LINEARLY CONVERGENT FIRST-ORDER ALGORITHMS FOR SEMIDEFINITE PROGRAMMING

Cong D. Dang  
*Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611  
Email: congdd@ufl.edu

Guanghui Lan  
H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology,  
Atlanta, GA 30332, USA  
Email: george.lan@isye.gatech.edu

Zaiwen Wen  
Beijing International Center for Mathematical Research, Peking University, Beijing 100871, China  
Email: wenzw@pku.edu.cn

Abstract

In this paper, we consider two different formulations (one is smooth and the other one is nonsmooth) for solving linear matrix inequalities (LMIs), an important class of semidefinite programming (SDP), under a certain Slater constraint qualification assumption. We then propose two first-order methods, one based on subgradient method and the other based on Nesterov’s optimal method, and show that they converge linearly for solving these formulations. Moreover, we introduce an accelerated prox-level method which converges linearly uniformly for both smooth and non-smooth problems without requiring the input of any problem parameters. Finally, we consider a special case of LMIs, i.e., linear system of inequalities, and show that a linearly convergent algorithm can be obtained under a much weaker assumption.

Mathematics subject classification: 90C25, 90C06, 90C22, 49M37.  
Key words: Semi-definite Programming, Linear Matrix Inequalities, Error Bounds, Linear Convergence.

1. Introduction

Semidefinite Programming (SDP) is one of most interesting branches of mathematical programming in last twenty years. Semidefinite Programming can be used to model many practical problems in various fields such as convex constrained optimization, combinatorial optimization, and control theory, please refer to [1] for a general survey on SDP. Algorithms for solving SDP have been explosively studied since the landmarking works were made by Nesterov and Nemirovski [2–5], in which they showed that the interior point (IP) methods for linear programming (LP) can be extended to SDP (related topics can be found in [6–10]). Despite the fact that SDP can be solved in polynomial time by IP methods, the cost per iteration for these methods become prohibitively large as the number of variables or constraints increase. The recent research during the last few years has been focused on first-order methods due to their reduced iteration cost for solving large scale SDP (e.g., Nesterov’s optimal methods [11, 12], Nemirovski’s mirror-prox method [13], and spectral bundle methods [14]).
In [15], Lan, Lu and Monteiro proposed a class of first-order methods which exhibit an $O(1/\epsilon)$ iteration complexity for solving large-scale semidefinite programming. The basic idea is to reformulate the primal and dual SDP problems as a linear matrix inequality (LMI) and then apply Nesterov’s method [11,12] or Nemirovski’s mirror-prox method [13] to solve the resulting reformulation. Their work shows the possibility of solving general SDPs by developing efficient algorithms for solving LMIs. It is also worth noting that LMIs are important modeling tools in their own right and have been widely studied in system and control theory, system identification and signal processing [16,17]. In [18,19], the positive definite variable $X$ is replaced by $RR^\top$ in the primal augmented Lagrangian framework, then the limited-memory BFGS method is applied to minimize each augmented Lagrangian function. In [20,21], the dual augmented Lagrangian function is first reformulated by using a projection operator, then it is minimized by a semismooth Newton approach combined with the conjugate gradient method. An alternating direction method of multipliers [22] also minimizes the dual augmented Lagrangian function sequentially with respect to the Lagrange multipliers corresponding to the linear constraints, then the dual slack variables and finally the primal variables, while in each minimization keeping the other variables fixed.

Inspired by the importance of SDP and the related LMIs, we propose in this paper a new class of first-order methods that converge linearly to solve LMIs. Our development utilizes the error bounding techniques that have been intensively studied in nonlinear optimization (e.g., Luo and Tseng [23–25] on the error bounds for solving a variety of nonlinear optimization problems, Zhang [26] on the error bounds for general convex conic problem under some various conditions, Deng and Hu [27] and Jourani and Ye [28] on the error bound for semidefinite programming, and other related topics in [29, 6, 7]). More specifically, consider the optimization problem of

$$f^* = \min_{x \in X} f(x),$$

(1.1)

where $X \subseteq \mathbb{R}^n$ is a closed convex set and $f : X \rightarrow \mathbb{R}$ is a closed convex function. Let $X^*$ be the set of optimal solutions and denote

$$d(x, X^*) := \min_{y \in X^*} \|x - y\|$$

These error bounds provide, under certain slater conditions, the relationship between the distance to the set of optimal solutions and the function values, e.g.,

$$d(x, X^*) \leq \mu |f(x) - f^*|$$

for some $\mu > 0$. It should be noted, however, that the value of $\mu$ is usually unknown and difficult to estimate.

Our contribution in this paper mainly lies on the following several aspects. Firstly, we propose a nonsmooth reformulation for LMIs, given in the form of (1.1) with objective function being a nonsmooth convex function. By properly restarting the classical subgradient method, we show that the iteration complexity for solving SDP can be bounded by $O(\mu^2 \log(1/\epsilon))$. Each iteration of this method involves only a maximum eigenvalue decomposition. Secondly, in order to improve the previous iteration complexity bound, we study a smooth reformulation of LMIs, still given in the form of (1.1), but with a smooth objective function $f$. We show that by properly restarting Nesterov’s optimal method, the iteration complexity for solving the smooth reformulation of LMIs can be bounded by $O(\mu \log(1/\epsilon))$, but each iteration of this algorithm requires a full eigenvalue decomposition. Thirdly, while the aforementioned methods require
the estimation on the unknown quantity $\mu$ and some other problem parameters, we develop a variant of the accelerated prox-level (APL) method by Lan [30] and show that it can achieve the same complexity bounds as the aforementioned (sub)gradient type for solving the nonsmooth and smooth reformulations, but without requiring the estimation on any problem parameters, including the unknown parameter $\mu$. Fourthly, observing that linear system of inequalities is a special case of LMIs, we propose a linearly convergent algorithm for solving a system of linear inequalities, which requires a weaker assumption than the one for general LMIs. We refer to [31] for other linear convergent algorithms for solving a system of linear inequalities.

The paper is organized as follows. In Section 2, we introduce the problem of interest and state the Slater constraint qualification condition. We then present in Sections 3 and 4, Respectively, a non-smooth and a smooth reformulation LMI reformulation, and propose two different linearly convergent first-order algorithms for solving them. A uniform linearly convergent APL algorithm for solving both reformulations and its convergence properties are presented in Section 5. We also discuss a special cases of LMIs, the system of linear inequalities in Section 6. Finally, we have some conclusions and remarks in the last section.

## 2. The Problem of Interest

In this section, we discuss the relationship between SDP and its LMI reformulation. In particular, we show that a pair of strictly feasible primal-dual SDP problems can be represented equivalently as an LMI.

Let $S^q$ be the set of $q \times q$ symmetric matrices. Given a linear operator $A : \mathbb{R}^p \rightarrow S^q$, $c \in \mathbb{R}^p$ and $b \in S^q$, we consider the SDP problem

$$\min_x \{ \langle c, u \rangle : Au \preceq b \}$$

(2.1)

and its associated dual

$$\max_y \{ \langle b, v \rangle : A^Tv = c, v \preceq 0 \},$$

(2.2)

where $v \in S^q$ and $A^T$ denotes the adjoint of $A$. We make the following assumption.

**Assumption 2.1.** Both the primal and dual SDP problems (2.1) and (2.2) are strictly feasible.

It is well-known that in view of Assumption 2.1, the pair of primal and dual SDP problems in (2.1) and (2.2) satisfy the Slater's condition, hence they have optimal solutions and their associated gap duality is zero (see [32]). Moreover, a pair of primal-dual optimal solutions of (2.1) and (2.2) can be found by solving the following linear matrix inequalities

$$\begin{align*}
Au & \preceq b \\
A^Tv &= c \\
v & \preceq 0 \\
\langle c, u \rangle - \langle b, v \rangle & \leq 0.
\end{align*}$$

Note that a system of LMIs can be easily represented by a single LMI (see [32]). For convenience, from now on we just consider a single LMI defined as follows.

Given a symmetric matrix $B \in S^m$ and a linear operator $A : \mathbb{R}^n \rightarrow S^m$ defined as

$$Ax = A_1x_1 + \ldots + A_nx_n,$$

we consider the LMI

$$\begin{align*}
Au & \preceq b \\
A^Tv &= c \\
v & \preceq 0 \\
\langle c, u \rangle - \langle b, v \rangle & \leq 0.
\end{align*}$$

(2.3)
where $A_1, A_2, \ldots, A_n \in S^m$, we are interested in finding a feasible solution $x \in \mathbb{R}^n$ s.t.

$$Ax - B \preceq 0,$$  \hfill (2.3)

under the assumption that the feasible solution set $S$ of (2.3) is nonempty. The LMI (2.3) can be represented in the conic form

$$Ax - B \in S^m_{-}, \quad \hfill (2.4)$$

where $S^m_{-}$ denotes the set of non-positive semidefinite matrices. The following assumption is made throughout the paper.

**Assumption 2.2.** There exist $\sigma > 0$ and $d \in \mathbb{R}^n$ such that

$$\sigma I_n - Ad \in S^m_{-}.$$  \hfill (2.5)

Observe that the Assumption 2.2 implies that the Slater constraint qualification condition for the feasible set of (2.4) holds (see [26–28, 33]). Also note that the quantity

$$\mu := \frac{\|d\|}{\sigma}$$

will be used frequently in the design and analysis of first-order algorithms for solving (2.3) in Sections 3–5.

### 3. A Non-smooth Optimization Method for LMI

In this section, we introduce a non-smooth formulation for the LMI in (2.3) and propose a linearly convergent algorithm for solving this non-smooth formulation.

Consider the problem of

$$f^* \equiv \min_{x \in \mathbb{R}^n} \{ f(x) := \max \{ \lambda_1(Ax - B), 0 \} \}$$ \hfill (3.1)

where $\lambda_1(Ax - B)$ denotes the maximum eigenvalue of $Ax - B$. Clearly, the objective function of (3.1) is non-differentiable. Note that computing the objective function $f(x)$ and its subgradient $g(x) \in \partial f(x)$ requires the computation of a maximal eigenvalue and its associated eigenvectors. Moreover, the objective function $f(x)$ is Lipschitz continuous, i.e., there exists a positive number $M$ such that for any $g(x) \in \partial f(x)$

$$\|g(x)\| \leq M, \quad \forall x \in \mathbb{R}^n.$$  \hfill (3.2)

The constant $M$ can be computed as follows

$$M = \|A\| := \sqrt{\sum_{i=1}^{n} \|A_i\|^2},$$

where $\|A_i\|$ is operator norm of $A_i$.

It is not difficult to see that the two problems (3.1) and (2.3) are equivalent in the following sense. If $x^*$ is an optimal solution to (3.1), then it is also a feasible solution to (2.3) and vice versa. In addition, the optima value of (3.1) is $f^* = 0$. Let $X^*$ be the set of optimal solutions of (3.1). The following lemma describes the relationship between the distance from an arbitrary point $x$ to $X^*$ and the objective function value $f(x)$.
Lemma 3.1. For any \( x \in \mathbb{R}^n \), we have
\[
d(x, X^*) \leq \mu f(x),
\]
where \( X^* \) is the solution set of (2.3) and \( \mu \) is defined in (2.5).

Proof. Note that \( X^* \) is the set of optimal solutions for (3.1). We consider the following two cases. If \( x \in X^* \), then obviously, \( d(x, X^*) = 0 \) and
\[
f(x) = \min \{ \lambda_1 (Ax - B), 0 \} = 0,
\]
which implies (3.3) holds for any \( x \in X^* \). Moreover, if \( x \notin X^* \), then the result follows from Corollary 1 in [28]. \( \square \)

Relation (3.3) is also called the growth condition of the objective function \( f \). Using this relation, we are now ready to describe our non-smooth algorithm to solve (3.1), which consists of an outer-inner loop. More specifically, in the outer loop, this algorithm iteratively calls the classic subgradient method (see, e.g., [34]) for \( K = 4M^2 \mu^2 \) steps with \( x_{k-1} \) as the starting point. In other words, we restart the subgradient algorithm after a constant number \( K = 4M^2 \mu^2 \) of inner iterations. We use \( \{ x_k \} \) to denote the sequence obtained by the outer loop of our algorithm and \( \{ \bar{x}_i \} \) to denote the sequence computed by the subgradient method in Step 1. Hence each subgradient step can be represented by \( \bar{x}_i = \bar{x}_{i-1} - \gamma_i g(\bar{x}_{i-1}) \), \( i = 1, \ldots, K \). This non-smooth optimization scheme is formally described as follows.

**Algorithm 3.1.** The non-smooth optimization method for LMI.

\textbf{Input:} \( x_0 \in \mathbb{R}^n \).

\textbf{Output:} \( x_k \in \mathbb{R}^n \).

1) \( k^{th} \) iteration, \( k \geq 1 \).

Run the subgradient method with an initial solution \( \bar{x}_0 = x_{k-1} \) and constant stepsizes \( \gamma_i = f(\bar{x}_0)/(2M^2) \) for \( K = 4M^2 \mu^2 \) iterations to generate \( \{ \bar{x}_i, i = 1, \ldots, K \} \).

Set \( f_k^* := \min_{i=1, \ldots, K} f(\bar{x}_i) \).

Set \( x_k := \bar{x} \) such that \( f(\bar{x}) = f_k^* \).

2) Go to Step 1.

Observe that the above algorithm differs from the simple subgradient method in that we need to restart the subgradient method from time to time. In order to prove the main convergence results of our algorithm, we first discuss the convergence of the subgradient method used in Step 1. We provide the proof here mainly for the sake of completeness.

Lemma 3.2. Let \( \{ \bar{x}_i, i = 1, \ldots, K \} \) be generated by the subgradient method step 1 of Algorithm 3.1. Then we have
\[
f_k^* := \min_{i=1, \ldots, K} f(\bar{x}_i) \leq \frac{M \mu f(\bar{x}_0)}{\sqrt{K}},
\]
where \( M \) is defined in (3.2).

Proof. Let \( \gamma_i > 0 \) be the stepsize used in the \( i \)-th iteration of the subgradient method. For any \( i \geq 1 \) and \( x^* \in X^* \), we have
\[
\frac{1}{2} \| \bar{x}_{i+1} - x^* \|^2 = \frac{1}{2} \| \bar{x}_i - x^* \|^2 - \gamma_i \langle g(\bar{x}_i), \bar{x}_i - x^* \rangle + \frac{\gamma_i^2}{2} \| g(\bar{x}_i) \|^2,
\]
or equivalently,
\[ \gamma_i (g(\bar{x}_i), \bar{x}_i - x^*) = \frac{1}{2} \| \bar{x}_i - x^* \|^2 - \frac{1}{2} \| \bar{x}_{i+1} - x^* \|^2 + \frac{\gamma_i^2}{2} \| g(\bar{x}_i) \|^2. \]

By the convexity of \( f \) and the fact that \( f(x^*) = 0 \), we have
\[ f(\bar{x}_i) = f(\bar{x}_i) - f(x^*) \leq \langle g(\bar{x}_i), \bar{x}_i - x^* \rangle, \]
which implies
\[ \gamma_i f(\bar{x}_i) \leq \frac{1}{2} \| \bar{x}_i - x^* \|^2 - \frac{1}{2} \| \bar{x}_{i+1} - x^* \|^2 + \frac{\gamma_i^2}{2} \| g(\bar{x}_i) \|^2. \]

Summing up the above inequalities, we obtain
\[ \sum_{i=1}^{K} \gamma_i f(\bar{x}_i) \leq \frac{1}{2} \| \bar{x}_0 - x^* \|^2 - \frac{1}{2} \| \bar{x}_{i+1} - x^* \|^2 + \frac{1}{2} \sum_{i=1}^{K} \gamma_i^2 \| g(\bar{x}_i) \|^2 \]
\[ \leq \frac{1}{2} \| \bar{x}_0 - x^* \|^2 + \frac{1}{2} \sum_{i=1}^{K} \gamma_i^2 \| g(\bar{x}_i) \|^2. \]

Dividing both sides of the above inequality by \( \sum_{i=1}^{K} \gamma_i \), and using Lemma 3.1, we have
\[ \sum_{i=1}^{K} \gamma_i f(\bar{x}_i) \leq \frac{1}{2} \frac{\| \bar{x}_0 - x^* \|^2}{\sum_{i=1}^{K} \gamma_i} + \frac{1}{2} \frac{\sum_{i=1}^{K} \gamma_i^2 \| g(\bar{x}_i) \|^2}{\sum_{i=1}^{K} \gamma_i} \]
\[ \leq \frac{\mu^2 f^2(\bar{x}_0)}{2 \sum_{i=1}^{K} \gamma_i} + \frac{M^2 \gamma^2}{2 \sum_{i=1}^{K} \gamma_i}. \]

Using the constant stepsize \( \gamma_i = \frac{\gamma}{\sqrt{K}} \) with \( \gamma = \frac{\mu f(x_0)}{M} \), we obtain
\[ \min_{i=1, \ldots, K} \{ f(\bar{x}_i) \} \leq \frac{1}{2} \frac{\mu^2 f^2(\bar{x}_0)}{\gamma} + \frac{M^2 \gamma}{2 \gamma} = \frac{M \mu f(\bar{x}_0)}{\sqrt{K}}. \]
\[ \Box \]

We now establish the linear rate of convergence of Algorithm 3.1.

**Theorem 3.1.** The sequence \( \{ x_k \} \) generated by Algorithm 3.1 satisfies
\[ f_k \leq \frac{1}{2} f_{k-1}, k = 1, 2, \ldots. \]

As a consequence, given any \( \epsilon > 0 \), a solution \( \bar{x} \in \mathbb{R}^n \) satisfying \( f(\bar{x}) - f^* \leq \epsilon \) can be found in no more than
\[ 4 \| A \|^2 \mu^2 \log_2 \frac{f(\bar{x}_0)}{\epsilon} \]
subgradient iterations (or inner iterations), where \( \| A \| \) is defined in (3.4).
Proof. By Lemma 3.2, we have

\[ f_k^* \leq \frac{M \mu f_{k-1}^*}{\sqrt{K}}. \]

Noting that \( K \geq 4M^2 \mu^2 \), we have \( f_k^* \leq \frac{1}{2} f_{k-1}^* \). Following this result, after each outer iteration, the objective value is decreased by a factor of 1/2. This implies that to obtain \( \epsilon \)-solution of (3.1), we need to perform \( \log_2 \frac{f(x_0)}{\epsilon} \) outer iterations and \( 4M^2 \mu^2 \log_2(f(x_0)/\epsilon) \) subgradient iterations. \( \square \)

4. A Smooth Optimization Method for LMI

In this section, we discuss a smooth formulation for the LMI in (2.3) given by

\[ \phi^* = \min_{x \in \mathbb{R}^n} \left\{ \phi(x) := \min_{u \in S_n^+} \|Ax - B - u\|_F^2 \right\}. \]  

(4.1)

Note that \( \phi(x) \) is the square of the distance from \( Ax - B \) to the non-positive semidefinite cone \( S_n^- \). It is easy to see that if \( x^* \) is a feasible solution to (2.4) then it must be an optimal solution of (4.1) and vice versa. Furthermore, if \( x^* \) is a feasible solution to (2.4) then we also have \( \phi(x^*) = 0 \). The following lemma, whose proof is given in Proposition 1 of [15], shows that \( \phi \) is a smooth convex function.

**Lemma 4.1.** Given a linear operator \( A : \mathbb{R}^n \to S^n \), the objective function \( \phi \) given in (4.1) is convex, and its gradient is Lipschitz continuous with constant \( L = 2\|A\|_{2,F}^2 \), where

\[ \|A\|_{2,F} := \max_{\|u\| \leq 1} \{ \|Au\|_F : \|u\| = 1 \} \]  

(4.2)

It is not difficult to see that \( \|A\|_{2,F} = \sqrt{\sum_{i=1}^n \|A_i\|_F^2} \) and that \( \|A\|_{2,F} \geq \|A\| \), where \( \|A\| \) is defined in (3.2).

The following lemma provide an error bound associated with problem (4.1).

**Lemma 4.2.** For any \( x \in \mathbb{R}^n \), we have

\[ d^2(x, X^*) \leq \mu^2 \phi(x), \]  

(4.3)

where \( X^* \) is the set of feasible solutions of (2.4).

Proof. Note that \( X^* \) is also the set of optimal solutions of (4.1). We consider the following two cases.

**Case 1:** \( x \in X^* \). Then

\[ d(x, X^*) = 0, \]

\[ \phi(x) = \min_{u \in S_n^+} \|Ax - B - u\|_F^2 = 0. \]

That implies (4.3) holds for any \( x \in X^* \).

**Case 2:** \( x \notin X^* \). By Corollary 1 of [28], we have

\[ d(x, X^*) \leq \mu \lambda_1(Ax - B). \]
Since \( x \notin X^* \), \( \lambda_1(Ax - B) > 0 \). It is easy to see that
\[
\lambda_1^2(Ax - B) \leq \|Ax - B - u\|_F^2, \quad \forall u \in S^n,
\]
which implies
\[
d(x, X^*) \leq \mu^2 \phi(x), \quad \forall x \notin X^*.
\]
This completes the proof of Lemma 4.2.

We are now ready to describe the smooth optimization method for LMI. Similarly to the nonsmooth method, this algorithm consists of an outer-inner loop. In each outer iteration \( k \), it calls the Nesterov’s optimal method (see [15], [11]) for \( K = 4\mu\|A\| \) iterations with the previous iterate \( x_{k-1} \) as the starting point to compute \( x_k \). In other words, we restart the Nesterov’s optimal once every \( K \) iterations. We use \( \{x_k\} \) to denote the sequence obtained in the outer iterations and \( \{\bar{x}_i\} \) to denote the sequence obtained in the inner iterations performed by Nesterov’s optimal method. This method is formally stated in Algorithm 4.1.

**Algorithm 4.1.** The smooth optimization method for LMI.

**Input:** \( x_0^0 \in \mathbb{R}^n \).
**Output:** \( x_k \in \mathbb{R}^n \).

1) \( k^{th} \) iteration, \( k \geq 1 \).

- Run Nesterov’s algorithm with initial solution \( \bar{x}_0 = x_{k-1} \) for \( K = 4\mu\|A\| \) iterations.
- \( x_k := \bar{x}_K \).
2) Go to Step 1.

Theorem 4.1 below describes the main convergence properties of Algorithm 4.1.

**Theorem 4.1.** The sequence \( \{x_k\} \) generated by Algorithm 4.1 satisfies

\[
\phi(x_k) \leq \frac{1}{2} \phi(x_{k-1}), \quad \forall k \geq 1.
\]

As a consequence, given any \( \epsilon > 0 \), a solution \( \bar{x} \) satisfying \( f(\bar{x}) - f^* \leq \epsilon \) can be found in no more than

\[
4\mu\|A\|_2, F \log_2 \frac{\phi(x_0)}{\epsilon} (4.4)
\]
inner iterations, where \( \|A\|_2, F \) is defined in (4.2).

**Proof.** By the convergence properties of Nesterov’s algorithm (see, e.g., [11, 15]), and noting that \( \phi(x) \) has \( 2\|A\|^2, F \)-Lipschitz continuous gradient, we have

\[
\phi(x_k) - \phi^* \leq \frac{8\|A\|^2_2, F d^2(x_{k-1}, X^*)}{K^2},
\]
which, in view of the fact that \( \phi^* = 0 \), then implies that

\[
\phi(x_k) \leq \frac{8\|A\|^2_2, F d^2(x_{k-1}, X^*)}{K^2}.
\]

By Lemma 4.3, we have

\[
d^2(x_{k-1}, X^*) \leq \mu^2 \phi(x_{k-1}).
\]
Observing that $K \geq 4\mu\|A\|_{2,F}$, we conclude that
\[
\phi(x_k) \leq \frac{8\|A\|_{2,F}^2\phi(x_{k-1})}{K^2} \leq \frac{1}{2}\phi(x_{k-1}).
\]
Hence, to obtain an $\epsilon$-solution of (4.1), we need to run at most $\log_2(f(x_0)/\epsilon)$ and $4\mu\|A\|_{2,F}\log_2(f(x_0)/\epsilon)$ outer and inner iterations, respectively.

It is interesting to compare the results obtained in (3.4) and (4.4). First, the complexity bound in (4.4) has a better dependence on $\mu$ than the one in (3.4). It should also be noted, however, that in general $\|A\|_{2,F} \geq \|A\|$. Second, the cost per iteration for solving (3.1) will be smaller than the one for solving (4.1), as it involves a maximum eigenvalue decomposition rather than a full eigenvalue decomposition. Therefore, the non-smooth formulation might work better for problems with large $m$ and sparse matrix $A$, while the smooth formulation would be a better fit for dense problems with a relatively small dimension $m$.

5. Uniform Linearly Convergent Algorithm for LMI

One problem associated with the two linearly convergent algorithms for solving LMIs in Sections 3 and 4 is that they require quite a few parameters, e.g., the Lipschitz constants $M$ and $L$, and the error bounding constant $\mu$. However, some of these constants, e.g., $\mu$, are often hard to estimate. In this section, we present a new first-order method which converges linearly not only for smooth but also for non-smooth formulations. Moreover, this algorithm does not require the input of any problem parameters such as $L$, $M$, and $\mu$.

Consider a general convex programming problem of
\[
\psi^* := \min_{x \in \mathbb{R}^n} \psi(x),
\]
where the optimal value $\psi^*$ is known and $\psi$ satisfies
\[
\psi(y) - \psi(x) - \langle \psi'(x), y - x \rangle \leq \frac{L}{2}\|y - x\|^2 + M\|y - x\|, \quad \forall x, y \in \mathbb{R}^n,
\]
for some $L, M \geq 0$ and $\psi'(x) \in \partial\psi(x)$. Moreover, letting $X^*$ be the set of optimal solutions of (5.1), we assume that the following error bound condition
\[
[d(x,X^*)]^{\rho} \leq \eta[\psi(x) - \psi^*], \quad \forall x \in \mathbb{R}^n,
\]
holds for some $\rho, \eta > 0$. Clearly, this class of problems covers both the non-smooth formulation in (3.1) (with $\psi = f$, $L = 0, M = \|A\|$, $\rho = 1$, and $\eta = \mu$) and the smooth formulation in (4.1) (with $\psi = \phi$, $L = 2\|A\|_{2,F}^2, M = 0$, $\rho = 2$, and $\eta = \mu^2$). Moreover, we have $\psi^* = 0$ for both these cases. In [30], Lan propose two algorithms which are uniformly optimal for solving both non-smooth and smooth convex programming problems. Moreover, these algorithms do not require any smoothness information, such as the size of the Lipschitz constants. Our goal in this section is to specialize the accelerated prox-level method (APL) in [30] for solving (5.1) and show that the resulting algorithm exhibits uniformly linear rate of convergence for solving (5.1) without requiring the input of any problem parameters, such as $L$, $M$, $\rho$, and $\eta$. 
5.1. The APL algorithm for solving LMIs

The basic idea of APL method is to construct a sequence of upper and lower bounds on $\psi^*$ whose gap converges to 0. Since the optimal value $\psi^*$ is given and hence it is not necessary to construct a lower bound on $\psi^*$, we introduce below a much simplified gap reduction procedure than the original one in [30]. We will then iteratively call this simplified gap reduction procedure to solve problem (5.1).

**The simplified APL gap reduction procedure.**

**Input:** $x^u_0 \in \mathbb{R}^n$.

**Output:** $\bar{x} \in \mathbb{R}^n$.

**Initialize:** Set $\bar{\psi}_0 = \psi(x^u_0)$ and $t = 1$. Also let $x_0$ be arbitrary chosen, say $x_0 = x^u_0$.

1) Set
\[
x_t^l = (1 - \alpha_t)x_{t-1}^u + \alpha_t x_{t-1}.
\] (5.4)

2) Update the prox-center: set
\[
x_t \in \text{argmin}\{\|x - x_{t-1}\|^2 : h(x_t, x) \leq l\},
\] where $l = \psi^*$ and
\[
h(z, x) := \psi(z) + \langle \psi'(z), x - z \rangle.
\] (5.6)

3) Update the upper bound: choose $x_t^u \in \mathbb{R}^n$ such that
\[
\psi(x_t^u) \leq \min\{\bar{\psi}_{t-1}, \psi(\alpha_t x_t + (1 - \alpha_t)x_{t-1}^u)\},
\]
and set $\bar{\psi}_t = \psi(x_t^u)$. In particular, set $x_t^u = \bar{x}_t^u \equiv \alpha_t x_t + (1 - \alpha_t)x_{t-1}^u$ if $\psi(\bar{x}_t^u) \leq \bar{\psi}_{t-1}$, and set $x_t^u = x_{t-1}^u$ otherwise.

4) If $\bar{\psi}_t \leq \frac{1}{2} \psi_0$, terminate the procedure with output $\bar{x} = x_t^u$.

5) Set $t = t + 1$ and go to Step 1.

The above simplified gap reduction procedure differs from the APL gap reduction procedure described in [30] in the following several aspects. Firstly, we do not need to update the lower bound because the optimal value is known. Secondly, we use the same level $l = \psi^*$ for every step. Thirdly, each bundle contains only one cutting plane, which makes our subproblem in Step 2 very simple. Note that in order to guarantee the convergence of the above simplified gap reduction procedure, we need to properly specify the stepsizes $\{\alpha_t\}$. Our selection of $\alpha_t$ will be similar to the APL gap reduction procedure. More specifically, we denote
\[
\Gamma_t := \begin{cases} 1, & t = 1, \\ \Gamma_t(1 - \alpha_t), & t \geq 2. \end{cases}
\] (5.7)

and assume that $\alpha_t \in (0, 1], t \geq 1$, are chosen such that
\[
\alpha_1 = 1, \quad \frac{\alpha_t^2}{\Gamma_t^2} \leq C_1, \quad \Gamma_t \leq \frac{C_2}{t^2},
\]
\[
\Gamma_t \left[ \sum_{\tau=1}^{t} \left( \frac{\alpha_\tau}{\Gamma_\tau} \right)^2 \right]^{\frac{1}{2}} \leq \frac{C_3}{\sqrt{t}}, \quad \forall t \geq 1.
\] (5.8)
The following result, whose proof can be found in Lemma 6 of [30], provides two different ways to specify the stepsizes $\alpha_t, t \geq 1$. It is worth noting that these stepsizes policies do not depend on any information of $L, M, \psi(x_0)$, and any error bounding conditions.

**Lemma 5.1.** a) If $\alpha_t, t \geq 1$, are set to

$$\alpha_t = \frac{2}{t + 1},$$

then the condition (5.8) holds with $C_1 = 2, C_2 = 2$ and $C_3 = 2/\sqrt{3}$.

b) If $\alpha_t, t \geq 1$, are computed recursively by

$$\alpha_1 = \Gamma_1 = 1, \quad \alpha_t^2 = (1 - \alpha_t)\Gamma_{t-1} = \Gamma_t, \quad \forall t \geq 2,$$

then we have $\alpha_t \in (0, 1]$ for any $t \geq 2$. Moreover, condition (5.8) holds with $C_1 = 1, C_2 = 4$ and $C_3 = \frac{4}{\sqrt{3}}$.

Observe that whenever the simplified APL gap reduction procedure terminates, the objective value is reduced by $1/2$. Our main APL algorithm will iteratively call this procedure to solve (5.1).

**Algorithm 5.1.** The APL method for LMI.

**Input:** Initial point $p_0 \in \mathbb{R}^n$ and tolerance $\epsilon > 0$.

0) Set $ub_1 = \psi(x_0)$ and $s = 1$.

1) If $ub_s \leq \epsilon$, terminate.

2) Call the simplified APL gap reduction procedure with input $x_0 = p_s$.

   - Set $p_{s+1} = \bar{x}, ub_{s+1} = \psi(\bar{x})$, where $\bar{x}$ is the output of the gap reduction procedure.

3) Set $s = s + 1$ and go to Step 1.

We say that an outer iteration of the APL method occurs whenever $s$ increments 1 and an iteration performed by gap reduction procedure will be called an inner iteration of the APL method.

The main convergence properties of the above algorithm are described as follows.

**Theorem 5.1.** Suppose that $\alpha_t \in (0, 1], t \geq 1$, in the simplified APL method are chosen such that (5.8) holds. Then the total number of inner iterations performed by Algorithm 5.1 can be bounded by $T \max\{0, \log_2 \frac{\psi(p_0) - \psi^*}{\epsilon}\}$, where

$$T := \max \left\{ \left( \frac{2LC_1C_2\eta^2}{\varepsilon^2} \right)^\frac{1}{2}, \left( \frac{4MC_3\eta}{\varepsilon^2} \right)^2 \right\}. \quad (5.11)$$

The following results follow as immediate consequences of Theorem 5.1.

**Corollary 5.1.** Suppose that $\alpha_t \in (0, 1], t \geq 1$, in the simplified APL method are chosen such that (5.8) holds.

a) The total number inner iterations performed by Algorithm 5.1 applied to the non-smooth formulation in (3.1) can be bounded by

$$O \left\{ \mu^2 \| A \|^2 \log_2 \frac{f(p_0)}{\varepsilon} \right\}. \quad (5.12)$$
b) The total number inner iterations performed by Algorithm 5.1 applied to the smooth formulation in (4.1) can be bounded by

$$O \left\{ \mu \|A\|_2 \varphi \log_2 \frac{\phi(p_0)}{\varepsilon} \right\}. \quad (5.13)$$

We add a few remarks about the results obtained in Theorem 5.1 and Corollary 5.1. First, observe that both complexity bounds in (5.12) and (5.13) are obtained by the same algorithm with the same parameter settings applied to different problems. This differs from the complexity results in (3.4) and (4.4) obtained by running two different algorithms to different problem formulations.

Second, note that the growth condition (5.3) plays an important role in the development and analysis of our algorithm. Indeed, the proposed algorithm can be applied to many more convex optimization problems other than LMIs, as long as Assumption (5.3) holds.

5.2. Convergence analysis for the APL method

In this section, we provide the proof of Theorem 5.1. We first establish the convergence properties of the simplified gap reduction procedure before proving Theorem 5.1.

The following lemma shows that the simplified APL gap reduction procedure generates a sequence of prox-centers \( x_t \) which are “close” enough to each other.

**Lemma 5.2.** Suppose that \( x_{\tau}, \tau = 0, 1, \ldots, T, \) are the prox-centers generated by the simplified APL gap reduction procedure, then we have

$$\sum_{\tau=1}^{T} \| x_{\tau} - x_{\tau-1} \|^2 \leq \| x_0 - x^* \|^2,$$

where \( x^* \) is an arbitrary optimal solution of (5.1).

**Proof.** Denote the bundle sets by

$$\mathcal{L}_t := \left\{ x \in \mathbb{R}^n : h(x^t, x) \leq l \right\}, \quad t = 1, \ldots,$$

and let \( x^* \) be an arbitrary optimal solution of (5.1). Then by the convexity of the objective function \( \psi \), it is easy to see that \( x^* \) is a feasible solution to (5.5) for every step, i.e., \( x^* \in \mathcal{L}_t, t = 1, 2, \ldots, T - 1 \). Furthermore, using Lemma 1 in [35] and the definition of \( x_{\tau} \) in (5.5), we have

$$\| x_{\tau} - x^* \|^2 + \| x_{\tau-1} - x_{\tau} \|^2 \leq \| x_{\tau-1} - x^* \|^2, \quad \tau = 1, \ldots, T.$$

Summing up the above inequalities we obtain

$$\| x_T - x^* \|^2 + \sum_{\tau=1}^{T} \| x_{\tau-1} - x_{\tau} \|^2 \leq \| x_0 - x^* \|^2, \quad \forall x^* \in X^*.$$

This completes the proof of Lemma 5.2. \( \square \)

The following result describes the main recursion for the simplified APL gap reduction procedure which together with the global error bound (5.3) imply the rate of convergence of Algorithm 5.1.
Lemma 5.3. Let \((x^t_1, x^t_2, x^t_3), t \geq 1\), be the search points computed by the simplified APL gap reduction procedure. Also, let \(\Gamma_t\) be defined in (5.7) and suppose that the stepsizes \(\alpha_t, t \geq 1\), are chosen such that the relation (5.8) holds. Then for any \(x^* \in X^*\) we have

\[
\psi(x^t_i) - \psi^* \leq \frac{LC_1 C_2}{2t^2} \|x_0 - x^*\|^2 + \frac{MC_3}{\sqrt{t}} \|x_0 - x^*\|. \tag{5.14}
\]

Proof. Denote \(\tilde{x}^t_i = \alpha_t x_t + (1 - \alpha_t)x^t_{-1}\). By the definition of \(x^t_i\), we have

\[
\tilde{x}^t_i - x^t_i = \alpha_t (x_t - x_{t-1}).
\]

Using this observation, (5.2), (5.4)–(5.6), and the convexity of \(\psi\), we have

\[
\psi(x^t_i) \leq \psi(\tilde{x}^t_i) \leq (1 - \alpha_t)\psi(x^t_{-1}) + \alpha_t \psi(x_t) + \frac{L\alpha_t^2}{2} \|x_t - x_{t-1}\|^2 + M\alpha_t \|x_t - x_{t-1}\|.
\]

Subtracting \(l = \psi^*\) from both sides of the above inequality, we obtain, for any \(t \geq 1\),

\[
\psi(x^t_i) - \psi^* \leq (1 - \alpha_t)\psi(x^t_{-1}) - \psi^* + \frac{L\alpha_t^2}{2} \|x_t - x_{t-1}\|^2 + M\alpha_t \|x_t - x_{t-1}\|.
\]

Dividing both sides of the above inequality by \(\Gamma_t\), and using (5.7) and (5.8), we have

\[
\frac{\psi(x^t_i) - \psi^*}{\Gamma_t} \leq \frac{1 - \alpha_t}{\Gamma_t} \psi(x^t_{-1}) - \psi^* + \frac{L\alpha_t^2}{2} \frac{\|x_t - x_{t-1}\|^2}{\Gamma_t} + M\frac{\alpha_t}{\Gamma_t} \|x_t - x_{t-1}\|
\]

and for any \(t \geq 2\)

\[
\frac{1}{\Gamma_{t-1}} \psi(x^t_i) - \psi^* \leq \frac{1 - \alpha_t}{\Gamma_t} \psi(x^t_{-1}) - \psi^* + \frac{L\alpha_t^2}{2} \frac{\|x_t - x_{t-1}\|^2}{\Gamma_t} + M\frac{\alpha_t}{\Gamma_t} \|x_t - x_{t-1}\|
\]

Summing up the above inequalities, we have, for any \(t \geq 1\),

\[
\frac{1}{\Gamma_t} \psi(x^t_i) - \psi^* \leq \frac{LC_1}{2} \sum_{\tau = 1}^{t} \|x_\tau - x_{\tau-1}\|^2 + M \sum_{\tau = 1}^{t} \frac{\alpha_t}{\Gamma_t} \|x_\tau - x_{\tau-1}\|
\]

where the second inequality follows from the Cauchy-Schwartz inequality. It then follows from (5.8) and Lemma 5.2 that for any \(t \geq 1\) and \(x^* \in X^*\),

\[
\psi(x^t_i) - \psi^* \leq \frac{LC_1 \Gamma_t}{2} \|x_0 - x^*\|^2 + MC_3 \Gamma_t \left[ \sum_{\tau = 1}^{t} \left( \frac{\alpha_t}{\Gamma_t} \right)^2 \right]^{\frac{1}{2}} \|x_0 - x^*\|
\]

\[
\leq \frac{LC_1 C_2}{2t^2} \|x_0 - x^*\|^2 + \frac{MC_3}{\sqrt{t}} \|x_0 - x^*\|.
\]

This completes the proof of the lemma. \(\Box\)
Now we are ready to prove the Theorem 5.1.

**Proof of Theorem 5.1.** By Lemma 5.3 and (5.3), we have
\[
\psi(x_u^t) - \psi^* \leq \frac{LC_1 C_2}{2t^2} [d(x_0, X^*)]^2 + \frac{MC_3}{\sqrt{t}} d(x_0, X^*)
\]
\[
\leq \frac{LC_1 C_2}{2t^2} [\eta(\psi(x_0) - \psi^*)]^2 + \frac{MC_3}{\sqrt{t}} [\eta(\psi(x_0) - \psi^*)]\frac{3}{2},
\]
which, in view of the definition of $T$ in (5.11), implies that $\psi(x_u^t) - \psi^* \leq (\psi(x_0) - \psi^*)/2$.
Therefore, the number of iterations performed by each call to the simplified APL gap reduction procedure must be bounded by $T$. Moreover, it is easy to see that the number of outer iteration is bounded by
\[
\max \left\{ 0, \log_2 \frac{\psi(p_0) - \psi^*}{\epsilon} \right\}.
\]
Hence, the total number of inner iterations is bounded by $T \max \{0, \log_2 \frac{\psi(p_0) - \psi^*}{\epsilon} \}$.

---

**6. Application to a System of Linear Inequalities**

In this section, we consider a system of linear inequalities, which is a special case of LMIs. We show that we still preserve the linear rate of convergence for solving a linear system of inequalities system under a weaker assumption than Assumption 2.2.

Consider the system of linear inequalities given by

\[
Ax \leq b,
\]

or

\[
\begin{align*}
& a_i^T x \leq b_i \quad (i \in I_{\leq}) \\
& a_i^T x = b_i \quad (i \in I_{=})
\end{align*}
\]

where $A$ is an $m \times n$ matrix, and $I_{\leq}$ and $I_{=}$ are index sets corresponding to inequalities and equalities, respectively. We make the following assumption in this section.

**Assumption 6.1.** The solution set of (6.1) is non-empty.

Note that this assumption is weaker than A 2.2 which requires the strict feasibility of (6.1).

We introduce the function $e : \mathbb{R}^m \rightarrow \mathbb{R}^m$ such that
\[
e(y)_i = \begin{cases} 
 y_i^+ & (i \in I_{\leq}) \\
 y_i & (i \in I_{=})
\end{cases},
\]
where $y_i^+ = \max \{0, y_i\}$.

Then, an equivalent optimization problem of (6.1) is given by

\[
\min_{x \in \mathbb{R}^n} \left\{ \theta(x) := \frac{1}{2} ||e(Ax - b)||^2 \right\},
\]

(6.2)

Note that $x^*$ is a solution of (6.1) if and only if $x^*$ is optimal solution of (6.2), and $\theta(x^*) = 0$.

Lemma 6.1 shows the smoothness of the the objective function $\theta(x)$. 

Lemma 6.1. Given a matrix $A \in \mathbb{R}^{m \times n}$ and a vector $b \in \mathbb{R}^m$, the objective function given in (6.2) is convex, and its gradient is Lipschitz continuous with constant $\|A\|^2$.

Proof. Denote
\[
C := \{ y \in \mathbb{R}^m : y_i \leq 0 \text{ for } i \in I_\leq, y_i = 0 \text{ for } i \in I_\geq \}.
\]
Then $\|e(y)\|$ is the distance from a point $y$ to the closed convex set $C$. Using Proposition 15 in [15], we can show that $\|e(Ax - b)\|^2$ is convex and differentiable. Moreover, its gradient is given by
\[
\nabla f(x) = A^T(y - \Pi_C(y)),
\]
where $y = Ax - b$ and $\Pi_C(y)$ is projection of $y$ on $C$. Also, we have
\[
\begin{aligned}
&\|A^T(y_1 - \Pi_C(y_1)) - A^T(y_2 - \Pi_C(y_2))\| \\
\leq &\|A\|\|\Pi_C(y_1) - \Pi_C(y_2)\| \\
= &\|A\|\|y_1 - y_2\| \\
\leq &\|A\|^2\|x_1 - x_2\|,
\end{aligned}
\]
which implies
\[
\|\nabla f(x_1) - \nabla f(x_2)\| \leq \|A\|^2\|x_1 - x_2\|.
\]
This completes the proof of the lemma.

The growth condition of the objective function $\theta$ is described in the following lemma, which was proved by Hoffman (see [36]).

Lemma 6.2. For any right-hand side vector $b \in \mathbb{R}^m$, let $S_b$ be the set of feasible solutions of the linear system (6.2). Then there exists a constant $L_H$, independent of $b$, with the following property:
\[
x \in \mathbb{R}^m \text{ and } S_b \neq \emptyset \Rightarrow d(x, S_b) \leq L_H\|e(Ax - b)\|.
\]  
(6.3)

The objective function $\theta$ can be viewed as an error measure which determines the violation in the corresponding equalities or inequalities of a given arbitrary point. Lemma 6.2 provides an error bound on the distance from a arbitrary point to the set of feasible solutions of (6.2). The minimum constant $L_H$ satisfying the growth condition (6.3) is called the Hoffman constant (see, e.g., [26, 29, 37–39]). Such a constant can be easily estimated in some cases, e.g., in linear system of equations. In that case, the Hoffman constant is the smallest non-zero singular value of the matrix $A$.

In view of the previous two results, we can apply the APL method in Section 5 to solve (6.2) without estimating any problem parameters. The following iteration complexity result is an immediate consequence of Theorem 5.1.

Corollary 6.1. Given any $\epsilon > 0$, a point $\bar{x}$ satisfying $\theta(\bar{x}) - \theta^* \leq \epsilon$ can be found by the APL method applied to (6.2) in no more than
\[
\mathcal{O}\left\{ \|A\|L_H \log_2 \frac{\theta(x_0)}{\epsilon} \right\}
\]
inner iterations.
7. Conclusions

We present a nonsmooth and smooth method for solving an LMI, one based on the subgradient method and the other based on Nesterov’s optimal method. We show that both methods exhibit a linear rate of convergence for solving LMIs and hence SDP. The basic idea of these algorithms is to restart an optimal method for smooth and non-smooth convex problems after a constant number of iterations and to utilize a global error bound for LMI. While these (sub)gradient-type algorithms require knowledge about many problems parameters, we introduce a uniform linearly convergent algorithm for both smooth and nonsmooth formulations of an LMI by properly modifying the APL method. This algorithm does not require the input of any problem parameters such as the Lipschitz constant and the parameters for the global error bound. We also show that a special case of LMI, a system of linear inequalities can be solved with a linear rate of convergence under a weaker assumption than general LMIs.

References


