Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Structure exploitation in Semi-Definite Programs for Systems Analysis ⋆

Janne Harju Johansson * **Anders Hansson** **

* *Department of Electrical Engineering, Linköping University, 581 83 Linköping, Sweden (e-mail: harju@isy.liu.se).*
** *Department of Electrical Engineering, Linköping University, 581 83 Linköping, Sweden (e-mail: hansson@isy.liu.se).*

Abstract: A wide variety of problems involving analysis of systems can be rewritten as a semidefinite program. When solving these problems optimization algorithms are used. Large size makes the problems unsolvable in practice and computationally more effective solvers are needed. This paper investigates how to exploit structure and problem knowledge in some control applications. It is shown that inexact search directions are useful to reduce the computational burden and that operator formalism can be utilized to derive tailored calculations.

Keywords: Optimization; Linear Matrix Inequalities; Semidefinite programming; Interior-point methods; Iterative methods.

1. INTRODUCTION

In this paper semidefinite programming (SDP) for analysis of dynamical systems are discussed and solved. The problem formulation in the following section can be applied to analysis of polytopic linear differential inclusions (LDIs), see Boyd et al. [1994] and Gahinet et al. [1996] for details of the analysis methods.

There are many software packages for semidefinite programming that can be applied to the optimization problem described in this paper. Some examples are SDPT3, Toh et al. [2006] and SeDuMi, Sturm [2001] and Pólik [2005]. However, these solvers formulate the optimization problem on a general form. When the size of the problem increases, the solution time will be too large. Then the structural properties of a problem can be utilized to speed up the performance. In this paper knowledge of the optimization problem is taken into account and integrated in the algorithm. The suggested algorithm works with inexact search directions, which enables the use of an iterative solver for the linear equations that is terminated prior to convergence.

Using an iterative solver to find the search directions and the topic of preconditioning have been investigated in Vandenberghe and Boyd [1995], Keller et al. [2000], Bergamaschi et al. [2004], Rozloznik and Simoncini [2003] and Bonettini and Ruggiero [2007]. Inexact search directions have been applied for a potential reduction method in Vandenberghe and Boyd [1995] and Gillberg and Hansson [2003], and for quadratic problems using a potential reduction algorithm in Cafieri et al. [2007]. In the work of Gillberg and Hansson [2003] a feasible interior-point method was used, and the search directions were computed from the normal equations. Every iteration in the algorithm required a feasible point and hence an expensive projection was needed since the iterative equation solver does not guarantee a feasible search direction. In Vandenberghe and Boyd [1995] this was circumvented by solving one linear system of equations for the primal search direction and another linear system of equations for the dual search direction, however also at a high computational cost. Additionally, solving the normal equations in Gillberg and Hansson [2003] resulted in an increasing number of iterations in the iterative solver when tending towards the optimum. In this paper the augmented equations are solved, which results in an indefinite linear system of equations, and no increase in the number of iterations close to the optimum has been observed. Similar behavior has been observed in Hansson [2000] and Cafieri et al. [2007]. Since an infeasible interior-point method is used, no projection is needed in order to preserve feasibility.

The remaining part of the paper is organized as follows. First the optimization problem is formulated and some mathematical preliminaries are presented. Then a discussion of interior-point methods is presented which results in a detailed description of an algorithm that uses inexact search directions obtained from an iterative solver. In such a solver a preconditioner is needed for fast convergence and hence a tailored preconditioner is presented in detail. Finally, some computational results and conclusions are presented.

2. PROBLEM FORMULATION

The optimization problem in the variables $P \in \mathbb{S}^n$ and $x \in \mathbb{R}^{n_x}$ that is to be solved is

$$\begin{aligned} &\min\ c^T x + \langle C, P\rangle \\ &\text{s.t. } \mathcal{F}_i(P) + \mathcal{G}_i(x) + M_{i,0} = S_i,\ i = 1,\dots,n_i \\ &S_i \succeq 0 \end{aligned} \tag{1}$$

where

$$\mathcal{F}_i(P) = \begin{bmatrix} \mathcal{L}_i(P) & PB_i \\ B_i^T P & 0 \end{bmatrix} = \begin{bmatrix} A_i^T P + PA_i & PB_i \\ B_i^T P & 0 \end{bmatrix} \tag{2}$$

and

$$\mathcal{G}_i(x) = \sum_{k=1}^{n_x} x_k M_{i,k}, \tag{3}$$

⋆ This work was supported by CENIIT.

with $A_i \in \mathbb{R}^{n\times n}$, $B_i \in \mathbb{R}^{n\times m}$, $C \in \mathbb{S}^n$ and $M_{i,k} \in \mathbb{S}^{n+m}$. The inner product $\langle C, P\rangle$ is Trace(CP), and $\mathcal{L}_i : \mathbb{S}^n \to \mathbb{S}^n$ is the Lyapunov operator with adjoint $\mathcal{L}_i^*$.

Furthermore the adjoint operators of $\mathcal{F}$ and $\mathcal{G}$ are

$$\mathcal{F}_i^*(Z_i) = [A_i \ B_i]\, Z_i \begin{bmatrix} I_n \\ 0 \end{bmatrix} + [I_n \ 0]\, Z_i \begin{bmatrix} A_i^T \\ B_i^T \end{bmatrix} \tag{4}$$

and

$$\mathcal{G}_i^*(Z_i)_k = \langle M_{i,k}, Z_i\rangle,\ k = 1, \ldots, n_x \tag{5}$$

respectively, where $Z_i \in \mathbb{S}^{n+m}$.

When it is possible to study (1) on a higher level of abstraction the operator $\mathcal{A}(P,x) = \oplus_{i=1}^{n_i}(\mathcal{F}_i(P) + \mathcal{G}_i(x))$ is used. Its adjoint is $\mathcal{A}^*(Z) = (\mathcal{F}^*(Z), \mathcal{G}^*(Z)) = \sum_{i=1}^{n_i}(\mathcal{F}_i^*(Z_i), \mathcal{G}_i^*(Z_i))$ where $Z = \oplus_{i=1}^{n_i} Z_i$. Also define $S = \oplus_{i=1}^{n_i} S_i$ and $M_0 = \oplus_{i=1}^{n_i} M_{i,0}$.

For later use we define $z = (x, P, S, Z)$ and the corresponding finite-dimensional vector space $\mathcal{Z} = \mathbb{R}^{n_x} \times \mathbb{S}^{n+m} \times \mathbb{S}^{n+m} \times \mathbb{S}^{n+m}$ with its inner product $\langle \cdot, \cdot \rangle_{\mathcal{Z}}$.

Throughout the paper it is assumed that the mapping $\mathcal{A}$ has full rank. We remark that the optimization problem considered is not within the class of problems considered in e.g. Wallin and Hansson [2004] and Gillberg and Hansson [2003].

3. INEXACT INTERIOR-POINT METHOD

3.1 Introduction

For a thorough description of algorithms and theory within the area of interior-point methods see Wright [1997] for linear programming, while Wolkowicz et al. [2000] gives an extensive overview of semidefinite programming. In Boyd and Vandenberghe [2004] a thorough presentation of the main ideas in interior-point methods is given. Here follows a brief discussion on what optimality conditions that are to be fulfilled and how these are relaxed for use in an infeasible interior-point method. Then an inexact infeasible method for semidefinite programming is presented.

3.2 Optimality conditions

In this work a primal-dual interior-point method is implemented. For such algorithms the primal and dual problems are solved simultaneously. The primal and dual for (1) with the higher level of notation are

$$\begin{aligned} \min\ & c^T x + \langle C, P\rangle \\ \text{s.t}\ & \mathcal{A}(P,x) + M_0 = S \\ & S \succeq 0 \end{aligned} \tag{6}$$

$$\begin{aligned} \max\ & -\langle M_0, Z\rangle \\ \text{s.t}\ & \mathcal{A}^*(Z) = (C,c) \\ & Z \succeq 0 \end{aligned} \tag{7}$$

If strong duality holds then the Karush-Kuhn-Tucker conditions defines the solution to the primal and dual optimization problems, Boyd and Vandenberghe [2004]. The Karush-Kuhn-Tucker conditions for the optimization problems in (6) and (7) are

$$\mathcal{A}(P,x) + M_0 = S \tag{8}$$
$$\mathcal{A}^*(Z) = (C,c) \tag{9}$$
$$ZS = 0 \tag{10}$$
$$S \succeq 0,\ Z \succeq 0 \tag{11}$$

Definition The complementary slackness ν is defined as

$$\nu = \frac{\langle Z, S\rangle}{n} \tag{12}$$

Definition Define the central-path as the solution points for

$$\mathcal{A}(P,x) + M_0 = S \tag{13}$$
$$\mathcal{A}^*(Z) = (C,c) \tag{14}$$
$$ZS = \nu I \tag{15}$$
$$S \succeq 0,\ Z \succeq 0 \tag{16}$$

where $\nu \geq 0$. Note that the central-path converges to a solution of the Karush-Kuhn-Tucker conditions when ν tends to zero.

3.3 Infeasible interior-point method

In this section an infeasible interior-point method is discussed. Such a method is initiated with an infeasible or a feasible point $z \in \mathcal{Z}$ and then its iterates tend toward feasibility and optimality by computing a sequence of search directions and taking steps in these directions. To derive equations for the search directions the next iterate $z^+ = z + \Delta z$ is introduced and inserted into (13)-(16). This gives a nonlinear system of equations for Δz. Even after linearization the variables ΔS and ΔZ of the solution Δz to these equations are not guaranteed to be symmetric since this requirement is only implicit. A solution to this remedy is to introduce the symmetry transformation $\mathcal{H} : \mathbb{R}^{n\times n} \to \mathbb{S}^n$ that is defined by

$$\mathcal{H}(X) = \frac{1}{2}\big(R^{-1}XR + (R^{-1}XR)^T\big) \tag{17}$$

where $R \in \mathbb{R}^{n\times n}$ is the so called scaling matrix. For a thorough description of scaling matrices, see Wolkowicz et al. [2000] and Zhang [1998]. In Zhang [1998] it is shown that the relaxed complementary slackness condition $ZS = \nu I$ is equivalent with

$$\mathcal{H}(ZS) = \nu I \tag{18}$$

for any nonsingular matrix R. Hence we may replace (15) with (18). Replacing z with the next iterate $z + \Delta z$ in (13), (14), (16) and (18) results in

$$\mathcal{A}(\Delta P, \Delta x) - \Delta S = -(\mathcal{A}(P,x) + M_0 - S) \tag{19}$$
$$\mathcal{A}^*(\Delta Z) = (C,c) - \mathcal{A}^*(Z) \tag{20}$$
$$\mathcal{H}(\Delta Z S + Z \Delta S) = \nu I - \mathcal{H}(ZS) - \mathcal{H}(\Delta Z \Delta S) \tag{21}$$
$$S + \Delta S \succeq 0,\ Z + \Delta Z \succeq 0 \tag{22}$$

If the nonlinear term in (21) is ignored, ΔS and ΔZ will be symmetric. Several approaches to handling the nonlinear term in the complementary slackness equation have been presented in the literature. A direct solution is to ignore the higher order term which gives a linear system of equations. Another approach is presented in Mehrotra [1992].

3.4 Inexact predictor-corrector method

Here an inexact infeasible predictor-corrector method is presented. The idea of using inexact search directions has been applied to monotone variational inequality problems in Ralph and Wright [1997] and to model predictive control applications in Hansson [2000].

In a predictor-corrector method alternating steps are taken. There are two separate objectives in the strategy.

The predictor step decreases the duality gap while the corrector step moves the iterate towards the central-path. To this end the parameter ν in (21) is replaced with $\sigma\nu$. Then for small values of σ a step is taken to reduce the complementary slackness ν, and for values of σ close to 1 a step to find an iterate close to the central path is taken. Then the linear system of equations to be solved for the search directions is

$$\mathcal{A}(\Delta P, \Delta x) - \Delta S = -(\mathcal{A}(P,x) + M_0 - S) \tag{23}$$
$$\mathcal{A}^*(\Delta Z) = (C,c) - \mathcal{A}^*(Z) \tag{24}$$
$$\mathcal{H}(\Delta Z S + Z\Delta S) = \sigma\nu I - \mathcal{H}(ZS) \tag{25}$$

Lemma 1. If the operator $\mathcal{A}$ has full rank, i.e. $\mathcal{A}(x) = 0$ implies that $x = 0$, and if $Z \succ 0$ and $S \succ 0$, then the linear system of equations in (23)-(25) has a unique solution.

Proof See Theorem 10.2.2 in Wolkowicz et al. [2000].

For later use and to obtain an easier notation define $\mathcal{K} : \mathcal{Z} \to \mathbb{S}^n \times \mathbb{S}^n$ as

$$\mathcal{K}(z) = \begin{bmatrix} \mathcal{K}_p(z) \\ \mathcal{K}_d(z) \\ \mathcal{K}_c(z) \end{bmatrix} = \begin{bmatrix} \mathcal{A}(P,x) + M_0 - S \\ \mathcal{A}^*(Z) - (C,c) \\ \mathcal{H}(ZS) \end{bmatrix} \tag{26}$$

Now define the set Ω as

$$\begin{aligned} \Omega = \{z = (x,S,Z) \mid\ & S \succeq 0, Z \succeq 0, \\ & \|\mathcal{K}_p(z)\|_2 \le \beta\nu, \|\mathcal{K}_d(z)\|_2 \le \beta\nu, \\ & \gamma\nu I \succeq \mathcal{K}_c(z) \succeq \eta\nu I\} \end{aligned} \tag{27}$$

where the scalars β, γ and η will be defined later on. Then define the set Ω_+ as

$$\Omega_+ = \{z \in \Omega \mid S \succ 0, Z \succ 0\} \tag{28}$$

Finally define the set $\mathcal{S}$ for which the Karush-Kuhn-Tucker conditions (8)-(11) are fulfilled.

$$\mathcal{S} = \{z \mid \mathcal{K}_p(z) = 0,\ \mathcal{K}_d(z) = 0,\ \mathcal{K}_c(z) = 0,\ S \succeq 0,\ Z \succeq 0\} \tag{29}$$

Below the overall algorithm is summarized, which is taken from Ralph and Wright [1997], and adapted to semidefinite programming.

Algorithm: Inexact primal-dual method

0. Initialize the counter $j = 1$ and choose $0 < \eta < \eta_{max} < 1$, $\gamma \ge n$, $\beta > 0$, $\kappa \in (0,1)$, $0 < \sigma_{min} < \sigma_{max} < 1/2$, $\epsilon > 0$, $0 < \chi < 1$ and $z^0 \in \Omega$.
1. Evaluate stopping criteria. If fulfilled, terminate the algorithm.
2. Choose $\sigma \in (\sigma_{min}, \sigma_{max})$.
3. Compute the scaling matrix R.
4. Solve (23)-(25) for search direction Δz^j with a residual tolerance $\epsilon\sigma\beta\nu/2$.
5. Choose a step length α^j as the first element in the sequence $\{1, \chi, \chi^2, \ldots\}$ such that $z^{j+1} = z^j + \alpha^j \Delta z^j \in \Omega$ and such that $\nu^{j+1} \le \big(1 - \alpha\kappa(1-\sigma)\big)\nu^j$.
6. Update the variables, $z^{j+1} = z^j + \alpha^j \Delta z^j$ and the counter $j := j + 1$.
7. Return to step 1.

Note that any iterate generated by the algorithm is in Ω, which is a closed set, since it is defined as an intersection of closed sets, see Harju and Hansson [2007].

3.5 Convergence

For a detailed description and a convergence proof, see Harju and Hansson [2007].

4. SEARCH DIRECTIONS

Solving (23)-(25) is in practice done by either one of two approaches. The first eliminates both the slack variables S_i and the dual variables Z_i resulting in a positive definite linear system of equations called the normal equations. In the second approach the slack variables S_i are eliminated which results in an indefinite linear system of equations called the augmented equations. In this paper the augmented equations are solved. For a thorough discussion of general saddle point problems, see Benzi et al. [2005].

To eliminate the S variable, (25) is solved for S and inserted into (23). The details in these calculations are presented on page 34 in Vandenberghe et al. [2005]. Now study each constraint separately and define the resulting linear system of equations as

$$W_i \Delta Z_i W_i + \mathcal{F}_i(\Delta P) + \mathcal{G}_i(\Delta x) = D_{1,i}, \ \ \forall i \tag{30}$$
$$\sum_{i=1}^{n_i} \mathcal{F}_i^*(\Delta Z_i) = D_2 \tag{31}$$
$$\sum_{i=1}^{n_i} \mathcal{G}_i^*(\Delta Z_i) = D_3 \tag{32}$$

where $W_i = R_i R_i^T \in \mathbb{S}^n$ denotes the scaling matrices for the interior point method. Note that the linear system of equations in (30)-(32) is indefinite.

4.1 Iterative solver

When solving linear equations with an iterative solver there is a large number of algorithms to choose from. The choice of an applicable iterative method is based on the properties of the linear system of equations and possibly also on the choice of preconditioner. Definite/indefinite coefficient matrix or hermitian/non-hermitian coefficient matrix are the two main properties to consider.

Available solvers are for example the minimal residual (MINRES) method and the widely used conjugate gradient (CG) method. Here an indefinite system with an indefinite preconditioner is studied and hence algorithms for indefinite systems are the main focus. Examples of iterative solvers that handle indefinite matrices are conjugate gradient stabilized (CGS) method, the bi-conjugate gradient method and its stabilized version (BiCG and BiCGstab), the quasi minimal residual (QMR) method, and various versions of the generalized minimal residual (GMRES) method. In Greenbaum [1997] the algorithms are explained and studied in detail and in Barrett et al. [1994] the implementational details are discussed.

In this work the symmetric quasi-minimal residual method (SQMR) is chosen. The algorithm utilizes the fact that a symmetric coefficient matrix is available and an indefinite preconditioner can be used. It does not require as much storage as in GMRES which is a desired property since the intention is to solve large systems of equations. One undesired property is that the residual is not included in the algorithm and hence it must be calculated if a guaranteed residual is required.

In order to simplify the description rewrite (30)-(32) as

$$\mathcal{B}(\Delta z) = b \tag{33}$$

The described algorithm is SQMR without look-ahead for the linear system of equations on operator formalism. Denote the invertible preconditioner $\mathcal{P}(\Delta z) = p$. In Freund and Nachtigal [1991] and Freund and Nachtigal [1994] the original algorithm description is presented.

Algorithm: SQMR

0.) Choose $\Delta z_0 \in \mathcal{Z}$. Then set $r_0 = b - \mathcal{B}(z_0)$, $t = r_0$, $\tau_0 = \|t\|_2 = \sqrt{\langle r_0, r_0\rangle}$, $q_0 = \mathcal{P}^{-1}(r_0)$, $\vartheta_0 = 0$, $\rho_0 = \langle r_0, q_0\rangle$, and $d_0 = \mathbf{0}$.

For $j = 1, 2, \ldots$

1.) Compute $t = \mathcal{B}(q_{j-1})$, $v_{j-1} = \langle q_{j-1}, t\rangle$.
if $v_{j-1} = 0$, **then** Terminate
else
$\alpha_{j-1} = \frac{\rho_{j-1}}{v_{j-1}}$ and $r_j = r_{j-1} - \alpha_{j-1} t$
end

2.) Set $t = r_j$, $\vartheta_j = \|t\|_2/\tau_{j-1}$, $c_j = 1/\sqrt{1+\vartheta_j^2}$, $\tau_j = \tau_{j-1}\vartheta_j c_j$, $d_j = c_j^2\vartheta_{j-1}^2 d_{j-1} + c_j^2\alpha_{j-1}q_{j-1}$ and $\Delta z_j = \Delta z_{j-1} + d_j$.
if Δz_j has converged, **then** Terminate
end

3.) **if** $\rho_{j-1} = 0$, **then** Terminate
else
$u_j = \mathcal{P}^{-1}(t)$, $\rho_j = \langle r_j, u_j\rangle$, $\beta_j = \frac{\rho_j}{\rho_{j-1}}$, and $q_j = u_j + \beta_j q_{j-1}$.

Here b, p, r, t, q, $d \in \mathcal{Z}$ and τ, ϑ, ρ, v, α, $c \in \mathbb{R}$.

4.2 Preconditioner

The convergence of iterative algorithms is closely related to the condition number of the linear system of equations. Unfortunately, it is very unlikely that the condition number is small. Then a preconditioner that decreases the condition number is desirable. Good properties for a preconditioner is that it is an approximation of the linear system of equations (33) that is to be solved, and that the solution time for a preconditioner is preferably much lower than for the original system of equations, since it is called once in every iteration in the iterative solver. In Greenbaum [1997], Barrett et al. [1994] and Benzi et al. [2005] general preconditioners are described and discussed. In general preconditioners the vectorized form of the equations are used. To exploit structure, operator formalism is used, when possible, in this work. This was used also in Vandenberghe and Boyd [1995] for applications to systems analysis.

In Oliveira and Sorensen [2005] it is shown that every preconditioner for the normal equations system has an equivalent for the augmented system. It is also shown that the opposite is not true and hence it may be beneficial to search for preconditioners for the augmented equations.

To derive a preconditioner, assume that the constraints are closely related and therefore an approximation assuming $A_i = \bar{A}$, $B_i = \bar{B}$ and $M_{i,k} = \bar{M}_k$ is applicable. Inserting the average system matrices $\bar{A}$, $\bar{B}$ and $\bar{M}_k$ into (30)-(32) results in the following equations

$$W_i\Delta Z_i W_i + \bar{\mathcal{F}}(\Delta P) + \bar{\mathcal{G}}(\Delta x) = D_{1,i},\ \forall i \tag{34}$$

$$\bar{\mathcal{F}}^*(\sum_{i=1}^{n_i}\Delta Z_i) = D_2 \tag{35}$$

$$\bar{\mathcal{G}}^*(\sum_{i=1}^{n_i}\Delta Z_i) = D_3 \tag{36}$$

Approximate W_i with $w_i \cdot I$ and denote $w_\Sigma = \sum_{i=1}^{n_i} 1/w_i^2$. Now rescale the equations and define the new variables $\Delta Z_{tot} = \sum_i \Delta Z_i$, $\Delta P_\Sigma = \Delta P \cdot w_\Sigma$ and $\Delta x_\Sigma = \Delta x \cdot w_\Sigma$. The simplified linear system of equations that is to be solved by the preconditioner is

$$\Delta Z_{tot} + \bar{\mathcal{F}}(\Delta P_\Sigma) + \bar{\mathcal{G}}(\Delta x_\Sigma) = \sum_{i=1}^{n_i}\frac{D_{1,i}}{w_i^2} \tag{37}$$

$$\bar{\mathcal{F}}^*(\Delta Z_{tot}) = D_2 \tag{38}$$

$$\bar{\mathcal{G}}^*(\Delta Z_{tot}) = D_3 \tag{39}$$

Now define the block structure

$$\Delta Z_{tot} = \begin{pmatrix}\Delta Z_{11} & \Delta Z_{12}\\ \Delta Z_{12}^T & \Delta Z_{22}\end{pmatrix} \tag{40}$$

To derive a method for solving (37)-(39) we use the following change of variables

$$\Delta\tilde{Z}_{11} = \Delta Z_{11} + \mathcal{L}^{-*}\big(\bar{B}\Delta Z_{12}^T + \Delta Z_{12}\bar{B}^T\big) \tag{41}$$

$$\Delta\tilde{Z}_{12} = \Delta Z_{12} \tag{42}$$

$$\Delta\tilde{Z}_{22} = \Delta Z_{22} \tag{43}$$

$$\Delta\tilde{P} = \Delta P + \mathcal{L}^{-1}\big(\sum_{k=1}^{n_x}\Delta x_k\bar{M}_{k,11}\big) \tag{44}$$

$$\Delta\tilde{x} = \Delta x \tag{45}$$

where $\bar{M}_{k,11}$ denotes the (1,1) block of the $\bar{M}_j$ matrices and $\mathcal{L}$ is the Lyapunov operator using $\bar{A}$.

Now apply $\mathcal{L}^{-1}(\cdot)\bar{B}$ to the (1,1) block of (37) and subtract it from the (1,2) block in (37). Additionally apply $\langle\mathcal{L}^{-*}(\cdot), M_{k,11}\rangle$ to (38) and subtract it from the k:th element in (39). Using the variable change defined in (41)-(45) results in the following linear system of equations

$$\Delta\tilde{Z}_{11} - \mathcal{L}^{-*}(\bar{B}\Delta\tilde{Z}_{12}^T + \Delta\tilde{Z}_{12}\bar{B}^T) + \mathcal{L}\big(\Delta\tilde{P}\big) = \sum_{i=1}^{n_i}\frac{D_{1,11}^i}{w_i^2} \tag{46}$$

$$\begin{aligned}\Delta\tilde{Z}_{12} + \mathcal{L}^{-1}(\mathcal{L}^{-*}(\bar{B}\Delta\tilde{Z}_{12}^T + \Delta\tilde{Z}_{12}\bar{B}^T))\bar{B} -\\ -\mathcal{L}^{-1}(\Delta\tilde{Z}_{11})\bar{B} + \sum_{k=1}^{n_x}x_k\big(\bar{M}_{k,12} - \mathcal{L}^{-1}(\bar{M}_{k,11})\bar{B}\big) =\\ = \sum_{i=1}^{n_i}\frac{D_{1,12}^i}{w_i^2} - \mathcal{L}^{-1}\big(\sum_{i=1}^{n_i}\frac{D_{1,11}^i}{w_i^2}\big)\bar{B}\end{aligned} \tag{47}$$

$$\Delta\tilde{Z}_{22} + \sum_{k=1}^{n_x}\Delta x_j\bar{M}_{k,22} = \sum_{i=1}^{n_i}\frac{D_{1,22}}{w_i^2} \tag{48}$$

$$\mathcal{L}^*\big(\Delta\tilde{Z}_{11}\big) = D_2 \tag{49}$$

$$\begin{aligned}\left\langle\begin{bmatrix}-\mathcal{L}^{-*}\big(\bar{B}\Delta\tilde{Z}_{12}^T + \Delta\tilde{Z}_{12}\bar{B}^T\big) & \Delta\tilde{Z}_{12}\\ \Delta\tilde{Z}_{12}^T & \Delta\tilde{Z}_{22}\end{bmatrix}, \begin{bmatrix}\bar{M}_k^{11} & \bar{M}_k^{12}\\ \bar{M}_k^{12^T} & \bar{M}_k^{22}\end{bmatrix}\right\rangle =\\ = D_{3,k} - \langle\mathcal{L}^{-*}\big(D_2\big), \bar{M}_{k,11}\rangle\end{aligned} \tag{50}$$

The resulting linear system of equations can be solved in a five step procedure.

Algorithm: Preconditioner

1. Solve (49) for $\Delta\tilde{Z}_{11}$ using a Lyapunov solver.
2. Vectorize and solve (47), (48) and (50) to retrieve $\Delta\tilde{Z}_{12}$, $\Delta\tilde{Z}_{22}$ and Δx. The vectorized linear system of equations has $(nm+m^2)+(m(m+1)/2)+n_x$ variables and can thus be solved in $\mathcal{O}(n^3)$ flops (assuming $n \gg m$ and $n \gg n_x$).
3. Solve (46) using a Lyapunov solver to obtain $\Delta\tilde{P}$.
4. Find the untransformed variables ΔZ_{tot}, ΔP and Δx.
5. Distribute the solution such that $\Delta Z_i = (D_{1,i} - \mathcal{F}_i(\Delta P) - \mathcal{G}_i(\Delta x))/w_i^2$.

Note that the coefficient matrix for the vectorized system of equations in step 2 in the algorithm needs only to be constructed once. The main cost is the solution of symmetric Lyapunov equations. This can be done at a cost of $\mathcal{O}(n^3)$.

It is noted that the assumptions made to derive the preconditioner is not guaranteed to be fulfilled for a general problem. However for stability analysis described in the introduction the system matrices are often closely related. It is obvious that if these assumptions are violated the convergence speed will deteriorate for the iterative solver.

4.3 Initialization

For comparable results the initialization scheme given in Toh et al. [2006] is used for the dual variables,

$$Z_i = \max\left(10, \sqrt{n+m}, \max_{k=1,\ldots,n_x} \frac{(n+m)(1+|c_k|)}{1+\|M_{i,k}\|_F}\right) \cdot I_{n+m} \tag{51}$$

where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. The slack variables are chosen as $S_i = Z_i$ while the primal variables are initialized as $P = I_n$ and $x = \bar{0}$.

5. COMPUTATIONAL RESULTS

The computer used for numerical evaluation is a Dell Optiplex GX620 with 2GB RAM, Intel P4 640 (3,2GHz) CPU with Linux running under CentOS 4.1. SDPT3 version 3.1 is used as underlying solver and it is interfaced via YALMIP version 3 (R20070810), Löfberg [2004]. Since the intention is to solve large optimization problems the tolerance for termination is set to 10^{-3} for the relative and absolute residual. Matlab version 7.4 (R2007a) is used.

Note that the implemented inexact primal-dual interior-point method is written in Matlab while the equations solver in SDPT3 is written in C which gives faster function executions for SDPT3.

The parameters in the algorithm are set to $\kappa = 10^{-7}$, $\sigma_{max} = 0.0035$, $\sigma_{min} = 0.001$, $\eta = 10^{-6}$, $\chi = 0.9$, $\epsilon = 5 \cdot 10^{-7}$ and $\beta = 10^8 \cdot \beta_{lim}$ where $\beta_{lim} = \max(\|\mathcal{K}_p(z_0)\|_2, \|\mathcal{K}_d(z_0)\|_2)$. The choice of parameter values are based on knowledge obtained during the development of the algorithm.

In order to avoid expensive calculations in the iterative solver, it is terminated after 100 iterations. This can be interpreted as a temporary choice of ϵ in that iteration.

5.1 Examples

To evaluate the suggested algorithm, randomly generated optimization problems are solved. This is done as follows. First $\bar{A}$, $\bar{B}$ and $\bar{M}_k$ are generated by `gallery.m`. This Matlab function generates random matrices with a desired condition number. For the examples in this section every system matrix has a condition number of 10. The number of constraints n_i is chosen to two and n_x is one. Every constraint has separate system matrices A_i, B_i and $M_{i,k}$ that are generated as $A_i = \bar{A} \pm 0.01 \cdot \delta_A$, $B_i = \bar{B} \pm 0.01 \cdot \delta_B$ and $M_{i,k} = \bar{M}_{i,k} \pm 0.01 \cdot \delta_{M_{i,k}}$. The matrix δ_A is a diagonal matrix where the diagonal is generated by `rand.m` while δ_B and $\delta_{M_{i,k}}$ are generated by `gallery.m`. c, and C are chosen to give a feasible optimization problem.

In the preconditioner the mean system matrices $\bar{A}$, $\bar{B}$ and $\bar{M}_k$ are available as an approximation of the system matrices A_i, B_i, and $M_{i,k}$.

To get the solution times the Matlab command `cputime` is used. Input to the solvers are the system matrices so any existing preprocessing of the problem is included in the total solution time.

The residual for the search directions is calculated in each iteration in the iterative solver. To further improve the solution times, the in SQMR available Biconjugate Gradient (BCG) residual should be used instead. Note that the BCG residual r_j is given in the algorithm for SQMR described in Section 4.1.

5.2 Results

For each system order, ten randomly generated problems were solved and the mean times are presented in Figure 1. It is clear that the in Matlab written solver using inexact search directions calculated by an iterative solver based on operator formalism is faster in absolute time for large values of the system order n.

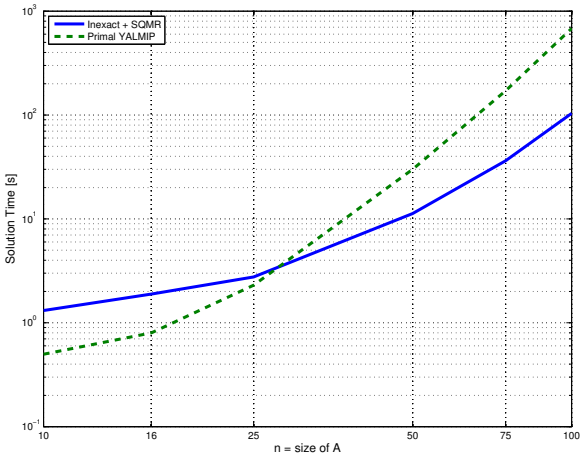

Fig. 1. Solution times for randomly generated optimization problems vs system order. Used solvers are the primal problem solved by SDPT3 and a solver based on the algorithm described in this paper.

6. CONCLUSION

An inexact primal-dual interior-point method has been presented and evaluated against a state of the art solver for semidefinite programming. The results show that solution times for optimization problems can be reduced by solving the linear system of equations for the search directions inexact. Structure exploitation was made by using operator formalism in a tailored preconditioner for the iterative solver.

REFERENCES

R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. M. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. V. der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods.* Philadalphia: Society for Industrial and Applied Mathematics., 1994.

M. Benzi, G. H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta numerica*, 14:1 – 137, 2005.

L. Bergamaschi, J. Gondzio, and G. Zilli. Preconditioning indefinite systems in interior point methods for optimization. *Computational Optimization and Applications*, 28(2):149 – 171, 2004.

S. Bonettini and V. Ruggiero. Some iterative methods for the solution of a symmetric indefinite KKT system. *Computational Optimization and Applications*, 38(1):3 – 25, 2007.

S. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge University Press, 2004.

S. Boyd, E. G. Laurent, E. Feron, and V. Balakrishnan. *Linear Matrix inequalities in System and Control Theory.* SIAM, 1994.

S. Cafieri, M. D'Apuzzo, V. De Simone, and D. di Serafino. On the iterative solution of KKT systems in potential reduction software for large-scale quadratic problems. *Computational Optimization and Applications*, 38(1):27 – 45, 2007.

R. W. Freund and N. M. Nachtigal. QMR: a quasi-minimal residual method for non-hermitian linear systems. *Numerische Mathematik*, 60(3):315 – 339, 1991.

R. W. Freund and N. M. Nachtigal. A new Krylov-subspace method for symmetric indefinite linear systems. In *Proceedings of the 14th IMACS World Congress on Computational and Applied Mathematics*, pages 1253–1256. IMACS, 1994.

P. Gahinet, P. Apkarian, and M. Chilali. Parameter-dependent Lyapunov functions for real parametric uncertainty. *IEEE Transactions on Automatic Control*, 41 (3):436 – 442, 1996.

J. Gillberg and A. Hansson. Polynomial complexity for a Nesterov-Todd potential-reduction method with inexact search directions. In *Proceedings of thet 42nd IEEE Conference on Decision and Control*, page 6, Maui, Hawaii, USA, December 2003.

A Greenbaum. *Iterative methods for solving linear systems.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997.

A. Hansson. A primal-dual interior-point method for robust optimal control of linear discrete-time systems. *Automatic Control, IEEE Transactions on*, 45(9):1639–1655, 2000.

J. Harju and A. Hansson. An inexact interior-point method, a description and convergence proof. Technical Report LiTH-ISY-R-2819, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden, September 2007.

C. Keller, N. I. M. Gould, and A. J. Wathen. Constraint preconditioning for indefinite linear systems. *SIAM Journal on Matrix Analysis and Applications*, 21(4): 1300 – 1317, 2000.

J. Löfberg. YALMIP : A toolbox for modeling and optimization in Matlab. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

S. Mehrotra. On the implementation of a primal-dual interior point method. *SIAM Journal on Optimization*, 2(4):575–601, 1992.

A. R. L. Oliveira and D. C. Sorensen. A new class of preconditioners for large-scale linear systems from interior point methods for linear programming. *Linear algebra and its applications*, 394:1 – 24, 2005.

I. Pólik. Addendum to the SeDuMi user guide, version 1.1, 2005.

D. Ralph and S. J. Wright. Superlinear convergence of an interior-point method for monotone variational inequalities. *Complementarity and Variational Problems: State of the Art*, 1997.

M. Rozloznik and V. Simoncini. Krylov subspace methods for saddle point problems with indefinite preconditioning. *SIAM journal on matrix analysis and applications*, 24(2):368 – 391, 2003.

J. F. Sturm. Using SeDuMi 1.02, a matlab toolbox for optimization over symmetric cones, 2001.

K. C. Toh, M. J. Todd, and R. Tütüncü. On the implementation and usage of SDPT3 — a Matlab software package for semidefinite-quadratic-linear programming, version 4.0. 2006.

L Vandenberghe and S. Boyd. A primal-dual potential reduction method for problems involving matrix inequalities. *Mathematical Programming*, 69:205 – 236, 1995.

L. Vandenberghe, V. R. Balakrishnan, R. Wallin, A. Hansson, and T. Roh. *Interior-point algorithms for semidefinite programming problems derived from the KYP lemma*, volume 312 of *Lecture notes in control and information sciences.* Springer, Feb 2005.

R. Wallin and A. Hansson. KYPD: A solver for semidefinite programs derived from the Kalman-Yakubovich-Popov lemma. In *IEEE Conference on Computer Aided Control Systems Design*, Taipei, Taiwan, September 2004. IEEE.

H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors. *Handbook of Semidefinite Programming: Theory, Algorithms, and Applications*, volume 27 of *International series in operations research & management science.* KLUWER, 2000.

S. J. Wright. *Primal-Dual Interior-Point Methods.* SIAM, 2 edition, 1997.

Y. Zhang. On extending some primal–dual interior-point algorithms from linear programming to semidefinite programming. *SIAM Journal on Optimization*, 8(2): 365–386, 1998.