SUCCESSIVE LINEARIZATION METHODS FOR NONLINEAR SEMIDEFINITE PROGRAMS[1]

Christian Kanzow[2], Christian Nagel[2] and Masao Fukushima[3]

[2] Institute of Applied Mathematics and Statistics
University of Würzburg
Am Hubland
97074 Würzburg
Germany
e-mail: kanzow@mathematik.uni-wuerzburg.de
nagel@mathematik.uni-wuerzburg.de

[3] Department of Applied Mathematics and Physics
Graduate School of Informatics
Kyoto University
Kyoto 606-8501
Japan
e-mail: fuku@i.kyoto-u.ac.jp

August 20, 2003

Abstract. We present a successive linearization method with a trust region-type globalization for the solution of nonlinear semidefinite programs. At each iteration, the method solves a quadratic semidefinite program, which can be converted to a linear semidefinite program with a second order cone constraint. A subproblem of this kind can be solved quite efficiently by using some recent software for semidefinite and second-order cone programs. The method is shown to be globally convergent under certain assumptions. Some numerical results are included in order to illustrate its behaviour.

Key Words. Nonlinear semidefinite programs, successive linearization method, global convergence.

[1]The research of the first and third authors was supported in part by a Grant-in-Aid for Scientific Research from the Ministry of Education, Science, Sports, and Culture of Japan. The research of the second author was supported by the DFG (Deutsche Forschungsgemeinschaft).

1 Introduction

In this paper, we consider the nonlinear semidefinite program

$$\min_{X \in \mathcal{S}^{n\times n}} f(X) \quad \text{s.t.} \quad g(X) \le 0,\ X \succeq 0, \tag{1}$$

where $f : \mathcal{S}^{n\times n} \to \mathbb{R}$ and $g : \mathcal{S}^{n\times n} \to \mathbb{R}^m$ are continuously differentiable functions, $\mathcal{S}^{n\times n}$ denotes the subset of all symmetric matrices in $\mathbb{R}^{n\times n}$, and $X \succeq 0$ indicates that X is symmetric positive semidefinite. Equality constraints may also be included, but we omit them in order to lessen the notational overhead.

The program (1) is an extension of the standard linear semidefinite program which has been studied extensively during the last decade, see, e.g., [19] and the references therein. Research activities on the nonlinear program (1) are much more recent and still in its preliminary phase. Some recent references include [11, 14, 6, 3, 13, 5, 12, 2, 10] where different algorithms are described and investigated theoretically or numerically. Further note that some of these references concentrate on special cases (like bilinear matrix inequalities) of problem (1). The methods investigated in [3, 5, 2] are of the sequential quadratic programming-type, while [13, 10] discuss augmented or modified augmented Lagrangian techniques, and [11, 12, 14] discuss interior-point methods for the nonlinear and possibly nonconvex problem (1). A branch-and-cut algorithm is presented in [6]. Several applications of nonlinear semidefinite programs, especially from control theory, may be found in [15, 14, 3, 9]. Optimality conditions for nonlinear semidefinite programs have been studied in [17, 4].

The algorithm to be investigated here is a successive linearization method. Successive linearization methods for standard nonlinear programs can be found in [8, 16, 22, 7]. These methods are typically quite robust and can usually be applied to larger problems than sequential quadratic programming algorithms since they deal with simpler subproblems. We present an extension of such a successive linearization method for the solution of the nonlinear semidefinite program (1). Using an exact penalty function and a trust region-type globalization, we show that our algorithm is globally convergent under certain assumptions. A subproblem we have to solve at each iteration is a quadratic semidefinite program, which can be reformulated either as a linear semidefinite program or as a semidefinite program with an additional second-order cone constraint. Hence our subproblems can be solved quite efficiently by means of the recent software developed for these types of problems, see, e.g., [20, 18].

We next introduce some notation that will be used throughout this paper. Let g_i, $i = 1, \dots, m$, be the component functions of g, and $Df(X)$ and $Dg_i(X)$ be the Fréchet derivatives of f and g_i, respectively, at X. We call a matrix $X^* \in \mathcal{S}^{n\times n}$ a *stationary point* of problem (1) if there exist *Lagrange multipliers* $(\lambda^*, U^*) \in \mathbb{R}^m \times \mathcal{S}^{n\times n}$ satisfying the following Karush-Kuhn-Tucker (KKT) conditions:

$$\begin{aligned} & Df(X^*) + \sum_{i=1}^m \lambda_i^* Dg_i(X^*) - U^* = 0, \\ & \lambda_i^* \ge 0,\ \ g_i(X^*) \le 0,\ \ \lambda_i^* g_i(X^*) = 0 \quad i = 1, \dots, m, \\ & X^* \succeq 0,\ \ U^* \succeq 0,\ \ \langle U^*, X^* \rangle = 0, \end{aligned} \tag{2}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product in $\mathbb{R}^{n\times n}$. The KKT conditions are necessary optimality conditions under certain constraint qualifications, and are also sufficient when the problem functions f and g_i are convex.

Associated with problem (1) is the *penalized problem*

$$\min_{X\in\mathcal{S}^{n\times n}} p_\alpha(X) \quad \text{s.t.} \quad X \succeq 0, \tag{3}$$

where $p_\alpha : \mathcal{S}^{n\times n} \to \mathbb{R}$ is the exact ℓ_1-penalty function defined by

$$p_\alpha(X) := f(X) + \alpha \sum_{i=1}^{m} \max\big\{0, g_i(X)\big\}$$

with penalty parameter $\alpha > 0$. Note that we do not include the cone constraint into this penalty function since we will treat this constraint separately.

The penalized problem (3) is equivalent to the following constrained optimization problem:

$$\begin{array}{cl} \displaystyle\min_{X\in\mathcal{S}^{n\times n},\xi\in\mathbb{R}^m} & \displaystyle f(X) + \alpha\sum_{i=1}^{m}\xi_i \\ \text{s.t.} & \xi_i \geq 0,\ \ \xi_i \geq g_i(X) \quad i = 1,\dots,m, \\ & X \succeq 0, \end{array} \tag{4}$$

where ξ_i, $i = 1,\dots,m$, are auxiliary variables. The KKT conditions for this problem can be written as

$$\begin{aligned} & Df(X^*) + \sum_{i=1}^{m}\lambda_i^* Dg_i(X^*) - U^* = 0, \\ & \alpha - \mu_i^* - \lambda_i^* = 0 \quad i = 1,\dots,m, \\ & \xi_i^* \geq 0,\ \ \mu_i^* \geq 0,\ \ \xi_i^*\mu_i^* = 0 \quad i = 1,\dots,m, \\ & \lambda_i^* \geq 0,\ \ g_i(X^*) - \xi_i^* \leq 0,\ \ \lambda_i^*\big(g_i(X^*) - \xi_i^*\big) = 0 \quad i = 1,\dots,m, \\ & X^* \succeq 0,\ \ U^* \succeq 0,\ \ \langle U^*, X^*\rangle = 0, \end{aligned} \tag{5}$$

where $(\lambda^*, \mu^*, U^*) \in \mathbb{R}^m \times \mathbb{R}^m \times \mathcal{S}^{n\times n}$ are Lagrange multipliers. We call X^* a *stationary point* of the penalized problem (3) if it satisfies (5) with some $\xi^* \in \mathbb{R}^m$ and $(\lambda^*, \mu^*, U^*) \in \mathbb{R}^m \times \mathbb{R}^m \times \mathcal{S}^{n\times n}$. Clearly a stationary point X^* of (3) that satisfies (5) with $\xi^* = 0$ is a stationary point of the original problem (1).

The organization of this paper is as follows: In Section 2 we present our basic successive linearization method and state some preliminary results. Section 3 investigates the global convergence behaviour of our method and shows that any accumulation point of a sequence generated by our method is at least a stationary point of a certain exact penalty problem. In Section 4 we present a modified algorithm using an automatic update of the penalty parameter with the aim that any accumulation point is feasible for the original problem (1) and therefore a stationary point of (1). Some preliminary numerical results are the contents of Section 5, and we conclude with some final remarks in Section 6. The details of the reformulation of our subproblem as a linear semidefinite program or a linear semidefinite program with a second-order cone constraint are given in Appendix A.

The notation used in this paper is quite standard in the community: We use the inner product $\langle A, B\rangle := \text{trace}(AB^T)$ in $\mathbb{R}^{n\times n}$. The corresponding norm $\|A\| := \big(\langle A, A\rangle\big)^{1/2}$ is equal to the Frobenius norm for matrices. Furthermore, we define the median of three numbers a, t, b with $a \le b$ as

$$\text{mid}\big(a, t, b\big) := \begin{cases} a, & \text{if } t < a, \\ t, & \text{if } t \in [a, b], \\ b, & \text{if } t > b. \end{cases}$$

2 Successive Linearization Method

In this section, we present a successive linearization method for solving the penalized problem (3). First let us define the function $\Phi_\alpha : \mathcal{S}^{n\times n} \times \mathcal{S}^{n\times n} \to \mathbb{R}$ by

$$\Phi_\alpha(X, \Delta X) := f(X) + \langle Df(X), \Delta X\rangle + \alpha \sum_{i=1}^{m} \max\big\{0, g_i(X) + \langle Dg_i(X), \Delta X\rangle\big\}.$$

This function serves as a first-order approximation of $p_\alpha(X + \Delta X)$.

We begin with a formal statement of the algorithm which is in the spirit of [8, 16, 22, 7].

Algorithm 2.1 *(Successive Linearization Method)*

(S.0) Choose $\alpha > 0$, $0 < \rho_1 < \rho_2 < 1$, $0 < \sigma_1 < 1 < \sigma_2$, $c_{\max} \ge c_{\min} > 0$, $c_0 \in [c_{\min}, c_{\max}]$, $X^0 \succeq 0$, *and set* $k := 0$.

(S.1) Find the (unique) solution $\Delta X^k \in \mathcal{S}^{n\times n}$ *of the subproblem*

$$\min_{\Delta X \in \mathcal{S}^{n\times n}} \frac{1}{2} c_k \langle \Delta X, \Delta X\rangle + \Phi_\alpha(X^k, \Delta X) \quad \text{s.t.} \quad X^k + \Delta X \succeq 0. \tag{6}$$

If $\Delta X^k = 0$*, then STOP.*

(S.2) Compute the ratio

$$r_k := \frac{p_\alpha(X^k) - p_\alpha(X^k + \Delta X^k)}{p_\alpha(X^k) - \Phi_\alpha(X^k, \Delta X^k)}.$$

If $r_k \ge \rho_1$*, then the* k*th iteration is called successful, and we set* $X^{k+1} := X^k + \Delta X^k$*; otherwise, the* k*th iteration is called unsuccessful, and we set* $X^{k+1} := X^k$*.*

(S.3) Update c_k *as follows:*
If $r_k < \rho_1$*, set* $c_{k+1} := \sigma_2 c_k$*.*
If $r_k \in [\rho_1, \rho_2)$*, set* $c_{k+1} := \text{mid}\big(c_{\min}, c_k, c_{\max}\big)$*.*
If $r_k \ge \rho_2$*, set* $c_{k+1} := \text{mid}\big(c_{\min}, \sigma_1 c_k, c_{\max}\big)$*.*

(S.4) Set $k \leftarrow k+1$, and go to (S.1).

We give some explanations regarding the philosophy of Algorithm 2.1. Basically, this algorithm may be viewed as a successive linearization method for problem (1) that employs a trust-region-type globalization technique. In fact, the subproblem (6) may be viewed as a linearization of the penalized problem (3) at the current iteration. However, rather than using a trust-region strategy explicitly by including an upper bound on the size of the correction ΔX, we use the trust-region idea implicitly by adding a quadratic term to the objective function multiplied by a parameter c_k. The parameter c_k plays more or less the role of the (inverse) trust-region radius.

This approach has several advantages compared to a direct use of the trust-region idea. In fact, the quadratic term in the objective function guarantees that the subproblem (6) is strongly convex and therefore has a unique solution for each iteration $k \in \mathbb{N}$ (note that the feasible set is obviously nonempty). Furthermore, this quadratic term may be viewed as a (very rough) second-order information although this is not the main motivation. In fact, calculating or approximating the second-order information seems to be very delicate and costly for the nonlinear semidefinite program (1). Hence we mainly work with the first-order information in our approach. Furthermore, since we already have the cone constraint in the subproblem (6), the explicit use of a trust-region bound may result in the conflicting situation where the intersection of the two constraints would be empty. Such a situation cannot occur when using a subproblem like (6).

The remaining part of Algorithm 2.1 is standard. The ratio r_k is the quotient of the actual and the predicted reductions for the function value of the penalty function p_α, where $\Phi_\alpha(X^k, \cdot)$ is used as a model for the function $p_\alpha(X^k + \cdot)$. If this ratio is sufficiently close to one, we accept $X^k + \Delta X^k$ as the new iterate X^{k+1}. Otherwise, we stay at X^k and increase the parameter c_k. The precise updating rule for c_k in step (S.3) is similar to those known in trust-region methods. Note, however, that we use lower and upper bounds $c_{\min}$ and $c_{\max}$, respectively, whenever the iteration is successful.

In the rest of this section, we will show that Algorithm 2.1 is well-defined. To this end, we only have to show that the denominator in the ratio r_k is positive as long as $\Delta X^k \neq 0$. Furthermore, we will justify the termination criterion in step (S.1).

We begin with the following simple result, which will be helpful in our subsequent analysis.

Lemma 2.2 *Let X^k be a given iterate and ΔX^k be the solution of the corresponding subproblem* (6)*. Then*

$$p_\alpha(X^k) - \Phi_\alpha(X^k, \Delta X^k) \geq \frac{1}{2} c_k \langle \Delta X^k, \Delta X^k \rangle.$$

Proof. Since $X^k \succeq 0$, the symmetric matrix $\Delta X := 0$ is feasible for the subproblem (6). But ΔX^k is a solution of this subproblem, so we obtain

$$\frac{1}{2} c_k \langle \Delta X^k, \Delta X^k \rangle + \Phi_\alpha(X^k, \Delta X^k) \leq \Phi_\alpha(X^k, 0) = p_\alpha(X^k).$$

This proves our statement. □

Lemma 2.2 ensures that the denominator in the ratio r_k is always nonnegative. Note that this implies that the sequence $\{p_\alpha(X^k)\}$ is monotonically nondecreasing. We next show that this denominator is equal to zero if and only if the termination criterion in step (S.1) is satisfied. Hence step (S.2) is visited only if the denominator is positive, so that Algorithm 2.1 is well-defined.

Lemma 2.3 *Let X^k be a given iterate and ΔX^k be the solution of the corresponding subproblem* (6)*. Then $p_\alpha(X^k) - \Phi_\alpha(X^k, \Delta X^k) = 0$ if and only if $\Delta X^k = 0$.*

Proof. First assume that $\Delta X^k = 0$. Then $p_\alpha(X^k) - \Phi_\alpha(X^k, \Delta X^k) = 0$ since the definition of Φ_α implies $\Phi_\alpha(X^k, 0) = p_\alpha(X^k)$. Conversely, let $p_\alpha(X^k) - \Phi_\alpha(X^k, \Delta X^k) = 0$. Lemma 2.2 then implies $0 = \frac{1}{2}c_k\langle \Delta X^k, \Delta X^k\rangle = \frac{1}{2}c_k\|\Delta X^k\|^2$ and hence $\Delta X^k = 0$. □

Next we have to justify our termination criterion in step (S.1). To this end, we will show that this criterion is satisfied if and only if the current iterate X^k is a stationary point of the exact penalty reformulation (3) of problem (1).

Before we arrive at this result, we first take a closer look at subproblem (6). Let X^k be a given iterate and let ΔX^k be the unique solution of (6). Then it is easy to see that the pair $(\Delta X^k, \xi^k)$ with components

$$\xi_i^k := \max\big\{0, g_i(X^k) + \langle Dg_i(X^k), \Delta X^k\rangle\big\} \quad i = 1, \dots, m \tag{7}$$

is the unique solution of the following optimization problem, which is equivalent to (6):

$$\begin{aligned}
\min_{\Delta X \in \mathcal{S}^{n\times n}, \xi \in \mathbb{R}^m} \quad & \frac{1}{2}c_k\langle \Delta X, \Delta X\rangle + f(X^k) + \langle Df(X^k), \Delta X\rangle + \alpha\sum_{i=1}^m \xi_i \\
\text{s.t.} \quad & \xi_i \geq 0 \quad i = 1, \dots, m, \\
& \xi_i \geq g_i(X^k) + \langle Dg_i(X^k), \Delta X\rangle \quad i = 1, \dots, m, \\
& X^k + \Delta X \succeq 0.
\end{aligned} \tag{8}$$

Since problem (8) is a convex program with a strictly feasible set, this problem is equivalent to its KKT conditions. In other words, $(\Delta X^k, \xi^k)$ is a solution of (8) if and only if there exist Lagrange multipliers $(\lambda^k, \mu^k, U^k) \in \mathbb{R}^m \times \mathbb{R}^m \times \mathcal{S}^{n\times n}$ such that the following KKT conditions hold:

$$\begin{aligned}
& c_k\Delta X^k + Df(X^k) + \sum_{i=1}^m \lambda_i^k Dg_i(X^k) - U^k = 0, \\
& \alpha - \mu_i^k - \lambda_i^k = 0 \quad i = 1, \dots, m, \\
& \xi_i^k \geq 0, \;\; \mu_i^k \geq 0, \;\; \xi_i^k\mu_i^k = 0 \quad i = 1, \dots, m, \\
& \lambda_i^k \geq 0, \;\; g_i(X^k) + \langle Dg_i(X^k), \Delta X^k\rangle - \xi_i^k \leq 0 \quad i = 1, \dots, m, \\
& \lambda_i^k\big(g_i(X^k) + \langle Dg_i(X^k), \Delta X^k\rangle - \xi_i^k\big) = 0 \quad i = 1, \dots, m, \\
& X^k + \Delta X^k \succeq 0, \;\; U^k \succeq 0, \;\; \langle U^k, X^k + \Delta X^k\rangle = 0.
\end{aligned} \tag{9}$$

Now, if $\Delta X^k = 0$ is the unique solution of the subproblem (6), then the system (9) yields

$$\begin{aligned}
&Df(X^k) + \sum_{i=1}^m \lambda_i^k Dg_i(X^k) - U^k = 0,\\
&\alpha - \mu_i^k - \lambda_i^k = 0 \quad i = 1, \dots, m,\\
&\xi_i^k \ge 0, \;\; \mu_i^k \ge 0, \;\; \xi_i^k \mu_i^k = 0 \quad i = 1, \dots, m,\\
&\lambda_i^k \ge 0, \;\; g_i(X^k) - \xi_i^k \le 0, \;\; \lambda_i^k\big(g_i(X^k) - \xi_i^k\big) = 0 \quad i = 1, \dots, m,\\
&X^k \succeq 0, \;\; U^k \succeq 0, \;\; \langle U^k, X^k \rangle = 0.
\end{aligned}$$

However, these conditions are nothing but the KKT conditions (5) for the penalized problem (3).

Summarizing these observations, we obtain the following result.

Theorem 2.4 *Let $\alpha > 0$. If $\Delta X^k = 0$ is the (unique) solution of the subproblem* (6) *for some $c_k > 0$, then X^k is a stationary point of the penalized problem* (3)*. Conversely, if X^k is a stationary point of* (3)*, then $\Delta X^k = 0$ is the unique solution of* (6) *for every $c_k > 0$.*

Proof. The statements follow immediately from the preceding arguments. □

3 Convergence Analysis

Throughout this section, we assume that Algorithm 2.1 generates an infinite sequence $\{X^k\}$. Our aim is to establish a global convergence result for Algorithm 2.1. More precisely, we will show that any accumulation point of $\{X^k\}$ is a stationary point of the penalized problem (3).

First, note that the KKT conditions (9) of the subproblem (6) immediately yield

$$\lambda_i^k \in [0, \alpha] \quad \text{and} \quad \mu_i^k \in [0, \alpha] \quad i = 1, \dots, m \tag{10}$$

for all $k \in \mathbb{N}$. Consequently, the sequences $\{\lambda_i^k\}$ and $\{\mu_i^k\}$ are bounded for all $i = 1, \dots, m$. We exploit this fact to show the following result, which will be used in the subsequent convergence analysis.

Lemma 3.1 *Let $\{X^k\}$ be a sequence generated by Algorithm 2.1, and let $\{X^k\}_{k \in K}$ be a subsequence converging to some matrix X^* in such a way that $\{c_k \|\Delta X^k\|\}_{k \in K} \to 0$. Then X^* is a stationary point of the penalized problem* (3)*.*

Proof. First note that X^* is symmetric positive semidefinite and hence feasible for problem (3). Furthermore, since $c_k \ge c_{\min}$ for all $k \in \mathbb{N}$, the assumption $\{c_k \|\Delta X^k\|\}_{k \in K} \to 0$ implies $\{\|\Delta X^k\|\}_{k \in K} \to 0$. By continuity, we also have $Df(X^k) \to Df(X^*)$, $g_i(X^k) \to g_i(X^*)$ and $Dg_i(X^k) \to Dg_i(X^*)$ as $k \to \infty, k \in K$. This implies that

$$\xi_i^k = \max\big\{0, g_i(X^k) + \langle Dg_i(X^k), \Delta X^k \rangle\big\} \to \max\big\{0, g_i(X^*)\big\} =: \xi_i^*$$

on the subsequence defined by the index set K, cf. (7). In view of (10), we may further assume without loss of generality that $\{\lambda_i^k\}_{k\in K} \to \lambda_i^*$ and $\{\mu_i^k\}_{k\in K} \to \mu_i^*$ for some $\lambda_i^*, \mu_i^* \in [0, \alpha]$ such that $\lambda_i^* + \mu_i^* = \alpha$, see (9). Using (9) once again, we then have

$$
\begin{aligned}
U^k &= c_k \Delta X^k + Df(X^k) + \sum_{i=1}^m \lambda_i^k Dg_i(X^k) \qquad (11)\\
&\to Df(X^*) + \sum_{i=1}^m \lambda_i^* Dg_i(X^*) =: U^* \qquad (12)
\end{aligned}
$$

as $k \to \infty$, $k \in K$. Therefore, taking the limit $k \to \infty$ on the subsequence K in the KKT conditions (9), we obtain (5). Hence we conclude that X^* is a stationary point of the penalized problem (3). □

Another main step toward our global convergence result is contained in the following technical lemma.

Lemma 3.2 *Let $\{X^k\}$ be a sequence generated by Algorithm 2.1 and $\{X^k\}_{k\in K}$ be a subsequence converging to some matrix X^*. If X^* is not a stationary point of the penalized problem* (3)*, then we have* $\limsup_{k\to\infty, k\in K} c_k < \infty$.

Proof. Let $\bar{K} := \{k-1 \,|\, k \in K\}$. Then we have $\{X^{k+1}\}_{k\in\bar{K}} \to X^*$. We will show that $\limsup_{k\to\infty, k\in\bar{K}} c_{k+1} < \infty$. Assume the contrary. Then, by subsequencing if necessary, we may suppose without loss of generality that

$$\lim_{k\to\infty, k\in\bar{K}} c_{k+1} = \infty. \qquad (13)$$

The updating rule in step (S.3) then implies that none of the iterations $k \in \bar{K}$ with k sufficiently large is successful since otherwise we would have $c_{k+1} \le c_{\max}$ for all these $k \in \bar{K}$. Hence we have

$$r_k < \rho_1 \qquad (14)$$

and $X^k = X^{k+1}$ for all $k \in \bar{K}$ large enough. Since $\{X^{k+1}\}_{k\in\bar{K}} \to X^*$, this implies $\{X^k\}_{k\in\bar{K}} \to X^*$, too. Further noticing that $c_{k+1} = \sigma_2 c_k$ for all unsuccessful iterations, we also have

$$\lim_{k\to\infty, k\in\bar{K}} c_k = \infty \qquad (15)$$

because of (13). We now want to show that

$$r_k \to 1 \quad \text{as } k \to \infty,\ k \in \bar{K},$$

which would then lead to the desired contradiction to (14). To this end, we first note that

$$\liminf_{k\to\infty, k\in\bar{K}} c_k \|\Delta X^k\| > 0. \qquad (16)$$

In fact, if $c_k\|\Delta X^k\| \to 0$ on a subsequence, we would deduce from Lemma 3.1 that X^* is a stationary point of the penalized problem (3) in contradiction to our assumption. Hence there is a constant $\gamma > 0$ such that

$$c_k\|\Delta X^k\| \geq \gamma \quad k \in \bar{K}.$$

By Lemma 2.2, this implies

$$p_\alpha(X^k) - \Phi_\alpha(X^k, \Delta X^k) \geq \frac{1}{2}c_k\|\Delta X^k\|^2 \geq \frac{1}{2}\gamma\|\Delta X^k\|$$

for all $k \in \bar{K}$ sufficiently large.

We further note that $\{\|\Delta X^k\|\}_{k\in\bar{K}} \to 0$. Otherwise, it would follow from (15) that $c_k\|\Delta X^k\|^2 \to \infty$ on a suitable subsequence. This, in turn, would imply that the optimal value of the subproblem (6) tends to infinity. However, this cannot be true since the feasible matrix $\Delta X := 0$ would give a smaller objective value. Hence we have $\{\|\Delta X^k\|\}_{k\in\bar{K}} \to 0$.

Taking this into account, and using $\{X^k\}_{k\in\bar{K}} \to X^*$ and the fact that f, g_i are continuously differentiable, we obtain through standard calculus arguments

$$\big|\Phi_\alpha(X^k, \Delta X^k) - p_\alpha(X^k + \Delta X^k)\big| = o(\|\Delta X^k\|) \quad \text{as } k \to \infty,\ k \in \bar{K}.$$

Summarizing these observations, we get

$$\begin{aligned}
\big|r_k - 1\big| &= \left|\frac{p_\alpha(X^k) - p_\alpha(X^k + \Delta X^k)}{p_\alpha(X^k) - \Phi_\alpha(X^k, \Delta X^k)} - 1\right| \\
&= \left|\frac{\Phi_\alpha(X^k, \Delta X^k) - p_\alpha(X^k + \Delta X^k)}{p_\alpha(X^k) - \Phi_\alpha(X^k, \Delta X^k)}\right| \\
&\leq \frac{o(\|\Delta X^k\|)}{\frac{1}{2}\gamma\|\Delta X^k\|} \to 0
\end{aligned}$$

as $k \to \infty$, $k \in \bar{K}$. This contradiction to (14) completes the proof. $\square$

As a direct consequence of this lemma, we obtain the following result.

Lemma 3.3 *Let $\{X^k\}$ be a sequence generated by Algorithm 2.1. Then there are infinitely many successful iterations.*

Proof. If not, there would exist an index $k_0 \in \mathbb{N}$ with $r_k < \rho_1$ and $X^k = X^{k_0}$ for all $k \geq k_0$. This implies $c_k \to \infty$ due to the updating rule in (S.3). However, since X^{k_0} is not a stationary point of problem (3) (otherwise we would have stopped in (S.1), cf. Theorem 2.4) and $\{X^k\} \to X^{k_0}$, we get a contradiction to Lemma 3.2. $\square$

We are now in the position to prove the main convergence result for Algorithm 2.1.

Theorem 3.4 *Let $\{X^k\}$ be a sequence generated by Algorithm 2.1. Then any accumulation point of this sequence is a stationary point of the penalized problem* (3).

Proof. Let X^* be an accumulation point and $\{X^k\}_{k\in K}$ be a subsequence converging to X^*. Since $X^k = X^{k+1}$ for all unsuccessful iterations k and since there are infinitely many successful iterations by Lemma 3.3, we may assume without loss of generality that all iterations $k \in K$ are successful.

Assume that X^* is not a stationary point of problem (3). Lemma 3.2 then implies

$$\limsup_{k\to\infty, k\in K} c_k < \infty.$$

Hence there is a constant $\gamma > 0$ such that

$$c_k \le \gamma \quad k \in K. \tag{17}$$

Since each iteration $k \in K$ is successful, we also have $r_k \ge \rho_1$. Consequently, we obtain from Lemma 2.2

$$\begin{aligned} p_\alpha(X^k) - p_\alpha(X^{k+1}) &\ge \rho_1\big(p_\alpha(X^k) - \Phi_\alpha(X^k, \Delta X^k)\big) \\ &\ge \frac{1}{2}\rho_1 c_k\langle \Delta X^k, \Delta X^k\rangle \\ &\ge \frac{1}{2}\rho_1 c_{\min}\|\Delta X^k\|^2 \end{aligned} \tag{18}$$

for all $k \in K$. Since $\{p_\alpha(X^k)\}$ is monotonically nonincreasing and bounded from below by, e.g., $p_\alpha(X^*)$, we have $p_\alpha(X^k) - p_\alpha(X^{k+1}) \to 0$ as $k \to \infty$. Therefore we obtain $\{\Delta X^k\}_{k\in K} \to 0$ from (18). By (17), this also implies $\{c_k\|\Delta X^k\|\}_{k\in K} \to 0$. But then Lemma 3.1 shows that X^* is a stationary point of (3) in contradiction to our assumption. This completes the proof. $\square$

4 Feasibility Issues

Theorem 3.4 guarantees that every accumulation point of a sequence $\{X^k\}$ generated by Algorithm 2.1 is a stationary point of the penalized problem (3). On the one hand, this result is quite nice because it holds without any assumptions. On the other hand, however, we are more interested in getting stationary points of the original program (1). This relies on the asymptotic feasibility of the generated sequence $\{X^k\}$, which may be achieved under certain assumptions by using an automatic updating rule for the penalty parameter α.

In this section, we investigate the convergence properties of the following modification of Algorithm 2.1.

Algorithm 4.1 *(Successive Linearization Method with Penalty Update)*

(S.0) Choose $\alpha_0 > 0$, $\delta > 0$, $0 < \rho_1 < \rho_2 < 1$, $0 < \sigma_1 < 1 < \sigma_2$, $c_{\max} \ge c_{\min} > 0$, $c_0 \in [c_{\min}, c_{\max}]$, $X^0 \succeq 0$, *and set* $k := 0$.

(S.1) Find the (unique) solution $\Delta X^k \in \mathcal{S}^{n\times n}$ *of the subproblem*

$$\min_{\Delta X\in\mathcal{S}^{n\times n}} \frac{1}{2}c_k\langle \Delta X, \Delta X\rangle + \Phi_{\alpha_k}(X^k, \Delta X) \quad \text{s.t.} \quad X^k + \Delta X \succeq 0. \tag{19}$$

(S.2) *Let* $\xi^k \in \mathbb{R}^m$ *be the vector with components*

$$\xi_i^k := \max\left\{0, g_i(X^k) + \langle Dg_i(X^k), \Delta X^k\rangle\right\} \quad i = 1, \dots, m.$$

If $\xi^k = 0$*, then the* k*th iteration is called feasible, and we go to (S.3). Otherwise, the* k*th iteration is called infeasible, and we set* $X^{k+1} := X^k, \quad \alpha_{k+1} := \alpha_k + \delta$*,* $c_{k+1} := \mathrm{mid}\big(c_{\min}, c_k, c_{\max}\big)$*,* $k \leftarrow k+1$*, and go to (S.1).*

(S.3) *If* $\Delta X^k = 0$*, then STOP. Otherwise, compute the ratio*

$$r_k := \frac{p_{\alpha_k}(X^k) - p_{\alpha_k}(X^k + \Delta X^k)}{p_{\alpha_k}(X^k) - \Phi_{\alpha_k}(X^k, \Delta X^k)}.$$

If $r_k \geq \rho_1$*, then the* k*th iteration is called successful, and we set* $X^{k+1} := X^k + \Delta X^k$*. Otherwise, the* k*th iteration is called unsuccessful, and we set* $X^{k+1} := X^k$*.*

(S.4) *Update* c_k *as follows:*
If $r_k < \rho_1$*, set* $c_{k+1} := \sigma_2 c_k$*.*
If $r_k \in [\rho_1, \rho_2)$*, set* $c_{k+1} := \mathrm{mid}\big(c_{\min}, c_k, c_{\max}\big)$*.*
If $r_k \geq \rho_2$*, set* $c_{k+1} := \mathrm{mid}\big(c_{\min}, \sigma_1 c_k, c_{\max}\big)$*.*

(S.5) *Set* $\alpha_{k+1} := \alpha_k$*,* $k \leftarrow k+1$*, and go to (S.1).*

The results shown in Section 2 remain valid for Algorithm 4.1 if we just replace the fixed penalty parameter α by α_k everywhere.

To prove suitable global convergence results for Algorithm 4.1, we make the following assumptions.

(A.1): The sequence $\{X^k\}$ generated by Algorithm 4.1 is bounded.

(A.2): The sequence $\{c_k \Delta X^k\}$ generated by Algorithm 4.1 is bounded.

(A.3): The sequence $\{\Delta X^k\}$ generated by Algorithm 4.1 converges to 0.

(A.4): For any given $X^* \succeq 0$, if $(\lambda^*, U^*) \in \mathbb{R}^m \times \mathcal{S}^{n\times n}$ satisfies

$$\sum_{i: g_i(X^*) \geq 0} \lambda_i^* Dg_i(X^*) - U^* = 0, \;\; \lambda^* \geq 0, \;\; U^* \succeq 0, \;\; \langle X^*, U^*\rangle = 0, \tag{20}$$

then we must have $\lambda^* = 0$ and $U^* = 0$.

Assumption (A.1) is standard in the constrained optimization literature, whereas Assumptions (A.2) and (A.3) are more restrictive and satisfied, for example, if the sequence $\{c_k \Delta X^k\}$ tends to 0. Finally, Assumption (A.4) corresponds to the extended Mangasarian-Fromovitz constraint qualification for ordinary nonlinear programs.

Our first result shows that these assumptions imply that there are only finitely many infeasible iterations.

Proposition 4.2 *Suppose that Assumptions (A.1), (A.2), and (A.3) hold and suppose that Assumption (A.4) holds at any accumulation point of a sequence* $\{X^k\}$ *generated by Algorithm 4.1. Then there are only finitely many infeasible iterations.*

Proof. For the proof, it will be convenient to write the Lagrangian of problem (8) as

$$L_k(\Delta X, \xi, \lambda, \mu, U) := \frac{1}{2} c_k \langle \Delta X, \Delta X \rangle + f(X^k) + \langle Df(X^k), \Delta X \rangle + \alpha_k \sum_{i=1}^m \xi_i - \alpha_k \sum_{i=1}^m \mu_i \xi_i$$
$$+ \alpha_k \sum_{i=1}^m \lambda_i \big(g_i(X^k) + \langle Dg_i(X^k), \Delta X \rangle - \xi_i \big) - \langle X^k + \Delta X, U \rangle,$$

where the fifth and the sixth terms are multiplied by α_k to normalize the Lagrange multipliers μ_i and λ_i, respectively. Then $(\Delta X^k, \xi^k)$ is a solution of (8) if and only if there exist multipliers $(\lambda^k, \mu^k, U^k) \in \mathbb{R}^m \times \mathbb{R}^m \times \mathcal{S}^{n \times n}$ such that the following KKT conditions hold:

$$\begin{aligned}
& c_k \Delta X^k + Df(X^k) + \alpha_k \sum_{i=1}^m \lambda_i^k Dg_i(X^k) - U^k = 0, \\
& 1 - \mu_i^k - \lambda_i^k = 0 \quad i = 1, \dots, m, \\
& \xi_i^k \geq 0, \;\; \mu_i^k \geq 0, \;\; \xi_i^k \mu_i^k = 0 \quad i = 1, \dots, m, \\
& \lambda_i^k \geq 0, \;\; g_i(X^k) + \langle Dg_i(X^k), \Delta X^k \rangle - \xi_i^k \leq 0 \quad i = 1, \dots, m, \\
& \lambda_i^k \big(g_i(X^k) + \langle Dg_i(X^k), \Delta X^k \rangle - \xi_i^k \big) = 0 \quad i = 1, \dots, m, \\
& X^k + \Delta X^k \succeq 0, \;\; U^k \succeq 0, \;\; \langle U^k, X^k + \Delta X^k \rangle = 0.
\end{aligned} \tag{21}$$

To prove the proposition by contradiction, let us assume that there are infinitely many iterations $k \in K$ such that $\xi^k \neq 0$ for all $k \in K$. Then, for each $k \in K$, there exists an index i_k such that $\xi_{i_k}^k > 0$. Subsequencing if necessary, we may assume without loss of generality that $i_k \equiv j$ for all $k \in K$ and some index $j \in \{1, \dots, m\}$ independent of k. Since (21) gives

$$0 = \big(1 - \mu_j^k - \lambda_j^k\big) \xi_j^k = \big(1 - \lambda_j^k\big) \xi_j^k,$$

we have $\lambda_j^k = 1$ for all $k \in K$. Since (21) implies $\lambda_i^k \in [0, 1]$ for all $k \in \mathbb{N}$ and all $i \in \{1, \dots, m\}$, we may also assume that $\{\lambda^k\}_{k \in K} \to \lambda^*$ for some vector $\lambda^* \geq 0$. Note that we must have $\lambda_j^* = 1$ for the particular index j.

Dividing the first equality in (21) by α_k yields

$$\frac{1}{\alpha_k} \big(c_k \Delta X^k + Df(X^k) \big) + \sum_{i=1}^m \lambda_i^k Dg_i(X^k) - \frac{1}{\alpha_k} U^k = 0. \tag{22}$$

Let X^* be an accumulation point of the subsequence $\{X^k\}_{k \in K}$ and assume without loss of generality that $\{X^k\}_{k \in K} \to X^*$. Then, Assumptions (A.1) and (A.2) together with the fact that $\{\alpha_k\} \to \infty$ (since we have $\xi^k \neq 0$ for all $k \in K$, cf. the updating rule in step (S.2)) imply

$$\frac{1}{\alpha_k} \big(c_k \Delta X^k + Df(X^k) \big) + \sum_{i=1}^m \lambda_i^k Dg_i(X^k) \;\to\; \sum_{i=1}^m \lambda_i^* Dg_i(X^*)$$

as $k \to \infty$, $k \in K$. This together with (22) gives

$$\frac{1}{\alpha_k} U^k \;\to\; U^* \quad \text{as } k \to \infty,\, k \in K$$

for some $U^* \succeq 0$, and hence we obtain

$$\sum_{i=1}^{m} \lambda_i^* D g_i(X^*) - U^* = 0.$$

Moreover, by Assumption (A.3), the last equality in (21) yields

$$0 = \frac{1}{\alpha_k} \langle U^k, X^k + \Delta X^k \rangle \;\to\; \langle U^*, X^* \rangle.$$

Furthermore, we have $\lambda_i^* = 0$ for all $i \in \{1, \ldots, m\}$ such that $g_i(X^*) < 0$. In fact, if $g_{i_0}(X^*) < 0$ for some index i_0, then it is not difficult to deduce from (21) together with Assumption (A.3), $\lambda_i^k \in [0,1]$ and $\xi_i^k \geq 0$ that $\lambda_{i_0}^* = 0$. Hence it follows that the pair (λ^*, U^*) satisfies (20). Assumption (A.4) then implies $\lambda^* = 0$, a contradiction to $\lambda_j^* = 1$. □

As a consequence of Proposition 4.2, we get the following corollary.

Corollary 4.3 *Under the assumptions of Proposition 4.2, the following statements hold:*

(a) The sequence of penalty parameters $\{\alpha_k\}$ stays constant eventually.

(b) Every accumulation point X^* of $\{X^k\}$ is feasible for the original program (1).

Proof. Statement (a) follows immediately from Proposition 4.2 together with the updating rule in step (S.2). Hence it remains to prove part (b). To this end, let X^* be an accumulation point of the sequence $\{X^k\}$, and $\{X^k\}_{k \in K}$ be a subsequence converging to X^*. From Proposition 4.2, we have

$$0 \;=\; \xi_i^k \;\geq\; g_i(X^k) + \langle D g_i(X^k), \Delta X^k \rangle \quad i = 1, \ldots, m$$

for all $k \in \mathbb{N}$ sufficiently large. Then, by Assumption (A.3), taking the limit in the above inequalities yields

$$g_i(X^*) \leq 0 \quad i = 1, \ldots, m.$$

Since $X^k \succeq 0$ for all $k \in \mathbb{N}$, we also obtain $X^* \succeq 0$. Hence X^* is feasible for the original program (1). □

Corollary 4.3 means that, under the given assumptions, Algorithm 4.1 eventually coincides with Algorithm 2.1. This observation allows us to state the following global convergence result.

Table 1: Numerical results for linear SDPs

problem	n	m	k	α_k	c_k	$\|\xi_k\|_\infty$	$\|\mathrm{svec}(\Delta X^k)\|_\infty$	$\lambda_{\min}(X^k)$	SDPT3 exit
random	10	10	4	50	1.00e-03	1.58e-12	3.24e-06	-3.14e-10	0
norm min	20	6	2	50	2.50e-03	1.22e-12	1.37e-06	-4.08e-12	0
maxcut	10	10	2	50	2.50e-03	1.48e-12	5.90e-05	1.75e-12	0
etp	20	10	6	50	1.00e-03	2.24e-06	8.00e+00	-4.49e-06	-1
lovasz	10	28	1	50	5.00e-03	8.85e-12	6.18e-05	1.00e-11	0
log cheby	60	6	2	50	2.50e-03	2.17e-11	1.59e-07	5.55e-12	0

Theorem 4.4 *Under the assumptions of Proposition 4.2, every accumulation point X^* of a sequence $\{X^k\}$ generated by Algorithm 4.1 is a stationary point of the original program* (1).

Proof. Since the sequence $\{\alpha_k\}$ stays constant eventually, we may argue as in the analysis of Algorithm 2.1 that any accumulation point X^* of $\{X^k\}$ is a stationary point of the penalized problem (3), cf. Theorem 3.4. Hence there exist a vector $\xi^* \in \mathbb{R}^m$ and Lagrange multipliers $(\lambda^*, \mu^*, U^*) \in \mathbb{R}^m \times \mathbb{R}^m \times \mathcal{S}^{n\times n}$ that satisfy (5). Moreover, by Proposition 4.2, we have $\xi^* = 0$, which implies that (5) reduces to the KKT conditions (2) for the original program (1). This completes the proof. $\square$

5 Numerical Results

To test the numerical performance of Algorithm 4.1, we implemented the method in Matlab (Version 6.5) [20, 21] using the SDPT3-Solver (Version 3.0) for the corresponding subproblems. More details on the reformulation of the subproblem (19) as a *linear* semidefinite program with a second-order cone constraint are given in Appendix A.

The parameters in Algorithm 4.1 were set to the following values:

$$\alpha_0 = 50, \quad \delta = 25, \quad \rho_1 = 0.1, \quad \rho_2 = 0.75,$$
$$\sigma_1 = 0.5, \quad \sigma_2 = 2, \quad c_0 = 0.01, \quad c_{\min} = 0.001, \quad c_{\max} = 1.$$

We stopped the algorithm if either

$$\left\|\mathrm{svec}(\Delta X^k)\right\|_\infty < 10^{-4} \quad \text{and} \quad \left\|\xi^k\right\|_\infty < 10^{-4}$$

were satisfied or if SDPT3 failed to solve a subproblem correctly.

As preliminary experiments, we first tested the algorithm on six linear test examples from SDPT3 [20], by writing an interface to the special block structure of these examples. The computational results are shown in Table 1, where n denotes the dimension of $X \in \mathbb{R}^{n\times n}$, m is the number of constraints, and k denotes the number of iterations spent by the algorithm. When the SDPT3 solver could solve all subproblems, the exit code is zero. Otherwise, the negative SDPT3 exit code is shown to indicate a failure in solving a subproblem.

Table 1 shows that Algorithm 4.1 successfully solved all the test examples except problem `etp` for which the SDPT3 encountered an error when solving a subproblem because there

Table 2: Numerical results for nonlinear SDP

problem	n	m	k	α_k	c_k	$\|\xi_k\|_\infty$	$\left\|\mathrm{svec}(\Delta X^k)\right\|_\infty$	$\lambda_{\min}(X^k)$	SDPT3 exit
example 1	4	4	8	50	1.00e-03	3.18e-12	8.73e-05	3.32e-12	0
example 2	5	1	3	50	2.50e-03	0.00e+00	8.20e-07	-3.16e-08	0
example 3	5	6	3	100	1.00e-02	1.82e-09	6.39e-08	2.08e-10	0

was not enough progress in the predictor phase of SDPT3. Moreover, for the first four test examples in Table 1, Algorithm 4.1 obtained the same objective function value as the one produced by applying SDPT3 directly to these linear SDP examples, whereas for problem `log cheby`, the objective function value achieved by Algorithm 4.1 was slightly larger.

The main aim of our numerical experiments consists in examining the performance of the algorithm on nonlinear semidefinite programs. Unfortunately, however, there is no standard test problem library (such as SDPLIB [1] for linear SDPs) available for nonlinear SDPs. So we have constructed the following three nonlinear test problems by ourselves:

1. Nonlinear objective function with linear constraints: Let $n = 4$, $m = 4$ and
$$f(X) := \exp(-\operatorname{tr}(X)),$$
$$g(X) := (\operatorname{tr}(X) - 3, X(1,1) - 1, -X(1,2), X(3,3))^T.$$

2. Nonlinear objective function with linear constraints: Let $n = 5$, $m = 1$ and
$$f(X) := \cos(X(1,1)) + X(2,2) - \sin(X(3,3)) - X(4,4) + \exp(X(5,5)),$$
$$g(X) := \operatorname{tr}(X) - 100.$$

3. Nonlinear objective function with nonlinear constraints: Let $n = 5$, $m = 6$ and
$$f(X) := \exp(\operatorname{tr}(X)),$$
$$g(X) := (X(1,1), X(2,2)^3, -X(3,3) + 3, X(5,5) - 2, -2X(5,5) + 3, \operatorname{tr}(X) - 1000)^T.$$

The computational results for these three nonlinear test problems are shown in Table 2. As can be seen from Table 2, the algorithm was able to solve all three examples successfully. The number of iterations is relatively small for all three problems. Finally, note that the penalty parameter had to be updated twice for the third example.

6 Final Remarks

We introduced a successive linearization method for the solution of nonlinear semidefinite programs. Using an exact penalty function and a trust region-type globalization, the method is shown to be globally convergent under certain assumptions. Some preliminary numerical results indicate that the method works quite reasonable. Of course, further numerical experiments are necessary in order to get a more complete picture regarding the behaviour of our algorithm. Furthermore, we would like to weaken Assumptions (A.2) and (A.3) used in Section 4.

A Reformulation of Subproblems

In order to solve nonlinear semidefinite programs of the form (1) by Algorithm 4.1, we have to be able to deal with a subproblem given by

$$\begin{aligned}\min \quad & \frac{1}{2}c_k\langle \Delta X, \Delta X\rangle + f(X^k) + \langle Df(X^k), \Delta X\rangle \\ & + \alpha_k \sum_{i=1}^m \max\{0, g_i(X^k) + \langle Dg_i(X^k), \Delta X\rangle\} \\ \text{s.t.} \quad & \Delta X \in \mathcal{S}^{n\times n}, X^k + \Delta X \succeq 0,\end{aligned} \tag{23}$$

cf. (19). For this purpose, we would like to use the SDPT3 solver (version 3.0) from [20]. This software is designed to solve linear semidefinite programs with cone constraints of the form

$$\begin{array}{llllll}\min & \displaystyle\sum_{j=1}^{n_s}\langle C_j^s, X_j^s\rangle & + & \displaystyle\sum_{i=1}^{n_q}(c_i^q)^T x_i^q & + & (c^l)^T x^l \\ \text{s.t.} & \displaystyle\sum_{j=1}^{n_s}(A_j^s)^T \operatorname{svec}(X_j^s) & + & \displaystyle\sum_{i=1}^{n_q}(A_i^q)^T x_i^q & + & (A^l)^T x^l \;=\; b, \\ & X_j^s \in \mathcal{S}_+^{s_j\times s_j} \quad \forall j, & & x_i^q \in K_q^{q_i} \quad \forall i, & & x^l \in \mathbb{R}_+^{n_l},\end{array} \tag{24}$$

where C_j^s, X_j^s are symmetric matrices of dimension s_j, c_i^q, x_i^q are vectors in $\mathbb{R}^{q_i}$, $\mathcal{S}_+^{s_j\times s_j}$ denotes the s_j-dimensional positive semidefinite cone defined by $\mathcal{S}_+^{s_j\times s_j} := \{X \in \mathcal{S}^{s_j\times s_j} \ : \ X \succeq 0\}$, $K_q^{q_i}$ denotes the q_i-dimensional second-order cone defined by $K_q^{q_i} := \{x = (x_1, x_{2:q_i}^T)^T \in \mathbb{R}^{q_i} \ : \ x_1 \geq \|x_{2:q_i}\|\}$, c^l and x^l are vectors in $\mathbb{R}^{n_l}$, A_j^s are $\bar{s}_j \times m$ matrices with $\bar{s}_j = s_j(s_j+1)/2$, A_i^q and A^l are $q_i \times m$ and $l \times m$ matrices, respectively, and svec is the operator defined by $\operatorname{svec}(X) := (X(1,1), \sqrt{2}X(1,2), X(2,2), \sqrt{2}X(1,3), \sqrt{2}X(2,3), X(3,3), \dots)^T \in \mathbb{R}^{n(n+1)/2}$ for any symmetric matrix $X \in \mathcal{S}^{n\times n}$.

We now want to rewrite the problem (23) in the form (24). To this end, we need to make some reformulations, which will be described step by step in the following.

First, we drop the constant $f(X^k)$ from the objective function without affecting the problem. Next, we introduce the auxiliary variable $S \in \mathcal{S}^{n\times n}$ and set $X^k + \Delta X = S$. Because ΔX needs only to be symmetric and not to be positive semidefinite, we set $\Delta x = \operatorname{svec}(\Delta X)$ and write the problem in terms of $\Delta x \in \mathbb{R}^{\bar{n}}$ with $\bar{n} := n(n+1)/2$. Then problem (23) is equivalent to

$$\begin{aligned}\min \quad & \frac{1}{2}c_k \left\|\Delta x\right\|^2 + \operatorname{svec}(Df(X^k))^T \Delta x \\ & + \alpha_k \sum_{i=1}^m \max\{0, g_i(X^k) + \operatorname{svec}(Dg_i(X^k))^T \Delta x\} \\ \text{s.t.} \quad & \operatorname{svec}(X^k) + \Delta x = \operatorname{svec}(S), \\ & \Delta x \in \mathbb{R}^{\bar{n}}, \ S \succeq 0.\end{aligned} \tag{25}$$

By introducing the second-order cone constraint $\|\Delta x\| \leq t$, problem (25) can be further rewritten as

$$\begin{aligned} \min \quad & \frac{1}{2} c_k t^2 + \operatorname{svec}(Df(X^k))^T \Delta x \\ & + \alpha_k \sum_{i=1}^m \max\{0, g_i(X^k) + \operatorname{svec}(Dg_i(X^k))^T \Delta x\} \\ \text{s.t.} \quad & \operatorname{svec}(X^k) + \Delta x = \operatorname{svec}(S), \ \|\Delta x\| \leq t, \\ & \Delta x \in \mathbb{R}^{\bar{n}}, \ S \succeq 0, \ t \in \mathbb{R}. \end{aligned} \tag{26}$$

Unfortunately, the term t^2 is not linear as required in (24). So we replace t^2 by the new variable $s \geq 0$ and add the constraint $t^2 \leq s$. But this constraint can be rewritten as the semidefinite constraint

$$\begin{pmatrix} s & t \\ t & 1 \end{pmatrix} \succeq 0.$$

Introducing once again an auxiliary variable, problem (26) and hence the original subproblem (23) is equivalent to

$$\begin{aligned} \min \quad & \frac{1}{2} c_k s + \operatorname{svec}(Df(X^k))^T \Delta x \\ & + \alpha_k \sum_{i=1}^m \max\{0, g_i(X^k) + \operatorname{svec}(Dg_i(X^k))^T \Delta x\} \\ \text{s.t.} \quad & \Delta x - \operatorname{svec}(S) = -\operatorname{svec}(X^k), \ \|\Delta x\| \leq t, \\ & \begin{pmatrix} s & t \\ t & 1 \end{pmatrix} - W = 0, \\ & \Delta x \in \mathbb{R}^{\bar{n}}, \ W \succeq 0, \ S \succeq 0, \ t \in \mathbb{R}, \ s \in \mathbb{R}_+. \end{aligned} \tag{27}$$

In the next step, we replace the max-terms in the objective function by auxiliary variables ξ_i. This leads us to the following problem formulation:

$$\begin{aligned} \min \quad & \frac{1}{2} c_k s + \operatorname{svec}(Df(X^k))^T \Delta x + \alpha_k \sum_{i=1}^m \xi_i \\ \text{s.t.} \quad & \Delta x - \operatorname{svec}(S) = -\operatorname{svec}(X^k), \ \|\Delta x\| \leq t, \\ & \begin{pmatrix} s & t \\ t & 1 \end{pmatrix} - W = 0, \\ & \xi_i - g_i(X^k) - \operatorname{svec}(Dg_i(X^k))^T \Delta x \geq 0 \quad i = 1, \dots, m, \\ & \Delta x \in \mathbb{R}^{\bar{n}}, \ W \succeq 0, \ S \succeq 0, \ t \in \mathbb{R}, \ s \in \mathbb{R}_+, \ \xi = (\xi_1, \dots, \xi_m)^T \in \mathbb{R}^m_+. \end{aligned} \tag{28}$$

Once again, we rewrite the inequality constraints as equalities by setting $\omega_i = \xi_i - g_i(X^k) - \operatorname{svec}(Dg_i(X^k))^T \Delta x$. Moreover, we write the equality constraint

$$\begin{pmatrix} s & t \\ t & 1 \end{pmatrix} - W = 0$$

in the svec-notation. Then we get

$$\begin{aligned}
\min \quad & \frac{1}{2}c_k s + \operatorname{svec}(Df(X^k))^T \Delta x + \alpha_k \sum_{i=1}^{m} \xi_i \\
\text{s.t.} \quad & \Delta x - \operatorname{svec}(S) = -\operatorname{svec}(X^k), \ \|\Delta x\| \le t, \\
& \begin{pmatrix} s \\ \sqrt{2}t \\ 1 \end{pmatrix} - \operatorname{svec}(W) = 0, \\
& \xi_i - g_i(X^k) - \operatorname{svec}(Dg_i(X^k))^T \Delta x - \omega_i = 0 \quad i = 1, \dots, m, \\
& \Delta x \in \mathbb{R}^{\bar{n}}, \ W \succeq 0, \ S \succeq 0, \ t \in \mathbb{R}, \ s \in \mathbb{R}_+, \\
& \xi = (\xi_1, \dots, \xi_m)^T \in \mathbb{R}^m_+, \ \omega = (\omega_1, \dots, \omega_m)^T \in \mathbb{R}^m_+.
\end{aligned} \tag{29}$$

We are now in a position to give the explicit correspondence between the parameters, variables and input data in our last problem formulation (29) and those of the SDPT3 standard form (24). The problem parameters are given by

$$n_s := 2, \ n_q := 1, \ s_1 := n, \ s_2 := 2, \ q_1 := 1 + \bar{n}, \ l := 1 + 2m.$$

The variables are given by

$$\begin{aligned}
X_1^s &:= S \in \mathcal{S}_+^{n \times n}, \\
X_2^s &:= W \in \mathcal{S}_+^{2 \times 2}, \\
x_1^q &:= (t, \Delta x^T)^T \in K_q^{1+\bar{n}}, \\
x^l &:= (s, \xi^T, \omega^T)^T \in \mathbb{R}^{1+2m}.
\end{aligned}$$

The input data in the objective function are given by

$$\begin{aligned}
C_1^s &:= 0 \in \mathcal{S}^{n \times n}, \\
C_2^s &:= 0 \in \mathcal{S}^{2 \times 2}, \\
c_1^q &:= \big(0, \operatorname{svec}(Df(X^k))^T\big)^T \in \mathbb{R}^{1+\bar{n}}, \\
c^l &:= \big(\tfrac{1}{2}c_k, \alpha_k e, 0\big) \in \mathbb{R}^{1+2m}
\end{aligned}$$

with

$$e = (1, \dots, 1)^T \in \mathbb{R}^m.$$

Finally, the matrices $A_1^s \in \mathbb{R}^{\bar{n} \times (\bar{n}+3+m)}$, $A_2^s \in \mathbb{R}^{3 \times (\bar{n}+3+m)}$, $A_1^q \in \mathbb{R}^{(1+\bar{n}) \times (\bar{n}+3+m)}$, $A^l \in$

$\mathbb{R}^{(1+m+m)\times(\bar{n}+3+m)}$ and the vector $b \in \mathbb{R}^{\bar{n}+3+m}$ are given by

$$
\begin{aligned}
A_1^s &= \left(\begin{array}{c|c|c} -I & 0 & 0 \end{array} \right), \\
A_2^s &= \left(\begin{array}{c|c|c} 0 & -I & 0 \end{array} \right), \\
A_1^q &= \left(\begin{array}{c|ccc|ccc}
0 & 0 & \sqrt{2} & 0 & 0 & \cdots & 0 \\ \hdashline
 & 0 & 0 & 0 & & & \\
I & \vdots & \vdots & \vdots & -\operatorname{svec}(Dg_1(X^k)) & \cdots & -\operatorname{svec}(Dg_m(X^k)) \\
 & 0 & 0 & 0 & & &
\end{array} \right), \\
A^l &= \left(\begin{array}{c|ccc|ccc}
0 & 1 & 0 & 0 & 0 & \cdots & 0 \\ \hdashline
0 & & 0 & & & I & \\ \hdashline
0 & & 0 & & & -I &
\end{array} \right), \\
b &= \left(\begin{array}{c|ccc|c} -\operatorname{svec}(X^k)^T & 0 & 0 & -1 & g(X^k)^T \end{array} \right)^T.
\end{aligned}
$$

This is the desired reformulation.

It may be worth mentioning that problem (25) can also be reformulated as

$$
\begin{aligned}
\min \quad & \frac{1}{2}c_k t + \operatorname{svec}(Df(X^k))^T \Delta x + \alpha_k \sum_{i=1}^{m} \max\{0, g_i(X^k) + \operatorname{svec}(Dg_i(X^k))^T \Delta x\} \\
\text{s.t.} \quad & \operatorname{svec}(X^k) + \Delta x = \operatorname{svec}(S), \ \|\Delta x\|^2 \le t, \\
& \Delta x \in \mathbb{R}^{\bar{n}}, \ S \succeq 0, \ t \in \mathbb{R}.
\end{aligned}
\tag{30}
$$

Since the constraint $\|\Delta x\|^2 \le t$ is equivalent to

$$
\begin{pmatrix} t & \Delta x^T \\ \Delta x & I \end{pmatrix} \succeq 0,
$$

problem (30) can further be reformulated as a linear semidefinite program that involves a semidefinite cone constraint instead of a second-order cone constraint. However, such a semidefinite representation is much more expensive in terms of memory requirement. Therefore we adopted the reformulation (29) in our numerical experiments reported in Section 5.

References

[1] B. Borchers: *SDPLIB 1.2, A library of semidefinite programming test problems.* Optimization Methods and Software 11, 1999, pp. 597–611.

[2] R. Correa and H. Ramirez: *A global algorithm for nonlinear semidefinite programming.* Research Report 4672, INRIA, Le Chesnay Cedex, France, 2002.

[3] B. Fares, D. Noll and P. Apkarian: *Robust control via sequential semidefinite programming.* SIAM Journal on Control and Optimization 40, 2002, pp. 1791–1820.

[4] A. Forsgren: *Optimality conditions for nonconvex semidefinite programming*, Mathematical Programming 88, 2000, pp. 105–128.

[5] R.W. Freund and F. Jarre: *A sensitivity analysis and a convergence result for a sequential semidefinite programming method.* Technical Report, Bell Laboratories, Murray Hill, New Jersey, 2003.

[6] M. Fukuda and M. Kojima: *Branch-and-cut algorithms for the bilinear matrix inequality eigenvalue problem.* Computational Optimization and Applications 19, 2001, pp. 79–105.

[7] M. Fukushima, K. Takazawa, S. Ohsaki and T. Ibaraki: *Successive linearization methods for large-scale nonlinear programming problems.* Japan Journal of Industrial and Applied Mathematics 9, 1992, pp. 117–132.

[8] R.E. Griffith and R.A. Stewart: *A nonlinear programming technique for the optimization of continuous processing systems.* Management Science 7, 1961, pp. 379–392.

[9] C.W.J. Hol, C.W. Scherer, E.G. van der Meché and O.H. Bosgra: *A nonlinear SDP approach to fixed-order controller synthesis and comparison with two other methods applied to an active suspension system.* European Journal of Control 9, 2003, pp. 11–26.

[10] X.X. Huang, K.L. Teo and X.Q. Yang: *Approximate augmented Lagrangian functions and nonlinear semidefinite programs.* Technical Report, Department of Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong, China, 2003.

[11] F. Jarre: *An interior method for nonconvex semidefinite programs.* Optimization and Engineering 1, 2000, pp. 347–372.

[12] F. Jarre: *Some aspects of nonlinear semidefinite programs.* System Modeling and Optimization XX, F.W. Sachs and R. Tichatschke (eds.), Kluwer Academic Publishers, 2003.

[13] M. Kočvara and M. Stingl: *PENNON: A code for convex nonlinear and semidefinite programming.* Optimization Methods and Software 18, 2003, pp. 317–333.

[14] F. Leibfritz: *An LMI-based algorithm for designing suboptimal static $\mathcal{H}_2/\mathcal{H}_\infty$ output feedback controllers.* SIAM Journal on Control and Optimization 39, 2001, pp. 1711–1735.

[15] Z.-Q. Luo: *Optimal transceiver design via convex programming.* Technical Report, Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada, 1999.

[16] F. Palacios-Gomez, L. Lasdon and M. Engquist: *Nonlinear optimization by successive linear programming.* Management Science 28, 1982, pp. 1106–1120.

[17] A. Shapiro: *First and second order analysis of nonlinear semidefinite programs.* Mathematical Programming 77, 1997, pp. 301–320.

[18] J.F. Sturm: *Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones.* Optimization Methods and Software 11–12, 1999, pp. 625–653.

[19] M.J. Todd: *Semidefinite optimization.* Acta Numerica 10, 2001, pp. 515–560.

[20] K.C. Toh, R.H. Tütüncü and M.J. Todd: *SDPT3 version 3.02 – a MATLAB software for semidefinite-quadratic-linear programming.* updated in December 2002, http://www.math.nus.edu.sg/~mattohkc/sdpt3.html.

[21] R.H. Tütüncü, K.C. Toh and M.J. Todd: *Solving semidefinite-quadratic-linear programs using SDPT3.* Mathematical Programming 95, 2003, pp. 189–217.

[22] J. Zhang, N.-H. Kim and L. Lasdon: *An improved successive linear programming algorithm.* Management Science 31, 1985, pp. 1312–1331.