(Z) \qquad \text{for } Z \in \boldsymbol{A}.$$

Again, we emphasize the restriction to $\boldsymbol{A}$.

Lemma 1 *The gradient of Ψ is strongly semismooth and the generalized Hessian is positive definite at Z^{opt}.*

By Theorem 3.2 in [9], Lemma 1 implies quadratic convergence of Newton's method for minimizing Ψ. We therefore anticipate that also conjugate gradient type algorithms or limited memory BFGS algorithms will converge rapidly.

Proof. Strong semismoothness of the gradient of Ψ at Z^{opt} follows from [10]; here, we prove positive definiteness of the generalized Hessian.

We start by noting that in spite of $\hat{f}$ not being convex, the eigenvalues of the Hessian of $\hat{f}$ at Z^{opt} are nonnegative since Z^{opt} is a minimizer of $\hat{f}$. Hence it suffices to show that either ϕ or $\hat{f}$ has a positive curvature along any given direction through Z^{opt}.

Let a perturbation $\Delta Z = (\Delta X, \Delta S)$ with $Z^{opt} + \Delta Z \in \boldsymbol{A}$ and $\|\Delta X\|_F^2 + \|\Delta S\|_F^2 = 1$ be given. It suffices to show that there exists a $\rho > 0$ independent of ΔZ such that

$$\frac{1}{2} d(Z^{opt} + \lambda \Delta Z, \boldsymbol{C})^2 + \hat{f}(Z^{opt} + \lambda \Delta Z) \geq \lambda^2 \rho$$

for sufficiently small $\lambda > 0$. By complementarity, $X^{opt} S^{opt} = 0 = S^{opt} X^{opt}$, and thus the matrices $X^{opt} \succeq 0$ and $S^{opt} \succeq 0$ commute. This guarantees that there exists a unitary matrix U and diagonal matrices

$$\Lambda = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_l) \succeq 0 \quad \text{and} \quad \Sigma = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_l) \succeq 0 \tag{4}$$

such that

$$X^{opt} = U \Lambda U^T \quad \text{and} \quad S^{opt} = U \Sigma U^T. \tag{5}$$

By strict complementarity we may assume without loss of generality that there exists a $k \leq l$ such that

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_k > 0 = \lambda_{k+1} = \ldots = \lambda_l$$

and

$$\sigma_1 = \sigma_2 = \ldots = \sigma_k = 0 < \sigma_{k+1} \leq \ldots \leq \sigma_l.$$

As shown in Corollary 1 in [3], the following system of $m+2n$ linear equations for $2n+m$ unknowns $(\Delta X \Delta S, \Delta y)$:

$$\begin{aligned} \mathcal{A}(\Delta X) &= p, \\ \mathcal{A}^*(\Delta y) + \Delta S &= Q, \\ \Pi_{\mathrm{up}}(U^T(\Delta X S^{opt} + X^{opt}\Delta S)U) &= r, \end{aligned} \tag{6}$$

is nonsingular. Here, $\Pi_{\mathrm{up}}(U^T(\Delta X S^{opt} + X^{opt}\Delta S)U)$ denotes the upper triangular part of the matrix $U^T(\Delta X S^{opt} + X^{opt}\Delta S)U$; the right hand side of (6) consists of $p \in \mathbb{R}^m$, $Q \in \mathcal{S}^l$ and the upper triangular part r of an $l \times l$-matrix. For brevity we write $r \in \mathbb{R}^n$.

We eliminate the variable Δy from the second equation of (6). To this end let $\mathcal{F} : \mathcal{S}^l \to \mathbb{R}^{n-m}$ be a linear operator of full rank such that $\mathcal{F}(\mathcal{A}^*(y)) = 0$ for all $y \in \mathbb{R}^m$. Let $q := \mathcal{F}(Q)$. By construction of $\mathcal{F}$, also the linear system

$$M \begin{pmatrix} \Delta X \\ \Delta S \end{pmatrix} := \begin{pmatrix} \mathcal{A}(\Delta X) \\ \mathcal{F}(\Delta S) \\ \Pi_{\mathrm{up}}(U^T(\Delta X S^{opt} + X^{opt}\Delta S)U) \end{pmatrix} = \begin{pmatrix} p \\ q \\ r \end{pmatrix} \tag{7}$$

has full rank. Here, $(p^T, q^T)^T \in \mathbb{R}^n$ and $r \in \mathbb{R}^n$.

First note that $\|(p^T, q^T, r^T)^T\| = \|M\Delta Z\| \geq 1/\|M^{-1}\|_2$ since $\|\Delta Z\| = 1$. From $Z^{opt} + \Delta Z \in \boldsymbol{A}$ it follows that $p = 0$ and $q = 0$. Hence , $\|r\|_2 \geq 1/\|M^{-1}\|_2$.

Note that problems (P) and (D) remain invariant when replacing B with X^{opt} and C with S^{opt}. Hence, from $Z^{opt} + \Delta Z \in \boldsymbol{A}$ it follows that

$$\begin{aligned} 0 &= C \bullet \Delta X + B \bullet \Delta S \\ &= S^{opt} \bullet \Delta X + X^{opt} \bullet \Delta S. \end{aligned} \tag{8}$$

Let $\widetilde{\Delta X} := U^T \Delta X U$ and $\widetilde{\Delta S} := U^T \Delta S U$. The last equation in (7) then states that

$$\Pi_{\mathrm{up}}(\widetilde{\Delta X}\Sigma + \Lambda\widetilde{\Delta S}) = r, \tag{9}$$

while relation (8) and $UU^T = I$ imply that

$$0 = \widetilde{\Delta X} \bullet \Sigma + \widetilde{\Delta S} \bullet \Lambda. \tag{10}$$

As U is unitary, $\|\widetilde{\Delta X}\|_F = \|\Delta X\|_F$, $\|\widetilde{\Delta S}\|_F = \|\Delta S\|_F$.

We partition $\widetilde{\Delta X}$ conforming with the zero-structure of Λ and Σ,

$$\widetilde{\Delta X} = \begin{pmatrix} \widetilde{\Delta X}_{11} & \widetilde{\Delta X}_{12} \\ \widetilde{\Delta X}_{12}^T & \widetilde{\Delta X}_{22} \end{pmatrix},$$

where $\widetilde{\Delta X}_{11} \in \mathcal{S}^k$ and $\widetilde{\Delta X}_{22} \in \mathcal{S}^{l-k}$. Likewise we partition $\widetilde{\Delta S}$. Let

$$\epsilon := \min\{\lambda_k, \sigma_{k+1}, 1/\lambda_1, 1/\sigma_l, 1/2\}.$$

From (9) follows

$$\|r\|^2 \leq \|\widetilde{\Delta X}_{12}\Sigma_{22}\|_F^2 + \|\widetilde{\Delta X}_{22}\Sigma_{22} + \Lambda_{11}\widetilde{\Delta S}_{11}\|_F^2 + \|\Lambda_{11}\widetilde{\Delta S}_{12}\|_F^2,$$

and by $\|r\|_2 \geq 1/\|M^{-1}\|_2$ this, and the definition of ϵ imply

$$\|\widetilde{\Delta X}_{12}\|_F^2 + \|\widetilde{\Delta X}_{22}\|_F^2 + \|\widetilde{\Delta S}_{11}\|_F^2 + \|\widetilde{\Delta S}_{12}\|_F^2 \geq \epsilon^2/\|M^{-1}\|_2^2. \tag{11}$$

Let $\delta \in (0,1]$ be given.

Assume that $\|\widetilde{\Delta X}_{22}\|_F \geq \epsilon\delta/(4\|M^{-1}\|_2) =: \mu$. The maximum absolute value of the eigenvalues of $\widetilde{\Delta X}_{22}$ is at least μ/n. If the maximum diagonal element is at least $\mu/8n^2$ it follows from (10) and the definition of ϵ that the smallest diagonal element of $\operatorname{Diag}(\widetilde{\Delta X}_{22}, \widetilde{\Delta S}_{11})$ and hence its smallest eigenvalue is at most $-\epsilon^2\mu/8n^3$. If the maximum diagonal element is less than $\mu/8n^2$ there is a 2 by 2 submatrix of $\widetilde{\Delta X}_{22}$ with eigenvalue less than $-\mu/8n^2$. By the interlacing property, $\widetilde{\Delta X}_{22}$ has an eigenvalue less than $-\mu/8n^2$. Thus, in both cases, the distance of $Z^{opt} + \lambda\Delta Z$ to $\boldsymbol{C}$ is at least $\lambda\epsilon^2\mu/8n^3$, and the function ϕ grows quadratically with λ.

The same argument holds when $\|\widetilde{\Delta S}_{11}\|_F \geq \mu$.

Now assume that $\|\widetilde{\Delta X}_{22}\|_F + \|\widetilde{\Delta S}_{11}\|_F < 2\mu$. By (11) it follows that and $\|\widetilde{\Delta X}_{12}\|_F^2 + \|\widetilde{\Delta S}_{12}\|_F^2 \geq \epsilon^2/(4\|M^{-1}\|_2^2)$. By symmetry of (P) and (D) we may assume again without loss of generality that $\|\widetilde{\Delta S}_{12}\|_F \geq \epsilon/(4\|M^{-1}\|_2)$. Observe that

$$\begin{aligned} \hat{f}(Z^{opt} + \lambda\Delta Z) &= \|(\Lambda + \lambda\widetilde{\Delta X})(\Sigma + \lambda\widetilde{\Delta S}) - (\Sigma + \lambda\widetilde{\Delta S})(\Lambda + \lambda\widetilde{\Delta X})\|_F^2 \\ &= \lambda^2(\|\Lambda\widetilde{\Delta S} + \widetilde{\Delta X}\Sigma - \Sigma\widetilde{\Delta X} - \widetilde{\Delta S}\Lambda\|_F^2) + O(\lambda^4) \\ &\geq 2\lambda^2\|\Lambda_{11}\widetilde{\Delta S}_{12} + \widetilde{\Delta X}_{12}\Sigma_{22}\|_F^2 + O(\lambda^4). \end{aligned}$$

For small λ, the fourth order term is dominated, and

$$\hat{f}(Z^{opt} + \lambda\Delta Z) \geq \lambda^2\|\Lambda_{11}\widetilde{\Delta S}_{12} + \widetilde{\Delta X}_{12}\Sigma_{22}\|_F^2.$$

We now assume by contradiction that $\|\Lambda_{11}\widetilde{\Delta S}_{12} + \widetilde{\Delta X}_{12}\Sigma_{22}\|_F^2 < \delta^2 := \epsilon^8/(64\|M^{-1}\|_2^2)$. This implies $\|\Lambda_{11}\widetilde{\Delta S}_{12}\Sigma_{22}^{-1} + \widetilde{\Delta X}_{12}\|_F < \delta/\epsilon$ and

$$\begin{aligned} \widetilde{\Delta X}_{12} \bullet \widetilde{\Delta S}_{12} &\leq -\Lambda_{11}\widetilde{\Delta S}_{12}\Sigma_{22}^{-1} \bullet \widetilde{\Delta S}_{12} + (\delta/\epsilon)\|\widetilde{\Delta S}_{12}\|_F \\ &\leq -\Delta S \bullet \Delta S\epsilon^2 + (\delta/\epsilon)\|\widetilde{\Delta S}_{12}\|_F \\ &\leq -\|\widetilde{\Delta S}_{12}\|_F^2(\epsilon^2 - 4\|M^{-1}\|_2\delta/\epsilon^2) \\ &\leq -\|\widetilde{\Delta S}_{12}\|_F^2\epsilon^2/2 < -\mu. \end{aligned}$$

On the other hand, (since $\|\widetilde{\Delta X}_{11}\|_F^2 + \|\widetilde{\Delta S}_{22}\|_F^2 < 1$)

$$\begin{aligned} 0 &= \Delta X \bullet \Delta S \\ &= \widetilde{\Delta X} \bullet \widetilde{\Delta S} \\ &= \widetilde{\Delta X}_{11} \bullet \widetilde{\Delta S}_{11} + 2\widetilde{\Delta X}_{12} \bullet \widetilde{\Delta S}_{12} + \widetilde{\Delta X}_{22} \bullet \widetilde{\Delta S}_{22} \\ &< 2\mu + 2\widetilde{\Delta X}_{12} \bullet \widetilde{\Delta S}_{12} \\ &< 2\mu - 2\mu = 0, \end{aligned}$$

which yields the desired contradiction. □

Remark 5 *The computational effort of the evaluation of the gradient of $\hat f$ is comparable to the evaluation of the gradient of $\tilde\phi$. Thus, it may be cheaper to first minimize $\tilde\phi$ until convergence slows down, and then minimize $\tilde\phi + \kappa\hat f$ where the factor $\kappa > 0$ is chosen as to compensate for the fact that $\hat f$ is not invariant under scaling of X and S.*

6 Cheap computation of the projection onto $\boldsymbol{A}$

First note that a projection onto an $n-1$-dimensional affine subspace of $\mathbb{R}^{2n}$ can (after an initial factorization of the projection matrix) generally be done in order n^2 operations. To make our algorithm practical we show that it can be done in a cheaper way for the particular sets $\boldsymbol{A}$ arising in linear programming. (Of course, the same reasoning applies to semidefinite programming replacing A^T with the adjoint $\mathcal{A}^*$.)

The computation of the projection below is closely related to rank-one update formulae for inverse matrices. There are two differences: We update a projection rather than an inverse matrix and the matrix defining the projection is never explicitely formed. (The matrix defining the projection may be nonsparse while A and the Cholesky factor of AA^T used below may be sparse.)

We assume that $\mathcal{L} + b = \{x \mid Ax = Ab\} \subset \mathbb{R}^n$ where A has full row rank. Let a point $x \in \mathbb{R}^n$ be given. Then it is easy to verify that

$$x - \Pi_{\mathcal{L}+b}(x) = A^T(AA^T)^{-1}A(x-b) \quad \text{and} \quad \|x - \Pi_{\mathcal{L}+b}(x)\|_2^2 = (x-b)^T A^T(AA^T)^{-1}A(x-b).$$

Likewise, for $s \in \mathbb{R}^n$ we have

$$s - \Pi_{\mathcal{L}^\perp+c}(s) = (I - A^T(AA^T)^{-1}A)(s-c)$$

and

$$\|s - \Pi_{\mathcal{L}^\perp+c}(s)\|_2^2 = (s-c)^T(I - A^T(AA^T)^{-1}A)(s-c).$$

The factorization of AA^T can be computed once in a preprocessing stage at the beginning of Algorithm 2 and can then be used without modification throughout. It is the same matrix that is usually factored in interior-point methods. For semidefinite programs, however, the factorization of $\mathcal{A}\mathcal{A}^*$ may be substantially cheaper than the systems factored at each iteration of an interior point algorithm; prime example being semidefinite programs arising from the semidefinite relaxation of the max clique problem that results in the factorization of dense matrices in interior point methods while $\mathcal{A}\mathcal{A}^*$ is a *diagonal* matrix.

After this preprocessing the projection of a point $z = (x, s)$ onto

$$\boldsymbol{A_1} := (\mathcal{L} + b) \times (\mathcal{L}^\perp + c)$$

can be computed (separately for x and s) in order mn operations, namely two back-solves using the factorization of AA^T and two matrix vector products Ax or As and some order-n-operations. Let

$$\boldsymbol{A_2} := \{(x,s) \mid \langle c, x\rangle + \langle b, s\rangle = \langle b, c\rangle\},$$

so that $\boldsymbol{A} = \boldsymbol{A_1} \cap \boldsymbol{A_2} \neq \emptyset$.

The following simple scheme computes the projection of a point z^0 onto $\boldsymbol{A}$:

Algorithm 3 (Projection onto $\boldsymbol{A}$)
Set $z = \Pi_{\boldsymbol{A_1}}(z^0)$, $\hat{z} := \Pi_{\boldsymbol{A_2}}(z)$, $\hat{\hat{z}} := \Pi_{\boldsymbol{A_1}}(\hat{z})$, *and* $z^+ := z + \lambda(\hat{\hat{z}} - z)$ *where* $\lambda > 0$ *is chosen such that* z^+ *lies in the Hyperplane* $\boldsymbol{A_2}$.

Claim: $z^+ = \Pi_{\boldsymbol{A}}(z^0)$.

Note that projecting back and forth between $\boldsymbol{A_1}$ and $\boldsymbol{A_2}$ will yield a sequence that converges to a point in $\boldsymbol{A_1} \cap \boldsymbol{A_2}$. The above algorithm abbreviates this process by extrapolating within $\boldsymbol{A_1}$ until the hyperplane $\boldsymbol{A_2}$ is hit at z^+.

Proof. We first show $z^+ = z^* := \Pi_{\boldsymbol{A}}(z)$. By construction, clearly, $z^+ \in \boldsymbol{A}$. (The assumption that there exists no $\lambda > 0$ defining the point z^+ readily leads to a contradiction.) Now, assume by contradiction that $\|x - x^*\|_2^2 < \|x - x^+\|_2^2$.

Define the segments $l_1 := [z, z^+]$ and $l_2 := [\hat{z}, z^+]$. Any point in l_1 is mapped to l_2 under $\Pi_{\boldsymbol{A_2}}$ since the linear map $\Pi_{\boldsymbol{A_2}}(z)$ maps the end points of l_1 to the end points of l_2. Likewise, any point in l_2 is mapped to l_1 under $\Pi_{\boldsymbol{A_1}}$ since the end points are mapped to $\hat{\hat{z}}$ and z^+ in l_1.

As $z - \hat{z}$ is perpendicular to $\boldsymbol{A_2}$ it follows $(z - \hat{z})^T(\hat{z} - z^*) = 0$ and $(z - \hat{z})^T(\hat{z} - z^+) = 0$. Hence, there is a $\rho > 0$ such that

$$\|z - \hat{z}\|_2^2 + \|\hat{z} - z^*\|_2^2 = \|z - z^*\|_2^2 \leq \|z - z^+\|_2^2 - \rho = \|z - \hat{z}\|_2^2 + \|\hat{z} - z^+\|_2^2 - \rho,$$

i.e. $\|\hat{z} - z^*\|_2^2 \leq \|\hat{z} - z^+\|_2^2 - \rho$.

Likewise $\|\hat{\hat{z}} - z^*\|_2^2 \leq \|\hat{\hat{z}} - z^+\|_2^2 - \rho$.

Repeating the process of projecting back and forth between $\boldsymbol{A_1}$ and $\boldsymbol{A_2}$ we see that the projected points are always closer to z^* than to z^+, and the squared difference is at least ρ. On the other hand these points remain in l_1 and l_2 and converge to z^+ which is a contradiction.

Finally, if $z^0 \notin \boldsymbol{A_1}$, it follows as in the above proof that we may use a "zero-th" step to project z^0 onto $\boldsymbol{A_1}$ and then start the above extrapolation from the projected point z. □

Note that for linear programs the computation of the projection $\Pi_{\boldsymbol{A_2}}(z)$ can be done in order n operations and is thus negligible. Hence, the projection $\Pi_{\boldsymbol{A}}(z)$ of z onto $\boldsymbol{A}$ by the above process takes four back-solves with the factorization of AA^T (or two back-solves if $z \in \boldsymbol{A_1}$) and the same number of matrix vector multiplications Ax or As.

In particular, each iteration of Algorithm 2 takes four back-solves with the factorization of AA^T and four matrix vector multiplications Ax or As.

Note that Remark 4 is also true when $\boldsymbol{A}$ is replaced with the larger subspace $\boldsymbol{A_1}$ and $\tilde{\phi}(\tilde{z})$ is replaced with $\phi(\tilde{z}) + \frac{1}{2}d(\tilde{z}, \boldsymbol{A_2})^2$ for $\tilde{z} \in \boldsymbol{A_1}$. The projections onto $\boldsymbol{A_1}$ cost half as much as the projections onto $\boldsymbol{A}$. Likewise, the accelerated local method for semidefinite programs can be applied for $Z \in \boldsymbol{A_1}$ and the function

$$\hat{f}(Z) + \frac{1}{2}d(Z, \boldsymbol{C})^2 + \frac{1}{2}d(Z, \boldsymbol{A_2})^2.$$

Numerical examples to compare both approaches are the subject of future research.

7 Numerical Results

Algorithm 2 has been implemented in Matlab and tested on some randomly generated linear semidefinite programs. Initially, the algorithm is always applied to minimizing the function $\tilde{\phi}$ for $z \in \boldsymbol{A}$, and when this minimization slows down, a Phase 2 is started, where a "regularizing" function $\hat{f}$ is added to $\tilde{\phi}$.

Under standard assumptions Lemma 1 garantees that the term $\|XS - SX\|_F^2$ may serve as a regularizing function. Note that also the term $\|XS + SX\|_F^2$ is minimized at the optimal solution of the problem (P). Thus, at the point Z^{opt} it has nonnegative curvature as well and hence, Lemma 1 also applies to the function

$$\|XS\|_F^2 = \frac{1}{4}(\|XS - SX\|_F^2 + \|XS + SX\|_F^2).$$

This term yielded the best numerical results in our examples, and the results listed below refer to this regularizing term – while Lemma 1 is proved under slightly weaker conditions (namely just the term $\|XS - SX\|_F^2$).

7.1 The intersection of two cones

The prime application targeted by the apd approach is the Lovasz relaxation of the max-clique problem for which the matrix $\mathcal{A}\mathcal{A}^*$ is a diagonal matrix, while interior point methods factor a at full matrix of the same size at each iteration. The Lovasz-Schrijver relaxation is a sharper relaxation for which the semidefinite cone is replaced with the intersection of the semidefinite cone and the cone of matrices with nonnegative entries. Unfortunately, while the projection onto either of the two cones is straightforward, the projection onto their intersection is less trivial. We therefore present an approach that allows the application to problems of the form

$$(\hat{P}) \qquad \text{minimize } \langle c, x\rangle \mid x \in K \cap \hat{K} \cap (\mathcal{L} + b),$$

where K and $\hat{K}$ are both pointed closed convex cones such that the interior of $K \cap \hat{K}$ is nonempty. Again, we assume that $\mathcal{L} + b$ is given by a set of linear equations $Ax = \bar{b}$ for which a factorization of AA^T is computed once, and that projections onto K and $\hat{K}$ are easy to compute.

Problem $(\hat{P})$ is equivalent to

$$\text{minimize } \langle \begin{pmatrix} c \\ 0 \end{pmatrix}, \begin{pmatrix} x \\ \hat{x} \end{pmatrix} \rangle \mid \begin{pmatrix} x \\ \hat{x} \end{pmatrix} \in (K \times \hat{K}) \cap (\hat{\mathcal{L}} + \begin{pmatrix} b \\ b \end{pmatrix}),$$

where

$$\hat{\mathcal{L}} + \begin{pmatrix} b \\ b \end{pmatrix} =: \left\{ \begin{pmatrix} x \\ \hat{x} \end{pmatrix} \mid Ax = \bar{b}, \quad x = \hat{x} \right\}.$$

This is a problem of the form (P). By our assumption, projections onto $K \times \hat{K}$ – and hence also projections onto its dual – are easy to compute. Thus, in order to apply the apd algorithm it suffices to verify that projections onto $\hat{\mathcal{L}}$ are easily computable given a factorization of AA^T.

This, however, is readily seen, as

$$\begin{pmatrix} A & 0 \\ I & -I \end{pmatrix} \begin{pmatrix} A^T & I \\ 0 & -I \end{pmatrix} = \begin{pmatrix} AA^T & A \\ A^T & 2I \end{pmatrix} = \begin{pmatrix} 2(AA^T)^{-1} & -(AA^T)^{-1}A \\ -A^T(AA^T)^{-1} & \frac{1}{2}(I + A^T(AA^T)^{-1}A) \end{pmatrix}^{-1}$$

provides the desired factorization.

7.2 Rescaling

We emphasize that Algorithm 2 is essentially a first order method, and hence, it is sensitive to scaling of the data. Even for data that "looks nice" (all data integers of absolute value less than 10) the following rescaling may turn out to be crucial:

First, replace b with $\Pi_{\mathcal{L}^\perp} b$. (The set $\mathcal{L} + b$ remains invariant with this change!) Likewise, replace c with $\Pi_{\mathcal{L}} c$. Then set $b = b/\|b\|_2$, $c = c/\|c\|_2$, and rescale x, s accordingly. Note that by this normalization, the duality simplifies to $\langle b, s\rangle + \langle c, x\rangle = 0$, and in particular, the set $\boldsymbol{A_2}$ now is a linear subspace.

Moreover, the origin $x = 0$ has distance exactly 1 from $\mathcal{L} + b$, and likewise $s = 0$ has distance exactly 1 from $\mathcal{L}^\perp + c$. For a semidefinite program, the point $x^{(0)} = s^{(0)} = I/\sqrt{n}$ is a canonical starting point: Its duality gap satisfies $\langle x^{(0)}, s^{(0)}\rangle = 1$, and the distance of $x^{(0)}$ from $\mathcal{L} + b$ is bounded by 2, same as the distance of $s^{(0)}$ from $\mathcal{L}^\perp + c$.

While the above rescaling of b and c appears to be natural, it is certainly far from optimal. When convergence slows down, it may be possible to identify a more suitable scaling based on the current iterate. The numerical results below simply refer to the above scaling.

7.3 Preconditioning

We point out that the above rescaling may be generalized slightly. Indeed, let M be a nonsingular matrix, then the preconditioning $X \to MXM^T$, $B \to MBM^T$, $\mathcal{L} \to M\mathcal{L}M^T$ and $S \to M^{-T}SM^{-1}$, $C \to M^{-T}CM^{-1}$ results in an equivalent semidefinite program, and the solution of either program can easily be recoverd from the solution of the other. Of course, the functions $\tilde{\phi}$ and $\hat{f}$ change when replacing X, S with $MXM^T, M^{-T}SM^{-1}$, and thus the performance of Algorithm 2 will vary. It is still an open question how to determine suitable scalings that accelerate Algorithm 2. When M is a diagonal matrix, the projections onto $M\mathcal{L}M^T$ and its orthogonal complement can be performed just as cheaply as for $\mathcal{L}$ and $\mathcal{L}^\perp$.

Likewise, one may look at preconditionings of the form $\|XS\|_F^2 \to \|MXS\tilde{M}\|_F^2$ for some nonsingular $M, \tilde{M}$. Here, the function $\tilde{\phi}$ is not changed, and here as well, the selection of suitable preconditionings is subject to further research.

7.4 Randomly generated SDP

To give some impression of the practical behaviour of our approach, we provide some computational results on randomly generated SDP. These instances are generated as follows. First we select semidefinite matrices X and S with $XS = 0$. The nonzero eigenvalues of the matrices X and S in Table 1 are uniformly distributed in $[0, 100]$ and in $[0, 10]$, respectively. The common eigenbasis is obtained from the orthogonalization of another random matrix. Then we generate the linear constraints by selecting matrices A_i having specified sparsity properties. In our case, we generate A_i to have nonzero support only on a submatrix of small order. This defines $b := A(X)$. We select dual variables y normally distributed. This gives $C = A^T(y) + S$. Thus, we have generated an instance with known optimal solution. We also provide the seed for the random number generator to make the data reproducible. The generator was written in MATLAB, and is accessible through http://www.math.uni-klu.ac.at/or/Software.

In the following table we provide some preliminary computational results. The parameters n and m indicate the size of the problem as defined before. The parameter 'seed' is used to initialize the random number generator and makes the instances reproducible using MATLAB. The column 'opt' contains the optimal value of the SDP. Then we provide the objective value

n	m	seed	opt	apd	error
400	30000	400303	-339098.8	-339091.4	-0.0004
400	40000	400403	-114933.8	-114931.1	-0.0002
500	40000	500403	571791.9	571801.9	-0.0005
500	50000	500503	-47361.2	-47353.4	-0.0003
600	40000	600403	97145.8	97186.6	-0.0016
600	50000	600503	-279848.9	-279810.6	-0.0012
600	60000	600603	489181.8	489194.5	-0.0004
700	50000	700503	-83535.4	-83488.7	-0.0014
700	70000	700703	-364458.8	-364476.1	-0.0004
800	80000	800803	-112872.6	-112817.4	-0.0011
1000	100000	1000013	191886.2	191954.5	-0.0012

Table 1: Randomly generated SDP. The column labeled 'apd' contains the function value after 50 iterations of our augmented primal-dual method. The most negative eigenvalue of X and S is displayed in the last column.

of our approach (in column 'apd') after 50 function evaluations. The last column gives the most negative eigenvalue of X and S which measures the error of our approach. We have used a 'quick-and-dirty' implementation of our approach, without any parameter tuning. The error of our approach is rudely estimated by the most negative eigenvalue of X and S. We note that in all these instances, the most negative eigenvalue (which keeps us away from feasibility) is close to 0, and indicates that our approach has a potential for problems where the contraints are sufficiently sparse (AA^T is manageable), and n is not too large, so that the projection onto the semidefinite cone is tractable.

The computations were done on a Pentium IV (2.1 Ghz, 2G memory) using Matlab. It took about 45 minutes for the largest instance, and a few minutes for the smallest one. Since this is a preliminary implementation, we expect that there is quite some room for improvement. The present paper sets the theoretical stage for the new approach. A competitive implementation is beyond the scope of the current paper and will be presented in a separate study.

8 Concluding remarks

This paper proposes a reformulation of a linear program over a convex cone into the problem of minimizing a differentiable convex apd-function in a certain primal-dual space. The apd-function is related to the augmented Lagrangian function, but is slightly less data dependent. For large classes of conic programs including linear, semidefinite and SOC problems, its function and gradient evaluations are rather cheap. For the case of a semidefinite program, a certain regularization of the apd-function is analyzed. Numerical examples minimizing the function with a conjugate gradient method illustrate the potential of the approach. Extensions for the case that Slaters condition is not satisfied and to other cones are subject of future research.

Acknowledgement

The authors acknowledge partial financial support by the Marie Curie Training Network Algorithmic Discrete Optimization (ADONET), MRTN-CT-2003-504438, financed by the European Union, and by the GICOLAG program at the International Erwin Schrödinger Institute for Mathematical Physics (ESI) Vienna, Austria.

References

[1] Bauschke, H.H, Combettes, P.L., and Kruk, S.G. (2006): Extrapolation algorithm for affine-convex feasibility problems, Numerical Algorithms 41, 239–274.

[2] S. Burer and R.D.C Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming (Series B)*, 95:329–357, 2003.

[3] Freund, R.W. and Jarre, F. (2004): A sensitivity result for semidefinite programs, Operations Research Letters 32(2), 126–132.

[4] C. Helmberg and F. Rendl. A spectral bundle method for semidefinite programming. *SIAM Journal on Optimization*, 10(3):673–696, 2000.

[5] M. Kocvara and M. Stingl. On the solution of large-scale sdp problems by the modified barrier method using iterative solvers. *Mathematical Programming*, 109:413–444, 2007.

[6] J. Malick, J. Povh, F. Rendl and A. Wiegele. Regularization methods for semidefinite programming, working paper, University of Klagenfurt (2007), in preparation.

[7] Nesterov, Y., and Nemirovskii, A. (1994): Interior-Point Polynomial Algorithms in Convex Programming, Society for Industrial and Applied Mathematics, Philadelphia.

[8] J. Povh, F. Rendl, and A. Wiegele. Boundary point method to solve semidefinite programs. *Computing*, 78:277–286, 2006.

[9] Qi, L., Sun, J. (1993): A nonsmooth version of Newton's method, Math. Prog. 58, 353–367.

[10] Sun, J., Sun, D. (2002): Semismooth matrix-valued functions, MOR 27: 150–169.

[11] K.C. Toh. Solving large scale semidefinite programs via an iterative solver on the augmented systems. *SIAM Journal on Optimization*, 14:670–698, 2004.