Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 19, No. 1, pp. 1–20, 2023

AN EFFICIENT PARAMETERIZED LOGARITHMIC KERNEL FUNCTION FOR SEMIDEFINITE OPTIMIZATION

Zerari Amina, Benterki Djamel and Yassine Adnan

Abstract: In this work, we propose a primal-dual interior point algorithm for semidefinite optimization SDO based on a new kernel function with an efficient logarithmic barrier term. We show that the best result of iteration bounds can be achieved, namely $O(\sqrt{n}\log n\log\frac{n}{\varepsilon})$, for large-update and $O(\sqrt{n}\log\frac{n}{\varepsilon})$ for small-update methods, which improves significantly the so far obtained complexity results based on a logarithmic kernel function for SDO namely $O(n\log\frac{n}{\varepsilon})$, for large-update and $O(\sqrt{n}\log\frac{n}{\varepsilon})$ for small-update methods. Some numerical tests are reported to show the efficiency of the algorithm.

Key words: *semidefinite optimization, Kernel function, primal-dual interior point method, large and small-update methods, complexity bound*

Mathematics Subject Classification: *90C22, 30C40, 90C51*

1 Introduction

It is well known that the semidefinite optimization (SDO) problems have been changed as one of the most active research areas in mathematical programming due to their wide in the real world problems as in engineering problems, such as optimal control, combinatorics, image processing, sensor networks, financial mathematics and statistics [23]. The primal-dual interior-point method (IPM) is one of the most efficient numerical methods for solving large classes of optimization problems and highly efficient in both theory and practice. All the IPMs designed for linear optimization (LO) problems have been successfully extended to SDO problems and are widely used for solving these problems.

After the seminal paper of Karmarkar [14] in 1984, the polynomial time IPMs have been revitalized as an active area of research. Since then, many variants of this algorithm have been studied and developed for the many classes of convex optimization problems, including Linear Complementary Problem (LCP), Second Order Cone Optimization (SOCO) problems and Semidefinite Optimization (SDO) problems. These methods have shown their power not only in theoretical complexity results but also in practical performance. Due to these nice behaviors, nowadays, IPMs are of great interest for researchers in the optimization fields.

All variants of the interior point methods designed for Linear Optimization (LO) have been successfully extended to SDO. An extension of the primal-dual IPMs from LO to SDO was first done by Nesterov and Nemirovski [16] and obtained a polynomial complexity for solving conic problems by introducing the so called self concordant barrier functions which consist of the logarithmic barrier function. Peng et al. [18] proposed a new paradigm in

© 2023 Yokohama Publishers

the classical IPM for solving LO and some other extensions of this problem in which the logarithmic barrier function is replaced by the so called Self-Regular (SR) barrier functions. The iteration complexity of LO and its extensions, based on SR barrier functions, led them to obtain the so far best known iteration bound for small and large update IPMs as $O(\sqrt{n}\log\frac{n}{\varepsilon})$ and $O(\sqrt{n}\log n\log\frac{n}{\varepsilon})$, respectively. Note that, based on the logarithmic barrier functions, these bounds are $O(\sqrt{n}\log\frac{n}{\varepsilon})$ and $O(n\log\frac{n}{\varepsilon})$, respectively [19]. Moreover, a class of primal-dual interior-point algorithms for LO based on a new family of kernel functions which is fairly general and includes the classical logarithmic function, the prototype self-regular function and some non-self-regular kernel functions as its special case was proposed by Bai et al. in [3]. Many researchers have proposed different interior point algorithms based on kernel function for various optimization problems. Roos et al. [2] defined eligible kernel functions and proposed a primal-dual IPM for LO and simplified the complexity analysis of Peng et al. in [18]. EL Ghami et al. [10] proposed a primal-dual interior-point algorithm for LO based on a kernel function with a trigonometric barrier term but they didn't obtain the best known complexity result for large update method. Bouafia et al. [5] proposed a primal-dual interior point algorithm for LO based on a kernel function with a trigonometric barrier term and they obtained the best known complexity result for small and large update method.

Several interior point methods (IPMs) for LO have been successfully extended to SDO. Wang et al. [21] proposed a primal-dual IPM for SDO based on a generalized version of the kernel function in [2] and obtained $O(q^2\sqrt{n}\log\frac{n}{\varepsilon})$, $q>1$, and $O(\sqrt{n}\log n\log\frac{n}{\varepsilon})$ complexity results for small and large update methods, respectively. EL Ghami et al. [9] extended the IPM for LO in [2] to SDO and obtained the similar iteration bounds as analogue of LO. EL Ghami et al. [12] proposed a primal dual IPM for SDO based on a generalized version of the kernel function given in [1] and obtained $O(\sqrt{n}\log n\log\frac{n}{\varepsilon})$ for large update method. Lee et al. [15] defined a new class of kernel functions and obtained the best known complexity results of the small and large update IPMs based on the kernel function for LO and SDO. EL Ghami [8] generalize the analysis presented in [5] for SDO and obtained the best known complexity results for the small and large update method. Motivated by their works, we proposed a primal-dual interior point algorithm for SDO based on a new kernel function and obtained the best known complexity results for small and large update methods.

This paper is organized as follows: In section 2, we start with some notations and preliminaries. In section 3, we present the kernel function based on Nesterov-Todd direction and describe the generic primal dual algorithm. In section 4, a new kernel function and its growth properties for SDO are studied. In section 5, we present the growth behavior of the proximity function based on this kernel function. Then, the estimate of the step size and the decrease behavior of the new kernel function are discussed. Also, the inner iteration bound and the iteration bound of the algorithm are given in this section. In section 6, we illustrate the practical performance of the new proposed kernel function. Finally, a conclusion ends section 7.

2 Notations and Preliminaries

2.1 Notions

We will make use of the following notations throughout the paper: $\mathbb{R}^n$, $\mathbb{R}^n_+$ and $\mathbb{R}^n_{++}$ denote the set of real, nonnegative real and positive real vectors with n components, respectively. $\mathbb{S}^n$, $\mathbb{S}^n_+$ and $\mathbb{S}^n_{++}$ denote the set of symmetric, symmetric positive semidefinite and symmetric positive definite $n\times n$ matrices, respectively. $||.||$ denotes the Frobenius norm for matrices. $\succeq$ denotes the nonnegativity in the Löwner partial order for symmetric matrices, i.e., $A\succeq B$

($A \succ B$) if $A - B$ is symmetric positive semidefinite (positive definite). Given $A \in \mathbb{R}^{n\times n}$, $Tr(A)$ stands for the trace of the matrix A. For $A, B \in \mathbb{R}^{m\times n}$, the inner product is defined by $A \bullet B = Tr(AB^T)$. For $Q \in \mathbb{S}^n_{++}$, $Q^{1/2}$ denotes the symmetric square root of Q. For any $V \in \mathbb{S}^n$, we denote by $\lambda(V)$ the vector of eigenvalues of V arranged in non-increasing order, that is, $\lambda_1(V) \geq \lambda_2(V) \geq ... \geq \lambda_n(V)$ and $\Lambda = diag(\lambda(V))$, i.e., the diagonal matrix from a vector $\lambda(V)$. I denotes an $n \times n$ identity matrix. For $f(x), g(x) : \mathbb{R}_{++} \to \mathbb{R}_{++}$, $f(x) = O(g(x))$ if $f(x) \leq c_1 g(x)$ for some positive constant c_1 and $f(x) = \Theta(g(x))$ if $c_2 g(x) \leq f(x) \leq c_3 g(x)$ for some positive constants c_2 and c_3.

2.2 Preliminaries

In this section, we recall some basic concepts and we drive the classical Nestrov-Todd search direction for SDO. Consider the standard semidefinite programming (SDP) problem:

$$\min \{C \bullet X : A_i \bullet X = b_i, \ \forall i = 1, ..., m, \ X \succeq 0\} \tag{P}$$

and its dual problem:

$$\max\{b^T y : S = C - \sum_{i=1}^{m} y_i A_i, \ S \succeq 0\} \tag{D}$$

where $C, A_i \in \mathbb{S}^n, i = 1, ..., m, \, b, y \in \mathbb{R}^m$.

Throughout the paper, we assume that the matrices A_i, $i = 1, ..., m$, are linearly independent and the problems (P) and (D) satisfy the interior-point condition (IPC), i.e., there exist $X \in F_P, S \in F_D$ with $X \succ 0, S \succ 0$, where F_P and F_D denote the feasible sets of the problem (P) and (D), respectively.

Finding an optimal solution of the problem (P) and (D) is equivalent to solving the following system:

$$\begin{cases} A_i \bullet X = b_i, \quad i = 1, ..., m, \ X \succeq 0, \\ \sum\limits_{i=1}^{m} y_i A_i + S = C, \ S \succeq 0, \\ XS = 0. \end{cases} \tag{2.1}$$

The basic idea of primal dual IPMs is to replace the complementarity condition of (2.1), $XS = 0$, by the parametrized equation $XS = \mu I$ with $X, S \succ 0$ and $\mu > 0$. So we consider the following system:

$$\begin{cases} A_i \bullet X = b_i, \quad i = 1, ..., m, \ X \succ 0, \\ \sum\limits_{i=1}^{m} y_i A_i + S = C, \ S \succ 0, \\ XS = \mu I. \end{cases} \tag{2.2}$$

If IPC holds, then the system (2.2) has a unique solution $(X(\mu), y(\mu), S(\mu))$ for each $\mu > 0$ called the $\mu - center$ of both problems (P) and (D). The set of $\mu - center$ defines a homotopy called the central path of (P) and (D) which converge to the optimal solution of the problem (P) and (D) as μ goes to zero [23]. Now, to compute the search direction, we apply Newton's method to the system (2.2), we obtain the Newton system as follows:

$$\begin{cases} A_i \bullet \Delta X = 0, \quad i = 1, ..., m, \\ \sum\limits_{i=1}^{m} \Delta y_i A_i + \Delta S = 0, \\ X\Delta S + \Delta X S = \mu I - XS. \end{cases} \tag{2.3}$$

Since A_i are linearly independent and $X \succ 0, S \succ 0$, the system (2.3) has a unique search direction $(\Delta X, \Delta y, \Delta S)$. Note that ΔS is symmetric from the second equation of (2.3), but ΔX may not be symmetric. Various methods of symmetrizing the third equation of (2.3) are proposed, so that the new system has a unique symmetric solution. In this paper, we use the NT symmetrizing scheme [17].

Let

$$P = X^{1/2}(X^{1/2}SX^{1/2})^{-1/2}X^{1/2} = S^{-1/2}(S^{1/2}XS^{1/2})^{1/2}S^{-1/2}$$

and $D = P^{1/2}$ where $P^{1/2}$ denotes the symmetric square root of P. The matrix D is used to scale both matrices X and S to the same matrix V defined by

$$V = \frac{1}{\sqrt{\mu}}D^{-1}XD^{-1} = \frac{1}{\sqrt{\mu}}DSD = \frac{1}{\sqrt{\mu}}(D^{-1}XSD)^{1/2}. \tag{2.4}$$

Then, matrices D and V are symmetric positive definite. By using (2.4) the Newton system (2.3) can be rewritten as follows:

$$\begin{cases} \overline{A}_i \bullet D_X = 0, \quad i = 1, ..., m, \\ \sum\limits_{i=1}^{m} \Delta y_i \overline{A}_i + D_S = 0, \\ D_X + D_S = V^{-1} - V. \end{cases} \tag{2.5}$$

with

$$\overline{A}_i = \frac{1}{\sqrt{\mu}}DA_iD,\ i = \overline{1, m},\ D_X = \frac{1}{\sqrt{\mu}}D^{-1}\Delta XD^{-1},\ D_S = \frac{1}{\sqrt{\mu}}D\Delta SD. \tag{2.6}$$

The system (2.5) determines a uniquely symmetric NT direction with the matrices D_X and D_S be orthogonal, and it is evident to see

$$Tr(D_XD_S) = Tr(D_SD_X) = 0.$$

The above NT direction leads to the classical primal-dual IPM algorithms for SDO.

3 Generic Primal-Dual Algorithm for SDO

In this section, we recall the definition of a matrix function and we introduce our generic primal-dual IPM algorithm for SDO.

First of all, we recall the definition of a kernel function.

We call $\psi : \mathbb{R}_{++} \to \mathbb{R}_+$ a kernel function if it is twice differentiable and satisfies the following conditions:

$$\psi'(1) = \psi(1) = 0,\ \psi''(t) > 0,\ t > 0,\ \lim_{t\to 0^+} \psi(t) = \lim_{t\to\infty} \psi(t) = \infty. \tag{3.1}$$

For $V = Q^T diag(\lambda_1(V), \lambda_2(V), ..., \lambda_n(V))Q$, the spectral decomposition of $V \in \mathbb{S}^n_{++}$, we generalize a function $\psi(t) : \mathbb{R}_{++} \to \mathbb{R}_+$ to the matrix function $\psi(V) : \mathbb{S}^n_{++} \to \mathbb{S}^n$ as follows:

$$\begin{aligned} \psi(V) &= Q^T diag(\psi(\lambda_1(V)), \psi(\lambda_2(V)), ..., \psi(\lambda_n(V)))Q, \qquad (3.2)\\ \psi'(V) &= Q^T diag(\psi'(\lambda_1(V)), \psi'(\lambda_2(V)), ..., \psi'(\lambda_n(V)))Q. \end{aligned}$$

Replacing the right hand side $V^{-1} - V$ of the third equation of (2.5) by $-\psi'(V)$. Then we have the linear system:

$$\begin{cases} \overline{A}_i \bullet D_X = 0, \quad i = 1, ..., m, \\ \sum\limits_{i=1}^{m} \Delta y_i \overline{A}_i + D_S = 0, \\ D_X + D_S = -\psi'(V). \end{cases} \tag{3.3}$$

where $\psi(t)$ is a given kernel function, and $\psi(V), \psi'(V)$ are the associated matrix functions denote in (3.2), the system (3.3) has a unique symmetric solution.

For any kernel function $\psi(t)$, we define the barrier function $\Psi(V) : \mathbb{S}^n_{++} \to \mathbb{R}_+$ as follows

$$\Psi(V) = Tr(\psi(V)) = \sum_{i=1}^{n} \psi(\lambda_i(V)). \tag{3.4}$$

Then $\Psi(V)$ is strictly convex with respect to $V \succ 0$ and vanishes at its global minimal point $V = I$ and $\Psi(I) = 0$. Since D_X and D_S are orthogonal, for $\mu > 0$,

$$\Psi(V) = 0 \Leftrightarrow V = I \Leftrightarrow D_X = D_S = 0 \Leftrightarrow X = X(\mu), S = S(\mu).$$

We use $\Psi(V)$ as a proximity function to measure the distance between the current iterate and the $\mu-$center for given $\mu > 0$.

We can describe one step of the primal-dual interior-point algorithm for SDO based on kernel functions as follow: Starting with a strictly feasible point (X_0, y_0, S_0) which is in a $\tau-$neighborhood of the given $\mu-$center. We begin the outer iteration which aims to increase the value of $\Psi(V)$ above τ; (i.e., $\Psi(V) > \tau$) by decreasing the parameter μ as $\mu = (1-\theta)\mu$. After that we begin the inner iterations which are performed to bring back the situation where $\Psi(V) \leq \tau$. In each inner iteration we compute a new candidate point by applying Newton's method, we repeat this process until $\Psi(V) \leq \tau$. Then, the algorithm performs outer iterations until μ is small enough; i.e., $n\mu < \varepsilon$, at this stage, we have found an $\varepsilon-$ approximate solutions of SDO. The generic form of this algorithm is outlined as follow:

Primal-Dual Algorithm for SDO

```
Input
A threshold parameter $\tau \geq 1$;
an accuracy parameter $\varepsilon > 0$;
a fixed barrier update parameter $\theta$, $0 < \theta < 1$;
a strictly feasible $(X^0, S^0)$ and $\mu^0 = 1$ such that $\Psi(X^0, S^0, \mu^0) \leq \tau$;
begin
    $X = X^0; S = S^0; \mu = \mu^0$;
    while $n\mu > \varepsilon$ do
    begin (outer iteration)
        $\mu = (1-\theta)\mu$;
        while $\Psi(X, S, \mu) > \tau$ do
        begin (inner iteration)
            solve the system (3.3) and (2.6) to obtain $\Delta X$, $\Delta y$, $\Delta S$;
            determine a step size $\alpha$; and take
            $X = X + \alpha\Delta X$;
            $y = y + \alpha\Delta y$;
            $S = S + \alpha\Delta S$;
        end (inner iteration)
    end (outer iteration)
end.
```

$\boxed{4}$ New Kernel Function and its Properties

In this section, we introduce our new kernel function and some useful properties for our complexity analysis.

For $p \in \mathbb{R}$, $q \in \mathbb{R}$, we define our new kernel function $\psi(t)$ as follows:

$$\psi(t) = \frac{p(t^2-1)}{2} + \frac{p(t^{-pq+1}-1)}{(pq-1)(q+1)} - \frac{pq}{(q+1)} \log t,\ p \geq 1, q > 1, t > 0. \tag{4.1}$$

It is easy to check that $\psi(t)$ is indeed a barrier kernel function and its first, second and third derivatives are as follows:

$$\begin{aligned} \psi'(t) &= pt - \frac{p}{q+1} t^{-pq} - \frac{pq}{(q+1)} t^{-1}, \qquad (4.2) \\ \psi''(t) &= p + \frac{p(pq)}{q+1} t^{-pq-1} + \frac{pq}{(q+1)} t^{-2}, \\ \psi'''(t) &= -\frac{p(pq)(pq+1)}{q+1} t^{-pq-2} - \frac{2pq}{(q+1)} t^{-3}. \end{aligned}$$

$\psi(t)$ is expressed in term of its second derivative as follows:

$$\psi(t) = \int_1^t \int_1^\xi \psi''(\zeta) d\zeta d\xi. \tag{4.3}$$

From (4.2), we have

$$\psi''(t) > p, \quad \forall t > 0. \tag{4.4}$$

Now, we provide some properties of our kernel function which are used in the complexity analysis.

Lemma 4.1. *Let $\psi(t)$ be defined as in* (4.1)*, then*

(i) $t\psi''(t) + \psi'(t) > 0, \quad 0 < t < 1,$

(ii) $t\psi''(t) - \psi'(t) > 0, \quad t > 1,$

(iii) $\psi'''(t) < 0, \ t > 0.$

Proof. For (i), using (4.2), it follows that $t\psi''(t)+\psi'(t) = 2pt + \frac{p(pq-1)t^{-pq}}{q+1} > 0, \forall p \geq 1,\ q > 1$ and $t > 0$.

For (ii), we have $t\psi''(t) - \psi'(t) = \frac{p(pq+1)t^{-pq}}{q+1} + \frac{2pq}{(q+1)t} > 0$, for all $p \geq 1,\ \ q > 1$ and $t > 0$.

For (iii), it is clear from (4.2) that $\psi'''(t) < 0$, for $t > 0$. □

Remark 4.2 (Lemma 2.4 in [2])**.** If $\psi(t)$ satisfy (ii) and (iii) in Lemma 4.1, then

$$\psi''(t)\psi'(\beta t) - \beta\psi'(t)\psi''(\beta t) > 0, \quad t > 0, \quad \beta > 1.$$

Lemma 4.3. *For $\psi(t)$, we have for $p \geq 1$ and $q > 1$,*

(i) $\frac{p}{2}(t-1)^2 \leq \psi(t) \leq \frac{1}{2p}[\psi'(t)]^2, \ t > 0,$

(ii) $\psi(t) \leq \frac{1}{2}\psi''(1)(t-1)^2, \ t \geq 1.$

Proof. For (i), using the first condition of (3.1) and (4.4), we have

$$\psi(t) = \int_1^t \int_1^\xi \psi''(\zeta) d\zeta d\xi \geq \int_1^t \int_1^\xi p d\zeta d\xi = \frac{p}{2}(t-1)^2,$$

the second inequality is obtained as follows:

$$\begin{aligned}\psi(t) &= \int_1^t \int_1^\xi \psi''(\zeta)d\zeta d\xi \\ &\leq \frac{1}{p}\int_1^t \int_1^\xi \psi''(\xi)\psi''(\zeta)d\zeta d\xi \\ &= \frac{1}{p}\int_1^t \psi''(\xi)\psi'(\xi)d\xi \\ &= \frac{1}{p}\int_1^t \psi'(\xi)d\psi'(\xi) = \frac{1}{2p}[\psi'(t)]^2.\end{aligned}$$

For (ii), using Taylor's theorem, the first condition of (3.1) and Lemma 4.1 (*iii*), we have

$$\begin{aligned}\psi(t) &= \psi(1) + \psi'(1)(t-1) + \frac{1}{2}\psi''(1)(t-1)^2 + \frac{1}{3!}\psi'''(c)(t-1)^3 \\ &= \frac{1}{2}\psi''(1)(t-1)^2 + \frac{1}{3!}\psi'''(c)(t-1)^3 \\ &< \frac{1}{2}\psi''(1)(t-1)^2,\end{aligned}$$

for some c, such that $1 \leq c \leq t$. This completes the proof. □

Lemma 4.4. *Let $\varrho : [0,\infty) \to [1,\infty)$ be the inverse function of $\psi(t)$ for $t \geq 1$. Then we have*

$$1 + \sqrt{\frac{2s}{\psi''(1)}} \leq \varrho(s) \leq 1 + \sqrt{\frac{2s}{p}}, \quad p \geq 1,\ q > 1,\ s \geq 0.$$

Proof. Let $s = \psi(t)$ for $t \geq 1$, i.e., $\varrho(s) = t$. By the definition of $\psi(t)$, $s = \frac{p(t^2-1)}{2} + \frac{t^{-pq+1}-1}{q(q+1)} - \frac{pq}{(q+1)}\log t,\ \ p \geq 1,\ q > 1,\ t > 0$. Using Lemma 4.3 ($i$), we have $s = \psi(t) \geq \frac{p}{2}(t-1)^2$, which implies that $t = \varrho(s) \leq 1 + \sqrt{\frac{2s}{p}}$.

For the second inequality, using Lemma 4.3 (ii), then
$s = \psi(t) \leq \frac{1}{2}\psi''(1)(t-1)^2,\ \ t \geq 1$. It follows that $t = \varrho(s) \geq 1 + \sqrt{\frac{2s}{\psi''(1)}}$. This completes the proof. □

Lemma 4.5. *Let $\rho : [0,\infty) \to (0,1]$ be the inverse function of $-\frac{1}{2}\psi'(t)$ for $0 < t \leq 1$. Then we have*

$$\rho(z) \geq \left(\frac{p}{2(q+1)z+p}\right)^{\frac{1}{pq}}, \qquad p \geq 1,\ q > 1,\ z \geq 0.$$

Proof. Let $z = -\frac{1}{2}\psi'(t)$ for $0 < t \leq 1$. By the definition of ρ, $\rho(z) = t$, for $z \geq 0$. So, we have $z = \frac{1}{2}(\frac{p}{q+1}t^{-pq} - pt + \frac{pq}{(q+1)}t^{-1}) \geq \frac{1}{2}(\frac{p}{q+1}t^{-pq} - p + \frac{pq}{q+1})$, it follows that $t^{-pq} \leq \frac{2(q+1)z+p}{p}$. Hence, we obtain $t = \rho(z) \geq \left(\frac{p}{2(q+1)z+p}\right)^{\frac{1}{pq}}$. This completes the proof. □

To derive some other properties of our new kernel function, we define the norm-based proximity measure $\delta(V)$ as below:

$$\delta(V) = \frac{1}{2}||\psi'(V)|| = \frac{1}{2}\sqrt{\sum_{i=1}^{n}(\psi'(\lambda_i(V)))^2} = \frac{1}{2}||D_X + D_S||, \quad V \in \mathbb{S}^n_{++}. \tag{4.5}$$

In the following lemma, we drive a lower bound for $\delta(V)$ in term of $\Psi(V)$.

Lemma 4.6. *Let $\delta(V)$ and $\Psi(V)$ be defined as in* (4.5) *and* (3.4)*, respectively. Then, we have*

$$\delta(V) \geq \sqrt{\frac{p}{2}\Psi(V)}, \quad V \in \mathbb{S}^n_{++}.$$

Proof. Using (4.5) and the second inequality of Lemma 4.3 (i),

$$\delta^2(V) = \frac{1}{4}\sum_{i=1}^{n}(\psi'(\lambda_i(V)))^2 \geq \frac{p}{2}\sum_{i=1}^{n}\psi(\lambda_i(V)) = \frac{p}{2}\Psi(V).$$

Hence, we have $\delta(V) \geq \sqrt{\frac{p}{2}\Psi(V)}$. This completes the proof. □

Remark 4.7. Throughout the paper, we assume that $\tau \geq 1$. Using Lemma 4.6 and the assumption that $\Psi(V) \geq \tau$, we have $\delta(V) \geq \frac{1}{\sqrt{2}}$.

5 Complexity Analysis

This section presents the growth behavior of the proximity function after a $\mu-$update. Then we compute a default step size α and the resulting decrease of the barrier function after an inner iteration.

The following lemmas provide an upper bound for the growth of the proximity after a $\mu-$update.

Lemma 5.1 (Lemma 4.16 in [22])**.** *Let ϱ be defined as in Lemma* 4.4*. Then we have*

$$\Psi(\beta V) \leq n\psi\left(\beta\varrho\left(\frac{\Psi(V)}{n}\right)\right), \quad V \in \mathbb{S}^n_{++}, \quad \beta \geq 1.$$

Lemma 5.2. *Let $0 \leq \theta < 1$ and $V_+ = \frac{V}{\sqrt{1-\theta}}$. If $\Psi(V) \leq \tau$, then for $q > 1$ we have*

$$\Psi(V_+) \leq \frac{np\theta + 2\tau + 2\sqrt{2\tau np}}{2(1-\theta)}.$$

Proof. For $t \geq 1$, we have

$$\psi(t) \leq \frac{p(t^2-1)}{2}.$$

Using Lemma 5.1 with $\beta = \frac{1}{\sqrt{1-\theta}}$, Lemma 4.4 and $\Psi(V) \leq \tau$, we obtain

$$\begin{aligned}
\Psi(V_+) &\leq n\psi\left(\frac{1}{\sqrt{1-\theta}}\varrho\left(\frac{\Psi(V)}{n}\right)\right) \\
&\leq \frac{np}{2}\left(\left[\frac{1}{\sqrt{1-\theta}}\varrho\left(\frac{\Psi(V)}{n}\right)\right]^2 - 1\right) \\
&= \frac{np}{2(1-\theta)}\left(\varrho\left(\frac{\Psi(V)}{n}\right)^2 - (1-\theta)\right) \\
&\leq \frac{np}{2(1-\theta)}\left(\left[1+\sqrt{\frac{2\Psi(V)}{np}}\right]^2 - (1-\theta)\right) \\
&\leq \frac{np}{2(1-\theta)}\left(\theta + 2\frac{\tau}{np} + 2\sqrt{\frac{2\tau}{np}}\right) \\
&= \frac{np\theta + 2\tau + 2\sqrt{2\tau np}}{2(1-\theta)},
\end{aligned}$$

This completes the proof. □

Denote

$$\Psi_0 = \frac{np\theta + 2\tau + 2\sqrt{2\tau np}}{2(1-\theta)}. \tag{5.1}$$

Then Ψ_0 is an upper bound of $\Psi(V_+)$.
After a damped step we have

$$X_+ = X + \alpha\Delta X, \quad y_+ = y + \alpha\Delta y, \quad S_+ = S + \alpha\Delta S, \quad \alpha > 0.$$

Using (2.6), we have:

$$X_+ = \sqrt{\mu}D(V + \alpha D_X)D, \qquad S_+ = \sqrt{\mu}D^{-1}(V + \alpha D_S)D^{-1}. \tag{5.2}$$

From (2.5), we have

$$V_+ = \frac{1}{\sqrt{\mu}}(D^{-1}X_+S_+D)^{1/2}.$$

It is easily to see that the matrix V_+^2 is unitarily similar to the matrix $X_+^{\frac{1}{2}}S_+X_+^{\frac{1}{2}}$ and therefore to the matrix $\overline{V}_+^2 = (V + \alpha D_X)^{\frac{1}{2}}(V + \alpha D_S)(V + \alpha D_X)^{\frac{1}{2}}$. This implies that both of the matrices have the same eigenvalues. Then, we have

$$\Psi(V_+) = \Psi\left(\overline{V}_+\right). \tag{5.3}$$

Lemma 5.3 (Proposition 5.2.6 in [18])**.** *Suppose that matrices V_1 and V_2 are symmetric positive definite and Ψ is the real valued matrix function induced by the matrix function ψ. Then we have*

$$\Psi\left(\left[V_1^{\frac{1}{2}}V_2V_1^{\frac{1}{2}}\right]^{\frac{1}{2}}\right) \leq \frac{1}{2}(\Psi(V_1) + \Psi(V_2)).$$

Based on Lemma 5.3, we obtain

$$\Psi(V_+) \leq \frac{1}{2}(\Psi(V + \alpha D_X) + \Psi(V + \alpha D_S)). \tag{5.4}$$

which implies from (5.3) that

$$\Psi(\overline{V}_+) \leq \frac{1}{2}(\Psi(V+\alpha D_X)+\Psi(V+\alpha D_S)).$$

Define for $\alpha > 0$, the following univariante functions

$$\begin{aligned} f(\alpha) &= \Psi(V_+)-\Psi(V)=\Psi(\overline{V}_+)-\Psi(V), \\ f_1(\alpha) &= \frac{1}{2}(\Psi(V+\alpha D_X)+\Psi(V+\alpha D_S))-\Psi(V). \end{aligned}$$

From (5.4), $f(\alpha) \leq f_1(\alpha)$ and $f(0)=f_1(0)=0$.

Now, to estimate the decrease of the proximity during one step, we need the two successive derivatives of $f_1(\alpha)$ with respect to α. We have

$$\begin{aligned} f_1'(\alpha) &= \frac{1}{2}Tr(\psi'(V+\alpha D_X)D_X+\psi'(V+\alpha D_S)D_S), \\ f_1^{''}(\alpha) &= \frac{1}{2}Tr(\psi''(V+\alpha D_X)D_X^2+\psi''(V+\alpha D_S)D_S^2). \end{aligned}$$

It is obvious that $f_1^{''}(\alpha) > 0$, unless $D_X = D_S = 0$.

From the third equation of the system (3.3) and (4.5), we have

$$f_1'(0)=\frac{1}{2}Tr(\psi'(V)(D_X+D_S))=\frac{1}{2}Tr(-(\psi'(V))^2)=-2\delta^2(V).$$

For notational convenience, let $\delta=\delta(V)$ and $\Psi=\Psi(V)$.

In what follows, we are going to introduce conditions on the step size α in which the function $f(\alpha)$ is a decreasing function. We need the following lemmas.

Lemma 5.4 (Lemma 5.19 in [22])**.** *Let δ be defined as in* (4.5)*. Then we have*

$$f_1''(\alpha) \leq 2\delta^2\psi''(\lambda_n(V)-2\alpha\delta).$$

Lemma 5.5 (Lemma 4.2 in [2])**.** *If the step size α satisfies*

$$-\psi'(\lambda_n(V)-2\alpha\delta)+\psi'(\lambda_n(V)) \leq 2\delta,$$

then

$$f_1'(\alpha) \leq 0.$$

Lemma 5.6 (Lemma 4.4 in [2])**.** *Let ρ and $\overline{\alpha}$ be defined as in Lemma* 4.5 *and Lemma* 5.5 *,respectively. Then*

$$\overline{\alpha} \geq \frac{1}{\psi''(\rho(2\delta))}.$$

Lemma 5.7. *Let ρ and $\overline{\alpha}$ be defined as in Lemma* 5.6*. If $1 \leq \tau \leq \Psi(V)$, then we have*

$$\overline{\alpha} \geq \frac{1}{p+\left(\frac{pq(p+1)}{q+1}\right)\left(\frac{4(q+1)\delta}{p}+1\right)^{\frac{pq+1}{pq}}}.$$

Proof. Using Lemma 5.6, Lemma 4.5 and (4.2) we have

$$\begin{aligned}
\overline{\alpha} &\geq \frac{1}{\psi''(\rho(2\delta))} \\
&\geq \frac{1}{\psi''((\frac{p}{4(q+1)\delta+p})^{\frac{1}{pq}})} \\
&= \frac{1}{p + \frac{p(pq)}{(q+1)}(\frac{p}{4(q+1)\delta+p})^{\frac{-pq-1}{pq}} + \frac{pq}{q+1}(\frac{p}{4(q+1)\delta+p})^{\frac{-2}{pq}}} \\
&\geq \frac{1}{p + \left(\frac{pq(p+1)}{q+1}\right)\left(\frac{4(q+1)\delta}{p} + 1\right)^{\frac{pq+1}{pq}}}.
\end{aligned}$$

This completes the proof. □

Define the default step size $\widetilde{\alpha}$ as follows:

$$\widetilde{\alpha} = \frac{1}{p + \left(\frac{pq(p+1)}{q+1}\right)\left(\frac{4(q+1)\delta}{p} + 1\right)^{\frac{pq+1}{pq}}}, \tag{5.5}$$

with $\widetilde{\alpha} \leq \overline{\alpha}$.

The next lemma provides the amount of decrease in the proximity function after an inner iteration.

Lemma 5.8 (Lemma 4.5 in [2])**.** *If the step size α is chosen such that $\alpha \leq \overline{\alpha}$, then*

$$f(\alpha) \leq -\alpha\delta^2.$$

Lemma 5.9. *Let $\widetilde{\alpha}$ be defined as in* (5.5). *Then we have*

$$f(\widetilde{\alpha}) \leq -\frac{1}{26\sqrt{2}pq(p+1)(q+1)^{\frac{1}{pq}}}\Psi^{\frac{pq-1}{2pq}}.$$

Proof. Using Lemma 5.8 with $\alpha = \widetilde{\alpha}$ and (5.5), we have

$$\begin{aligned}
f(\widetilde{\alpha}) &\leq -\widetilde{\alpha}\delta^2 \\
&= -\frac{\delta^2}{p + \left(\frac{pq(p+1)}{q+1}\right)\left(\frac{4(q+1)\delta}{p} + 1\right)^{\frac{pq+1}{pq}}} \\
&\leq -\frac{\delta^2}{p(2\delta)^{\frac{pq+1}{pq}} + pq(p+1)(q+1)^{\frac{1}{pq}}(\frac{4\delta}{p} + \frac{2\delta}{q+1})^{\frac{pq+1}{pq}}} \\
&= -\frac{\delta^2}{4p + 25pq(p+1)(q+1)^{\frac{1}{pq}}\delta^{\frac{pq+1}{pq}}} \\
&\leq -\frac{\delta^{\frac{pq-1}{pq}}}{26pq(p+1)(q+1)^{\frac{1}{pq}}} \\
&\leq -\frac{\Psi^{\frac{pq-1}{2pq}}}{26\sqrt{2}pq(p+1)(q+1)^{\frac{1}{pq}}}.
\end{aligned}$$

This completes the proof. □

$\boxed{5.1}$ Complexity of the algorithm

$\boxed{5.1.1}$ Inner iteration bound

Lemma 5.10. *(Proposition 1.3.2 in [18]) Suppose that a sequence $\{t_k > 0,\ k = 0, 1, 2, ..., K\}$ satisfy the following inequality:*

$$t_{k+1} \leq t_k - \eta t_k^{\gamma}, \quad \eta > 0, \quad \gamma \in [0, 1[, \quad k = 0, 1, 2, ..., K.$$

Then

$$K \leq \left[\frac{t_0^{1-\gamma}}{\eta(1-\gamma)} \right].$$

After the update of μ to $(1-\theta)\mu$, we have

$$\Psi(V_+) \leq \Psi_0 = \frac{np\theta + 2\tau + 2\sqrt{2\tau np}}{2(1-\theta)}.$$

We need to count how many inner iterations are required to return to the situation where $\Psi(V) \leq \tau$. We denote the value of Ψ after $\mu-$update as Ψ_0, the subsequent values in the same outer iteration are denoted as Ψ_k, $k = 1, 2, ..., K$, where K denotes the total number of inner iterations per an outer iteration. The decrease in each inner iteration is given by Lemma 5.9.

Theorem 5.11. *Let K be the total number of inner iterations per an outer iteration and Ψ_0 be defined as in* (5.1). *Then for $q \geq 1$, we have*

$$K \leq \left[26\sqrt{2}pq(p+1)(q+1)^{\frac{1}{pq}} \Psi_0^{\frac{pq+1}{2pq}} \right].$$

Proof. Combining Lemma 5.9 and Lemma 5.10 with$\eta = \frac{1}{26\sqrt{2}pq(p+1)(q+1)^{\frac{1}{pq}}}$ and $\gamma = \frac{pq-1}{2pq}$, we have

$$K \leq \left[26\sqrt{2}pq(p+1)(q+1)^{\frac{1}{pq}} \Psi_0^{\frac{pq+1}{2pq}} \right].$$

This completes the proof. □

$\boxed{5.1.2}$ Total iteration bound

The number of outer iterations is bounded above by $\left[\frac{1}{\theta} \log \frac{n}{\varepsilon}\right]$ (see [19]). By multiplying the number of outer iterations by the number of inner iterations, we get an upper bound for the total number of iterations, namely,

$$26\sqrt{2}pq(p+1)(q+1)^{\frac{1}{pq}} \Psi_0^{\frac{pq+1}{2pq}} \frac{1}{\theta} \log \frac{n}{\varepsilon} \tag{5.6}$$

1. For large update methods, with $\tau = O(n)$ and $\theta = \Theta(1)$, we get $\Psi_0 = O(np)$, and by choosing $p = 1$, $q = \log n$, we obtain

$$O(\sqrt{n} \log n \log \frac{n}{\varepsilon}) \text{ iterations complexity.}$$

2. For small update methods, with $\tau = O(1)$ and $\theta = \Theta(\frac{1}{\sqrt{n}})$, substitution of these values into (5.6) does not give the best possible bound. A better bound is obtained as follows

Since $\frac{1}{\sqrt{1-\theta}} \geq 1$ and $\varrho\left(\frac{\Psi(V)}{n}\right) \geq 1$, we have $\frac{\varrho\left(\frac{\Psi(V)}{n}\right)}{\sqrt{1-\theta}} \geq 1$. Using Lemma 5.1 with $\beta = \frac{1}{\sqrt{1-\theta}}$, Lemma 4.3 (ii), Lemma 4.4 and $\Psi(V) \leq \tau$, we have

$$\begin{aligned}
\Psi(V_+) &\leq n\psi\left(\frac{1}{\sqrt{1-\theta}}\varrho\left(\frac{\Psi(V)}{n}\right)\right) \\
&\leq \frac{np}{2}\left(\frac{2pq+q+1}{q+1}\right)\left(\frac{\varrho\left(\frac{\Psi(V)}{n}\right)}{\sqrt{1-\theta}} - 1\right)^2 \\
&\leq \frac{np(2pq+q+1)}{2(q+1)}\left(\frac{1+\sqrt{\frac{2\tau}{np}} - \sqrt{1-\theta}}{\sqrt{1-\theta}}\right)^2 \\
&\leq \frac{np(2pq+q+1)}{2(q+1)}\left(\frac{\theta+\sqrt{\frac{2\tau}{np}}}{\sqrt{1-\theta}}\right)^2 \\
&= \frac{p(2pq+q+1)}{2(q+1)(1-\theta)}\left(\sqrt{n}\theta + \sqrt{\frac{2\tau}{p}}\right)^2 = \Psi_0,
\end{aligned}$$

where the last inequality holds from $1-\sqrt{1-\theta} = \frac{\theta}{1+\sqrt{1-\theta}} < \theta, \;\; 0 \leq \theta < 1$. Using this upper bound for Ψ_0, we get $\Psi_0 = O(\frac{p(2pq+q+1)}{(q+1)})$ and by choosing $p = 1$, $q =$ any constant we obtain

$$O(\sqrt{n}\log\frac{n}{\varepsilon}) \text{ iterations complexity.}$$

These are the best known complexity results for such methods.

6 Numerical Results

In this section, our main focus is to provide a numerical experiences regarding the practical performance of the new proposed kernel function in comparison with:

1. The kernel function which has been proposed in [13]

$$\psi_f(t) = \frac{t^2-1}{2} - \int_1^t \left(\frac{e-1}{e^x-1}\right)^p dx, \qquad p \geq 1;$$

and showed that it is well promising in practice in comparison with other eight famous kernel functions including two logarithmic kernel functions, namely,

$$\begin{aligned}
\psi(t) &= \frac{t^2-1}{2} - \log(t), \\
\psi(t) &= \frac{t^2-1}{2} - \log(t) + \frac{1}{8}\tan^2(h(t)), \qquad h(t) = \frac{1-t}{2+4t}\pi.
\end{aligned}$$

2. the generalized logarithmic kernel function proposed in [11]

$$\psi_g(t) = \frac{1}{1+p}(t^{1+p}-1) - \log(t), \quad p \in [0,1]$$

In all tables we denote by "iter" and "CPU" the total number of iterations and the time required in second, respectively. Furthermore, ψ_{new} stands for our new proposed kernel function. We used the following parameters:

$$\begin{aligned} \varepsilon &= 10^{-8}, \qquad \tau = 1, \qquad \mu_0 = 1, \\ \theta &\in \{0.1, 0.5, 0.9\}, \quad p = 1, \quad q = \log n. \end{aligned}$$

Example 6.1. [22]: Consider the following (SDO) problem:

$$\begin{aligned} A_1 &= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -2 & -1 \\ 0 & -1 & 1 & -1 & -2 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & 0 & -2 & 2 & 0 \\ 0 & 2 & 1 & 0 & 2 \\ -2 & 1 & -2 & 0 & 1 \\ 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 2 \end{bmatrix}, \\ A_3 &= \begin{bmatrix} 2 & 2 & -1 & -1 & 1 \\ 2 & 0 & 2 & 1 & 1 \\ -1 & 2 & 0 & 1 & 0 \\ -1 & 1 & 1 & -2 & 0 \\ 1 & 1 & 0 & 0 & -2 \end{bmatrix}, \quad C = \begin{bmatrix} 3 & 3 & -3 & 1 & 1 \\ 3 & 5 & 3 & 1 & 2 \\ -3 & 3 & -1 & 1 & 2 \\ 1 & 1 & 1 & -3 & -1 \\ 1 & 2 & 2 & -1 & -1 \end{bmatrix}, \\ b &= \begin{bmatrix} -2 \\ 2 \\ -2 \end{bmatrix}, \quad X_0 = I, \quad S_0 = I, \quad y_0 = (1, 1, 1)^T. \end{aligned}$$

The following table summarize the obtained results

	ψ_f		ψ_g		ψ_{new}	
θ	*iter*	*CPU*	*iter*	*CPU*	*iter*	*CPU*
0.1	207	22.1097	310	8.3845	200	0.4766
0.5	45	4.8909	60	2.9874	43	0.1545
0.9	26	2.9296	33	2.0674	23	1.0956

Example 6.2. (Example 1 in [20]): Consider the following (SDO) problem:

$$\begin{aligned} A_1 &= \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \quad A_2 = I, \quad C_{ij} = -1, \qquad \forall i, j \in \{1, 2\} \\ b &= \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad X_0 = diag(0.5, 0.5), \quad S_0 = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}, \quad y_0 = \begin{bmatrix} 0 \\ -3 \end{bmatrix}. \end{aligned}$$

The following table summarize the obtained results

	ψ_f		ψ_g		ψ_{new}	
θ	*iter*	*CPU*	*iter*	*CPU*	*iter*	*CPU*
0.1	199	14.3190	308	10.6425	197	0.2147
0.5	43	3.4335	58	5.9818	40	0.1623
0.9	25	1.8708	36	2.0877	24	0.1059

Example 6.3. (Example 2 in [20]): In this example, we have

$$\begin{aligned} C &= diag(5,8,8,5), \quad A_4 = I, \quad b = (1,1,1,2)^T \text{ and for } k = 1,2,3: \\ A_k(i,j) &= \begin{cases} 1 & \text{if } \ i = j = k \text{ or } i = j = k+1; \\ -1 & \text{if } i = k, \ j = k+1 \text{ or } i = k+1, \ j = k; \\ 0 & \text{otherwise.} \end{cases} \\ X_0 &= \frac{1}{2}I, \quad S_0 = \begin{bmatrix} 2 & 1.5 & 0 & 0 \\ 1.5 & 3.5 & 1.5 & 0 \\ 0 & 1.5 & 3.5 & 1.5 \\ 0 & 0 & 1.5 & 2 \end{bmatrix}, \quad y_0 = (1.5, 1.5, 1.5, 1.5)^T. \end{aligned}$$

The following table summarize the obtained results

	ψ_f		ψ_g		ψ_{new}	
θ	*iter*	*CPU*	*iter*	*CPU*	*iter*	*CPU*
0.1	203	18.8729	290	28.7452	203	0.3841
0.5	44	4.1067	60	12.8451	44	0.1323
0.9	25	2.3846	33	1.6855	21	0.1120

Now, we pass to two examples that have variant size.

Example 6.4. (Example 4 in [20]): This (SDO) problem is defined as follow:

$$\begin{aligned} C &= -I, \ b(i) = 2, \ i = 1, ..., m. \\ A_k(i,j) &= \begin{cases} 1 & \text{if } i = j = k; \\ 1 & \text{if } i = j \text{ and } i = k+m; \quad k = 1, ..., m, \\ 0 & \text{otherwise.} \end{cases} \\ X_0 &= \begin{cases} 1.5 & i \leq j; \\ 0.5 & i > j; \end{cases}, \quad S_0 = I, \quad y_0(i) = -2, \quad i = 1, ..., m. \end{aligned}$$

The obtained results for $m \in \{10, 25, 50, 100\}$, are shown in the following tables:

for $m =$ 10

	ψ_f		ψ_g		ψ_{new}	
θ	*iter*	*CPU*	*iter*	*CPU*	*iter*	*CPU*
0.1	244	66.6537	389	30.8754	242	1.4856
0.5	61	16.8074	70	16.9895	61	0.6697
0.9	33	9.0325	45	10.0258	33	0.5430

for $m =$ 25

	ψ_f		ψ_g		ψ_{new}	
θ	*iter*	*CPU*	*iter*	*CPU*	*iter*	*CPU*
0.1	264	169.4989	356	120.8988	259	11.8889
0.5	64	38.3909	70	40.9858	61	4.8082
0.9	35	25.2266	47	29.6545	30	4.5558

for $m =$ 50

	ψ_f		ψ_g		ψ_{new}	
θ	$iter$	CPU	$iter$	CPU	$iter$	CPU
0.1	275	481.4549	364	458.6987	273	152.8225
0.5	65	174.0386	78	180.6959	62	79.8312
0.9	37	127.1565	54	130.9125	35	70.5128

for $m =$ 100

	ψ_f		ψ_g		ψ_{new}	
θ	$iter$	CPU	$iter$	CPU	$iter$	CPU
0.1	301	$6.7508e^{+3}$	330	$7.8561e^{+3}$	294	$4.7225e^{+3}$
0.5	67	$2.8665e^{+3}$	75	$3.2551e^{+3}$	61	$2.4653e^{+3}$
0.9	44	$2.9516e^{+3}$	60	$2.9525e^{+3}$	42	$2.4652e^{+3}$

Example 6.5. (Example cube in [6]): Consider the following (SDO) problem:

$$\begin{aligned} A_k(i,j) &= \begin{cases} 1 & \text{if } i=j=k \text{ or } i=j=k+m; \\ 25 & \text{if } i=j=k+1 \text{ or } i=j=k+m+1; \\ -5 & \text{if } i=k,\ j=k+1 \text{ or } i=k+m,\ j=k+m+1; \\ -5 & \text{if } i=k+1,\ j=k \text{ or } i=k+m+1,\ j=k+m; \\ 0 & \text{otherwise.} \end{cases} \quad ,k=1,...,m. \\ C &= -2I,\ b(i)=2,\ i=1,...,m, \quad X_0=S_0=I, \quad y_0=(0,0,...,0)^T. \end{aligned}$$

The algorithm is executed with $m \in \{100, 170, 250\}$ and $\theta = 0.9$, the following table summarize the obtained results

	ψ_f		ψ_g		ψ_{new}	
m	$iter$	CPU	$iter$	CPU	$iter$	CPU
100	44	$2.3945e^{+3}$	56	3.0548	43	$1.9658e^{+3}$
170	45	$3.7642e^{+4}$	53	3.8952	40	$2.6814e^{+4}$
250	50	$2.1550e^{+5}$	52	2.3688	51	$1.9942e^{+5}$

Example 6.6. (Test problems from SDPLIB [4]): The algorithm has been tested on reference problems from the SDPLIB test problem library. It is started with initial point $X_0 = S_0 = I$ and $y_0 = 0$ and with $\theta = 0.9$.

The following table summarize the obtained results

			ψ_f		ψ_g		ψ_{new}	
	m	n	$iter$	CPU	$iter$	CPU	$iter$	CPU
$control1$	21	15	27	6.1521	38	8.9645	25	0.5439
$control2$	66	30	27	17.3590	38	20.8744	25	7.6585
$control3$	136	45	27	84.3097	39	90.8421	25	74.1237
$control4$	231	60	27	423.7427	37	587.681	24	356.1534
$control5$	351	75	27	$1.4224e^{+3}$	38	$1.9984e^{+3}$	26	$1.2821e^{+3}$
$gpp100$	101	100	28	232.8007	39	333.8521	26	204.1248
$hinf1$	13	16	29	11.5243	39	18.9254	27	0.2655
$hinf2$	13	16	30	11.7002	40	17.5741	30	0.2843
$hinf3$	13	16	32	8.8578	44	10.9854	32	0.2993
$hinf4$	13	16	30	8.4722	39	12.8512	28	0.2828
$hinf5$	13	16	32	9.8052	40	12.6845	30	0.3148
$hinf6$	13	16	31	9.2530	45	13.0257	31	0.3049
$hinf7$	13	16	30	8.0903	47	11.0285	29	0.3229
$hinf8$	13	16	33	9.5336	48	12.8451	30	0.2958
$hinf9$	13	16	32	8.7731	40	10.9841	31	0.3122
$hinf10$	21	18	30	8.6530	45	10.4557	28	0.5635
$hinf11$	31	22	30	12.0619	45	17.6521	29	1.4668
$hinf12$	43	24	30	12.6508	43	15.9812	27	2.6228
$hinf13$	57	30	35	22.0333	50	30.8512	30	8.0645
$hinf14$	73	34	30	26.9598	41	36.0581	30	14.3958
$hinf15$	91	37	32	44.1425	47	57.8954	32	28.4352
$mcp100$	100	100	34	322.1379	46	368.2854	30	246.0535
$qap5$	136	26	27	34.2571	40	56.6524	25	24.6527
$theta1$	104	50	32	79.2318	41	90.8544	30	54.4034
$truss1$	6	13	28	6.0656	36	7.9541	28	0.0695
$truss2$	58	133	28	171.3178	35	203.8545	28	125.5758
$truss3$	27	31	30	12.2743	40	15.9845	27	1.5593
$truss4$	12	19	28	7.3415	39	9.6985	26	0.1798

From the obtained results, the following remarks are concluded:

- Our new kernel function produces better execution time than the others kernel functions.
- In most of cases, our new kernel function reduces the iteration number than the others kernel functions.
- The iteration number of the algorithm depend on the value of parameter θ, in most cases the larger θ gives better iteration number.
- The results show a very slow growth as n increasing which is precisely what is hoped for IPMs.

7 Conclusion

In this paper, we proposed a new efficient parametrized logarithmic kernel function. We have shown that the best result of iteration bounds for large and small update methods can be achieved, namely $O(\sqrt{n}\log n \log \frac{n}{\varepsilon})$ for large update and $O(\sqrt{n}\log \frac{n}{\varepsilon})$ for small update methods. The results obtained in this paper represent an important contributions to improve the convergence and the complexity analysis of primal-dual IPMs for SDO, and in our knowledge, these results are one of the best known complexity bound for large update with a logarithmic barrier term for SDO. Moreover, the numerical results were presented to illustrate the advantage of our kernel function.

Acknowledgments

The authors are very grateful and would like to thank the anonymous referees and the editor for their suggestions and helpful comments, which significantly improved the presentation of this paper.

References

[1] Y.Q. Bai, M. El Ghami and C. Roos, A new efficient large-update primal-dual interior-point method based on a finite barrier, *SIAM Journal on Optimization* 13 (2003) 766–782.

[2] Y.Q. Bai, M. El Ghami and C. Roos, A comparative study of kernel functions for primal-dual interior-point algorithms in linear optimization, *SIAM Journal on Optimization* 15 (2004) 101–128.

[3] Y.Q. Bai, G. Lesaja, C. Roos, G.Q. Wang and M. El Ghami, A class of large-update and small-update primal-dual interior-point algorithms for linear optimization, *Journal of Optimization Theory and Applications* 138 (2008) 341–359.

[4] B. Borchers, SDPLIB 1.2, A library of semidefinite programming test problems, *Optim. Methods Softw.* 11 (1999) 683–690.

[5] M. Bouafia, D. Benterki and A. Yassine, An efficient primal-dual interior point method for linear programming problems based on a new kernel function with a trigonometric barrier term, *Journal of Optimization Theory and Applications* 170 (2016) 528–545.

[6] J.P. Crouzeix and B. Merikhi, A logarithm barrier method for semidefinite programming, *RAIRO-Oper Res.* 42 (2008) 123–139

[7] E. De Klerk, *Aspects of Semidefinite Programming: Interior Point Algorithms and Selected Applications*, Kluwer Academic Publishers, 2002.

[8] M. El Ghami, Primal-dual algorithms for semidefinite optimization problems based on generalized trigonometric barrier function, *International Journal of Pure and Applied Mathematics* 114 (2017) 797–818.

[9] M. El Ghami, Y.Q. Bai and C. Roos, Kernel-function based algorithms for semidefinite optimization, *RAIRO Operations Research* 43 (2009) 189–199.

[10] M. El Ghami, Z. A. Guennoun, S. Bouali and T. Steihaug, Interior-point methods for linear optimization based on a kernel function with a trigonometric barrier term, *Journal of Computational and Applied Mathematics* 236 (2012) 3613–3623.

[11] M. El Ghami, I. D. Ivanov, C. Roos and T. Steihaug, A polynomial-time algorithm for LO based on generalized logarithmic barrier functions, *Int. J. Appl. Math.* 21 (2008), 99–115.

[12] M. El Ghami, C. Roos and T. Steihaug, A generic primal-dual interior-point method for semidefinite optimization based on a new class of kernel functions, *Optimization Methods and Software* 25 (2010) 387–403.

[13] S. Fathi-hafshejani and A. Fakharzadeh , An interior-point algorithm for semidefinite optimization based on a new parametric kernel function, *Journal of Nonlinear Functional Analysis* (2018).

[14] N.K. Karmarkar, A new polynomial-time algorithm for linear programming, *Combinatorica* 4 (1984) 373–395.

[15] Y.H. Lee, Y.Y. Cho, J.H. Jin and G.M. Cho, Interior-point algorithms for LO and SDO based on a new class of kernel functions, *Journal of Nonlinear and Convex Analysis* 13 (2012) 555–573.

[16] Y.E. Nesterov and A.S. Nemirovski, *Interior Point Polynomial Algorithms in Convex Programming*, SIAM Studies in Applied Mathematics, Volume 13, SIAM, Philadelphia, PA,1994.

[17] Y. Nesterov and M.J. Todd, Self-scaled barriers and interior-point methods for convex programming, *Mathematics of Operations Research* 22 (1997) 1–42.

[18] J. Peng, C. Roos and T. Terlaky, *Self-Regularity, A new Paradigm for Primal-Dual Interior-Point Algorithms*, Princeton University Press, 2002.

[19] C. Roos, T. Terlaky and J.P. Vial, *Theory and Algorithms for Linear Optimization, An Interior Approach*, John Wiley and Sons, Chichester, U.K., 1997.

[20] I. Touil, D. Benterki and A. Yassine, A feasible primal-dual interior-point method for linear semidefinite programming, *J.Comput. Appl. Math.* 312 (2017), 216–230.

[21] G.Q. Wang and Y.Q. Bai, A class of polynomial primal-dual interior-point algorithm for semidefinite optimization, *Journal of Shanghai University* 10 (2006) 198–207.

[22] G.Q. Wang, Y.Q. Bai and C. Roos, Primal-dual interior-point algorithms for semidefinite optimization based on a simple kernel function, *Journal of Mathematical Modelling and Algorithms* 4 (2005) 409–433.

[23] H. Wolkowicz, R. Saigal and L. Vandenberghe, *Handbook of Semidefinite Programming, Theory, Algorithms and Applications*, Kluwer Academic Publishers, 2000.

Manuscript received 31 October 2021
revised 9 February 2022
accepted for publication 20 February 2022

Zerari Amina
LMFN, Faculty of Sciences Setif-1 Ferhat Abbas University
19000, Algeria
E-mail address: amina.zerari@univ-setif.dz

Benterki Djamel
LMFN, Faculty of Sciences Setif-1 Ferhat Abbas University
19000, Algeria
E-mail address: djbenterki@univ-setif.dz

Yassine Adnan
Normandie University, UNIHAVRE, LMAH FR-CNRS-3335
ISCN, 76600 Le Havre, France
E-mail address: adnan.yassine@univ-lehavre.fr