Approximation of rank function and its application to the nearest low-rank correlation matrix [1]

Shujun Bi[†] and Shaohua Pan[‡]

July 8, 2011

Abstract. The rank function rank$(\cdot)$ is neither continuous nor convex which brings much difficulty to the solution of rank minimization problems. In this paper, we provide a unified framework to construct the approximation functions of rank$(\cdot)$, and study their favorable properties. Particularly, with two families of approximation functions, we propose a convex relaxation method for the rank minimization problems with positive semidefinite cone constraints, and illustrate its application by computing the nearest low-rank correlation matrix. Numerical comparisons with the convex relaxation method in [17] indicate that our method tends to yield a better local optimal solution.

Key Words: rank optimization problem; approximation; convex relaxation; nearest low-rank correlation matrix; semismooth Newton method

1 Introduction

Given a closed convex subset Ω of $\mathbb{R}^{m\times n}$, we consider the rank minimization problems

$$\begin{array}{ll} \min & f(X) \\ \text{s.t.} & \mathrm{rank}(X) \le r, \\ & X \in \Omega \end{array} \tag{1}$$

and

$$\begin{array}{ll} \min & f(X) + \nu\,\mathrm{rank}(X) \\ \text{s.t.} & X \in \Omega, \end{array} \tag{2}$$

[1] This work was supported by National Young Natural Science Foundation (No.10901058) and Guangdong Natural Science Foundation (No.9251802902000001) and the Fundamental Research Funds for the Central Universities (SCUT).

[†] Department of Mathematics, South China University of Technology, Tianhe District of Guangzhou City, China (beamilan@163.com).

[‡] Department of Mathematics, South China University of Technology, Tianhe District of Guangzhou City, China (shhpan@scut.edu.cn).

where $f\colon \mathbb{R}^{m\times n} \to \mathbb{R}$ is a continuously differentiable convex function, r is an integer with $1 \le r \le m$, and $\nu > 0$ is a fixed constant. The rank minimization problems in the form of (1) and (2) have numerous applications and arise in a diverse area such as system control [12, 20], matrix completion [6, 36], signal and image processing, statistics, computer vision, Euclidean embedding, and collaborative filtering. For more details, please refer to the recent survey paper by Recht, Fazel and Parrilo [26]. In addition, several well-known combinatorial optimization problems such as maximal cut and maximal stable set can be formulated as problem (1) (see, for example, [13, 4]).

Note that (1) and (2) are discontinuous nonconvex optimization problems due to the discontinuity and nonconvexity of rank function $\mathrm{rank}(\cdot)$. Hence, it is difficult even impossible to seek their global optimal solutions. A common and effective strategy is to replace the original problems (1) and (2) by a sequence of convex optimization problems that are easy to solve, so that a favorable feasible or local optimal solution can be found. In [10, 26], Fazel et al. showed that the convex envelope of $\mathrm{rank}(\cdot)$ on the unit ball in the operator norm is the nuclear norm. Motivated by this result, the convex relaxation methods based on the nuclear norm minimization are extensively studied, and many effective algorithms have been proposed for them (see, e.g., [18, 34, 19, 7, 22]). However, the nuclear norm has a drastic derivation from the rank function since the former is convex whereas the latter is concave. This means that the nuclear norm is not a high-quality approximation to $\mathrm{rank}(\cdot)$. In fact, as pointed out in [26, 27], the nuclear norm relaxation exhibits a phase transition where for sufficiently small rank the method always succeeds, but in the complement of the region it may fail even never succeed.

In this paper, noting that $\mathrm{rank}(\cdot)$ is a composition of the one-side sign function and the singular value function, we give a unified framework to construct its approximation functions, and prove that $\mathrm{rank}(\cdot)$ can be approximated to any prescribed accuracy when the approximation parameter is smaller than a given threshold. Although these approximation functions are not differentiable, they are semismooth and directional differentiable everywhere in $\mathbb{R}^{m\times n}$. Particularly, with two families of approximation functions, we proposed a convex relaxation method for the rank minimization problems with the positive semidefinite (PSD) cone constraint $X \succeq 0$, and apply it for computing the nearest low-rank correlation matrices. Numerical comparisons with that of [17] for some common test problems show that our relaxation method may yield a better solution.

Throughout this paper, $\mathbb{R}_+$ represents the set of nonnegative real numbers, $\mathbb{R}^{m\times n}$ denotes the linear space of $m \times n$ real matrices equipped with the trace inner product $\langle\cdot,\cdot\rangle$, i.e., $\langle X, Y\rangle = \mathbf{Tr}(X^TY)$ for any $X, Y \in \mathbb{R}^{m\times n}$, and $\mathbb{S}^n$ denotes the linear space of $n \times n$ real symmetric matrices with $\mathbb{S}^n_+$ denoting the PSD cone in $\mathbb{S}^n$. For $X \in \mathbb{R}^{m\times n}$, we denote by $\|X\|_*$ the nuclear norm of X, i.e., the sum of singular values of X; and for $X \in \mathbb{S}^n$, we write $X \succeq 0$ to mean $X \in \mathbb{S}^n_+$. For a vector $x \in \mathbb{R}^m$, $\mathrm{Diag}(x)$ denotes the diagonal matrix with the components of x as the diagonal entries; and for $X \in \mathbb{S}^n$, $\mathrm{diag}(X)$ denotes the vector consisting of the diagonal entries of X. Without loss of

generality, in the rest of this paper, we always assume $m \leq n$.

2 Approximation of rank function

For any given $X \in \mathbb{R}^{m\times n}$, suppose that X has the singular value decomposition (SVD)

$$X = U\,[\Sigma \quad 0]\,V^T,$$

where $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are orthogonal matrices, and Σ is an $m \times m$ diagonal matrix of the singular value vector of X, i.e., $\Sigma = \mathrm{Diag}(\sigma(X)) = \mathrm{Diag}(\sigma_1(X), \ldots, \sigma_m(X))$. Without loss of generality, we let $\sigma_1(X) \geq \sigma_2(X) \geq \cdots \geq \sigma_m(X) \geq 0$.

Let $\theta : \mathbb{R}_+ \to \{0, 1\}$ denote the nonnegative one-side sign function defined by

$$\theta(t) := \begin{cases} 1 & \text{if } t > 0, \\ 0 & \text{if } t = 0. \end{cases} \tag{3}$$

Then, it is immediate to see that the rank function can be reformulated as

$$\mathrm{rank}(X) = \sum_{i=1}^{m} \theta(\sigma_i(X)) \quad \forall X \in \mathbb{R}^{m\times n}.$$

This reformulation implies that the discontinuity of $\mathrm{rank}(\cdot)$ is caused by that of θ, which motivates us to seek a continuously differentiable $\phi : \mathbb{R}_{++} \times \mathbb{R}_+ \to \mathbb{R}_+$ with

$$\phi(\varepsilon, 0) = 0 \quad \forall \varepsilon > 0 \quad \text{and} \quad \lim_{\varepsilon \downarrow 0} \phi(\varepsilon, t) = \theta(t) \quad \forall t \geq 0, \tag{4}$$

so that the discontinuous $\mathrm{rank}(\cdot)$ can be approximated by $\sum_{i=1}^{m} \phi(\varepsilon, \sigma_i(\cdot))$.

Let Φ be the Löwner's nonsymmetric matrix function associated to ϕ, that is,

$$\Phi(\varepsilon, X) := U\,[\mathrm{Diag}(\phi(\varepsilon, \sigma_1(X)), \ldots, \phi(\varepsilon, \sigma_m(X))) \quad 0]\,V^T \quad \forall \varepsilon > 0, X \in \mathbb{R}^{m\times n}. \tag{5}$$

This function is well defined by [9, Section 2.2] since $\phi(0, 0) = \lim_{\varepsilon\downarrow 0} \phi(\varepsilon, 0) = 0$. In fact, for any given $\varepsilon > 0$, $\Phi(\varepsilon, \cdot)$ is continuously differentiable in $\mathbb{R}^{m\times n}$. Indeed, setting

$$\tilde{\phi}(\varepsilon, t) := \begin{cases} \phi(\varepsilon, t) & \text{if } t \geq 0; \\ -\phi(\varepsilon, -t) & \text{if } t < 0, \end{cases} \tag{6}$$

we have that $\tilde{\phi}$ is continuously differentiable in $\mathbb{R}$, and for any $X \in \mathbb{R}^{m\times n}$,

$$\Phi(\varepsilon, X) = U\left[\mathrm{Diag}(\tilde{\phi}(\varepsilon, \sigma_1(X)), \ldots, \tilde{\phi}(\varepsilon, \sigma_m(X))) \quad 0\right] V^T. \tag{7}$$

From [9, Section 2.2] it then follows that $\Phi(\varepsilon, \cdot)$ is continuously differentiable in $\mathbb{R}^{m\times n}$. We see that $\sum_{i=1}^{m} \phi(\varepsilon, \sigma_i(X))$ is exactly the nuclear norm of $\Phi(\varepsilon, X)$, that is,

$$\|\Phi(\varepsilon, X)\|_* = \sum_{i=1}^{m} \phi(\varepsilon, \sigma_i(X)) = \sum_{i=1}^{m} \tilde{\phi}(\varepsilon, \sigma_i(X)) \quad \forall X \in \mathbb{R}^{m\times n}. \tag{8}$$

This means that the nuclear norm of the smooth Φ provides an approximation to rank$(\cdot)$.

The discussions above show that, to obtain an approximation of rank$(\cdot)$, the key is to seek a smoothing function of θ. In what follows, we provide a unified framework to construct the smoothing functions ϕ satisfying (4). It is easy to verify that θ can be written as $\theta(t) = \int_{-\infty}^{t} \delta(u)du$, where $\delta(\cdot)$ is the Dirac delta function which satisfies

$$\delta(u) \geq 0 \quad \text{and} \quad \int_{-\infty}^{+\infty} \delta(u)du = 1.$$

The fact that the function θ can be obtained by integrating the Dirac delta function, prompts us to propose probability density functions as a means of smoothing the Dirac delta function and its integral. Like Chen and Mangasarian [5], we consider the piecewise continuous density function $\rho(\cdot)$ with finite number of pieces, that is, it satisfies

$$\rho(u) \geq 0 \quad \text{and} \quad \int_{-\infty}^{+\infty} \rho(u)du = 1, \tag{9}$$

and then parameterize the density function with a function of parameter $\varepsilon > 0$ as follows:

$$\hat{\delta}(\varepsilon, u) := \frac{1}{\vartheta(\varepsilon)}\rho\left(\frac{u}{\vartheta(\varepsilon)}\right) \qquad \forall \varepsilon > 0, u \in \mathbb{R} \tag{10}$$

where $\vartheta : \mathbb{R}_+ \to \mathbb{R}_+$ is a continuous increasing function with $\vartheta(0) = 0$ and $\vartheta(\varepsilon) > 0$ when $\varepsilon > 0$. When $\varepsilon \downarrow 0$, the limit of $\hat{\delta}(\varepsilon, u)$ is $\delta(u)$, which motivates us to define a class of approximations to θ as

$$\phi(\varepsilon, t) := \int_{-\infty}^{t} \hat{\delta}(\varepsilon, u)du \qquad \forall t \geq 0. \tag{11}$$

To guarantee $\phi(\varepsilon, 0) = 0$, the density function ρ is also required to satisfy the assumption

Assumption (A): when $u < 0$, $\rho(u) = 0$; and when $u \geq 0$, $\rho(u) \geq 0$.

Under Assumption (A), it is easy to get $\phi(\varepsilon, t) = 0$ when $t = 0$; and when $t > 0$,

$$\phi(\varepsilon, t) - 1 = \int_{-\infty}^{t} \hat{\delta}(\varepsilon, u)du - 1 = \int_{-\infty}^{t/\vartheta(\varepsilon)} \rho(y)dy - 1 = -\int_{t/\vartheta(\varepsilon)}^{+\infty} \rho(y)dy, \tag{12}$$

which means that $\phi(\varepsilon, t) \approx 1$ when ε is small enough. Hence, $\phi(\varepsilon, t)$ defined by (11) with a density function ρ satisfying Assumption (A) is an approximation of θ[†]. Furthermore, as will be shown in the following proposition, such approximation is smooth when ρ is continuous on $\mathbb{R}_+$, and has some other favorable properties.

[†]Set $\hat{\phi}(\varepsilon, t) = \begin{cases} \phi(\varepsilon, t) & \text{if } t \geq 0 \\ \phi(\varepsilon, -t) & \text{if } t < 0 \end{cases}$. Then, $\hat{\phi}(\varepsilon, t)$ is an approximation to $\hat{\theta}(t) := \begin{cases} 1 & \text{if } t > 0 \\ 0 & \text{if } t = 0 \\ 1 & \text{if } t < 0 \end{cases}$.
Consequently, for any $x \in \mathbb{R}^n$, $\sum_{i=1}^{n} \hat{\phi}(\varepsilon, x_i)$ is an approximation to the zero norm $\|x\|_0$ of x.

Proposition 2.1 *Let $\phi\colon \mathbb{R}_{++}\times\mathbb{R}_+\to\mathbb{R}_+$ be defined by (11) with a density function ρ satisfying Assumption (A). Then, for any given $\varepsilon>0$, the following statements hold.*

(a) *The function $\phi(\varepsilon,\cdot)$ is continuously differentiable in $\mathbb{R}_+$ if ρ is continuous on $\mathbb{R}_+$*[‡]*.*

(b) *For any given $t\in\mathbb{R}_+$, $\phi(\varepsilon,t)\geq\phi(\varepsilon',t)$ whenever $\varepsilon'\geq\varepsilon$.*

(c) *If ρ is continuous and nonincreasing on $[c,+\infty)$ for some $c\in\mathbb{R}_+$, then the function $\phi(\varepsilon,\cdot)$ is concave in $[c\vartheta(\varepsilon),+\infty)$.*

(d) *If ρ is bounded above on $\mathbb{R}_+$, then $\phi(\varepsilon,\cdot)$ is globally Lipschitz continuous on $\mathbb{R}_+$.*

Proof. Part (a) follows from equation (12). We next prove part (b). When $t=0$, the result is clear by Assumption (A). When $t>0$, for $\tau=\varepsilon$ or ε', we have that

$$\phi(\tau,t)=\int_{-\infty}^{t}\frac{1}{\vartheta(\tau)}\rho\left(\frac{u}{\vartheta(\tau)}\right)du=\int_{-\infty}^{t}\rho\left(\frac{u}{\vartheta(\tau)}\right)d\frac{u}{\vartheta(\tau)}=\int_{-\infty}^{t/\vartheta(\tau)}\rho(y)dy \tag{13}$$

which together with $0<\varepsilon\leq\varepsilon'$ implies $\phi(\varepsilon,t)\geq\phi(\varepsilon',t)$.

(c) We only need to show that, for any $t_1,t_2\in[c\vartheta(\varepsilon),+\infty)$ and $0<\alpha<1$,

$$\phi(\varepsilon,\alpha t_1+(1-\alpha)t_2)\geq\alpha\phi(\varepsilon,t_1)+(1-\alpha)\phi(\varepsilon,t_2),$$

which by (13) can be equivalently written as

$$\int_{-\infty}^{\frac{\alpha t_1+(1-\alpha)t_2}{\vartheta(\varepsilon)}}\rho(y)dy\geq\alpha\int_{-\infty}^{\frac{t_1}{\vartheta(\varepsilon)}}\rho(y)dy+(1-\alpha)\int_{-\infty}^{\frac{t_2}{\vartheta(\varepsilon)}}\rho(y)dy. \tag{14}$$

Without loss of generality, we may assume $t_1<t_2$. Then inequality (14) is equivalent to

$$\begin{aligned}
&\int_{-\infty}^{\frac{t_1}{\vartheta(\varepsilon)}}\rho(y)dy+\int_{\frac{t_1}{\vartheta(\varepsilon)}}^{\frac{\alpha t_1+(1-\alpha)t_2}{\vartheta(\varepsilon)}}\rho(y)dy\geq\int_{-\infty}^{\frac{t_1}{\vartheta(\varepsilon)}}\rho(y)dy+(1-\alpha)\int_{\frac{t_1}{\vartheta(\varepsilon)}}^{\frac{t_2}{\vartheta(\varepsilon)}}\rho(y)dy\\
\iff\ &\int_{\frac{t_1}{\vartheta(\varepsilon)}}^{\frac{\alpha t_1+(1-\alpha)t_2}{\vartheta(\varepsilon)}}\rho(y)dy\ \geq\ (1-\alpha)\int_{\frac{t_1}{\vartheta(\varepsilon)}}^{\frac{t_2}{\vartheta(\varepsilon)}}\rho(y)dy\\
\iff\ &\frac{1}{(1-\alpha)(t_2-t_1)}\int_{\frac{t_1}{\vartheta(\varepsilon)}}^{\frac{\alpha t_1+(1-\alpha)t_2}{\vartheta(\varepsilon)}}\rho(y)dy\geq\frac{1}{t_2-t_1}\int_{\frac{t_1}{\vartheta(\varepsilon)}}^{\frac{t_2}{\vartheta(\varepsilon)}}\rho(y)dy\\
\iff\ &\frac{1}{(1-\alpha)(t_2-t_1)}\int_{\frac{t_1}{\vartheta(\varepsilon)}}^{\frac{\alpha t_1+(1-\alpha)t_2}{\vartheta(\varepsilon)}}\rho(y)dy\geq\frac{1}{\alpha(t_2-t_1)}\int_{\frac{\alpha t_1+(1-\alpha)t_2}{\vartheta(\varepsilon)}}^{\frac{t_2}{\vartheta(\varepsilon)}}\rho(y)dy.
\end{aligned} \tag{15}$$

[‡]Among others, the continuity of ρ at 0 means the right continuity.

Thus, we only need to prove that (15) holds. Indeed, since $\rho(\cdot)$ is continuous and non-increasing on $[\frac{t_1}{\vartheta(\varepsilon)}, \frac{\alpha t_1+(1-\alpha)t_2}{\vartheta(\varepsilon)}]$ and $[\frac{\alpha t_1+(1-\alpha)t_2}{\vartheta(\varepsilon)}, \frac{t_2}{\vartheta(\varepsilon)}]$, by the integral mean value theorem there exist $\bar{t}_1 \in [\frac{t_1}{\vartheta(\varepsilon)}, \frac{\alpha t_1+(1-\alpha)t_2}{\vartheta(\varepsilon)}]$ and $\bar{t}_2 \in [\frac{\alpha t_1+(1-\alpha)t_2}{\vartheta(\varepsilon)}, \frac{t_2}{\vartheta(\varepsilon)}]$ such that

$$\rho(\bar{t}_1) = \frac{1}{(1-\alpha)(t_2-t_1)} \int_{\frac{t_1}{\vartheta(\varepsilon)}}^{\frac{\alpha t_1+(1-\alpha)t_2}{\vartheta(\varepsilon)}} \rho(y)dy \quad \text{and} \quad \rho(\bar{t}_2) = \frac{1}{\alpha(t_2-t_1)} \int_{\frac{\alpha t_1+(1-\alpha)t_2}{\vartheta(\varepsilon)}}^{\frac{t_2}{\vartheta(\varepsilon)}} \rho(y)dy.$$

Noting that $\bar{t}_1 \le \bar{t}_2$, we have $\rho(\bar{t}_1) \ge \rho(\bar{t}_2)$, which shows that inequality (15) holds.

(d) Assume that $\rho(u) \le c_1$ $(u \ge 0)$ for some constant $c_1 > 0$. Then, for any $t_1, t_2 \in \mathbb{R}_+$,

$$|\phi(\varepsilon, t_1) - \phi(\varepsilon, t_2)| = \left| \int_{t_1}^{t_2} \frac{1}{\vartheta(\varepsilon)} \rho\left(\frac{u}{\vartheta(\varepsilon)}\right) du \right| \le \frac{c_1|t_1 - t_2|}{\vartheta(\varepsilon)}.$$

This shows that the function $\phi(\varepsilon, \cdot)$ is globally Lipschitz continuous on $\mathbb{R}_+$. $\square$

Next we present several smoothing functions of θ by using the above framework.

Example 2.1. Let $\rho(u) := \begin{cases} 0 & \text{if } u < 0 \\ \frac{1}{(u+1)^2} & \text{if } u \ge 0 \end{cases}$ and $\vartheta(\varepsilon) \equiv \varepsilon$. Then we easily get

$$\hat{\delta}(\varepsilon, u) = \begin{cases} 0 & \text{if } u < 0 \\ \frac{1}{\varepsilon} \frac{1}{(u/\varepsilon+1)^2} & \text{if } u \ge 0 \end{cases} \quad \text{and } \phi_1(\varepsilon, t) = \int_{-\infty}^{t} \hat{\delta}(\varepsilon, u)du = \frac{t}{t+\varepsilon} \quad \forall t \ge 0.$$

By the expression of ρ and Proposition 2.1, the function $\phi_1(\varepsilon, \cdot)$ is continuously differentiable, concave and globally Lipschitz continuous on $\mathbb{R}_+$. Figure 1 below plots the curves of $\phi_1(\varepsilon, t)$ corresponding to different ε at $[0, 10]$ and $[0, 0.1]$, respectively. We see that $\phi_1(\varepsilon, \cdot)$ with $\varepsilon = 0.2$ yields a good approximation to θ at $[1, +\infty)$; and at $[0, 10^{-i}]$ the desired approximation of $\phi_1(\varepsilon, \cdot)$ to θ requires ε to be less than 10^{-i-2}.

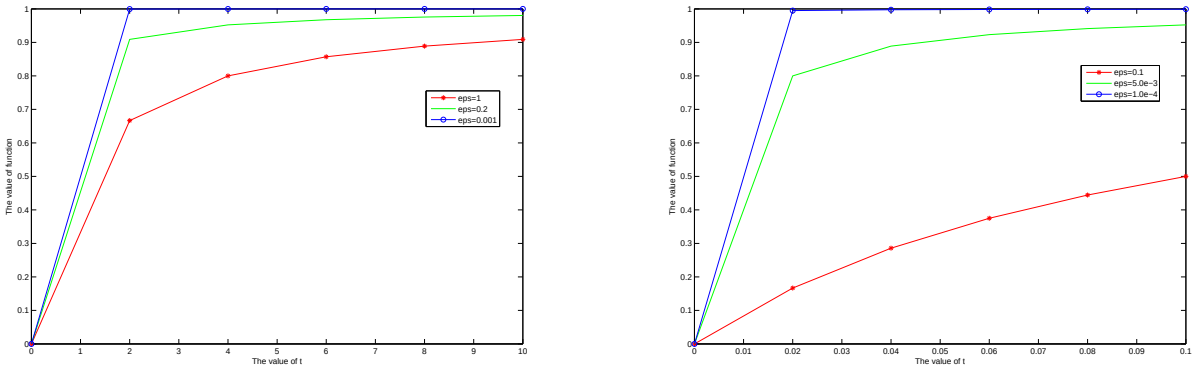

Figure 1: The curves of $\phi_1(\varepsilon, t)$ corresponding to different ε

Example 2.2. Let $\rho(u)=\left\{\begin{array}{ll} 0 & \text{if } u<0\\ \frac{pu^{p-1}}{(u^p+1)^2} & \text{if } u\geq 0\end{array}\right.$ with $p\geq 1$. When $\vartheta(\varepsilon)\equiv\varepsilon$, we have

$$\hat{\delta}(\varepsilon,u)=\left\{\begin{array}{ll} 0 & \text{if } u<0\\ \frac{1}{\varepsilon}\frac{p(u/\varepsilon)^{p-1}}{((\frac{u}{\varepsilon})^p+1)^2} & \text{if } u\geq 0\end{array}\right. \text{ and } \phi_p(\varepsilon,t)=\int_{-\infty}^{t}\hat{\delta}(\varepsilon,u)du=\frac{t^p}{t^p+\varepsilon^p}\quad \forall t\geq 0.$$

When $\vartheta(\varepsilon)\equiv\sqrt[p]{\varepsilon}$, we obtain that

$$\hat{\delta}(\varepsilon,u)=\left\{\begin{array}{ll} 0 & \text{if } u<0\\ \frac{1}{\sqrt[p]{\varepsilon}}\frac{p(u/\sqrt[p]{\varepsilon})^{p-1}}{((\frac{u}{\sqrt[p]{\varepsilon}})^p+1)^2} & \text{if } u\geq 0\end{array}\right. \text{ and } \varphi_p(\varepsilon,t)=\int_{-\infty}^{t}\hat{\delta}(\varepsilon,u)du=\frac{t^p}{t^p+\varepsilon}\quad \forall t\geq 0.$$

From Proposition 2.1, we know that the functions $\phi_p(\varepsilon,\cdot)$ and $\varphi_p(\varepsilon,\cdot)$ are concave on $[\varepsilon,+\infty)$ and $[\sqrt[p]{\varepsilon},+\infty)$, respectively, and they are globally Lipschitz continuous and continuously differentiable on $\mathbb{R}_+$. Figure 2 below plots the curves of $\phi_p(\varepsilon,t)$ corresponding to different p at $t\in[0,0.1]$ with $\varepsilon=0.01$ and at $t\in[0,10]$ with $\varepsilon=1$, respectively. We observe that ϕ_p with a larger p yields a better approximation to θ.

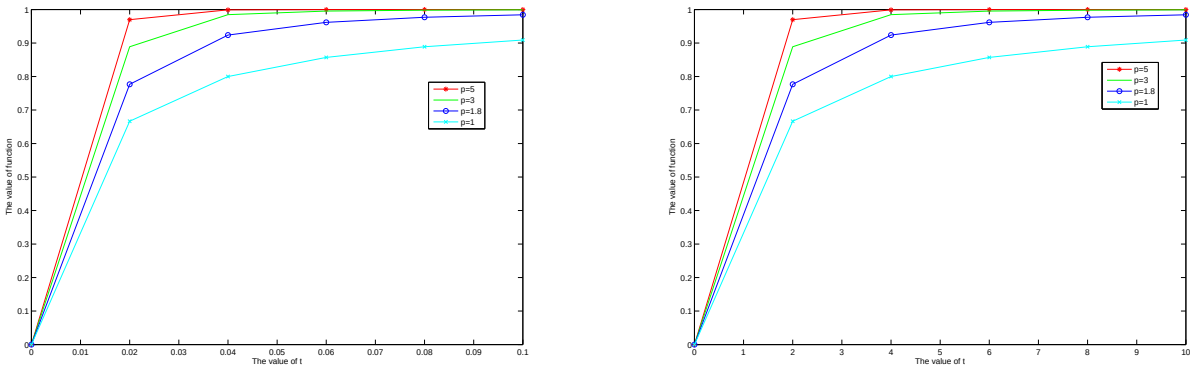

Figure 2: The curves of ϕ_p corresponding to different p

Figure 3 below plots the curves of $\varphi_p(\varepsilon,t)$ corresponding to different p at $t\in[0,0.1]$ with $\varepsilon=0.01$ and at $t\in[0,10]$ with $\varepsilon=0.3$, respectively. Comparing with Figure 2, we see that φ_p has a different performance from ϕ_p: when t has a small scale (for example $t\in[0,0.1]$), φ_p with a smaller p yields a better approximation to θ; and when t has a larger scale (for example $t\in[1,10]$), φ_p with a larger p yields a better approximation. We should mention that the function $\|\Phi(\varepsilon,X)\|_*$ defined with φ_2 was studied in [39].

Example 2.3. Let $\rho(u)=\left\{\begin{array}{ll} 0 & \text{if } u<0\\ \frac{1}{(u^p+1)^{1+\frac{1}{p}}} & \text{if } u\geq 0\end{array}\right.$ with $p>0$ and $\vartheta(\varepsilon)\equiv\varepsilon$. We have

$$\hat{\delta}(\varepsilon,u)=\left\{\begin{array}{ll} 0 & \text{if } u<0\\ \frac{1}{\varepsilon}\frac{1}{((\frac{u}{\varepsilon})^p+1)^{1+\frac{1}{p}}} & \text{if } u\geq 0\end{array}\right. \text{ and } \psi_p(\varepsilon,t)=\int_{-\infty}^{t}\hat{\delta}(\varepsilon,u)du=\frac{t}{\sqrt[p]{t^p+\varepsilon^p}}\quad \forall t\geq 0.$$

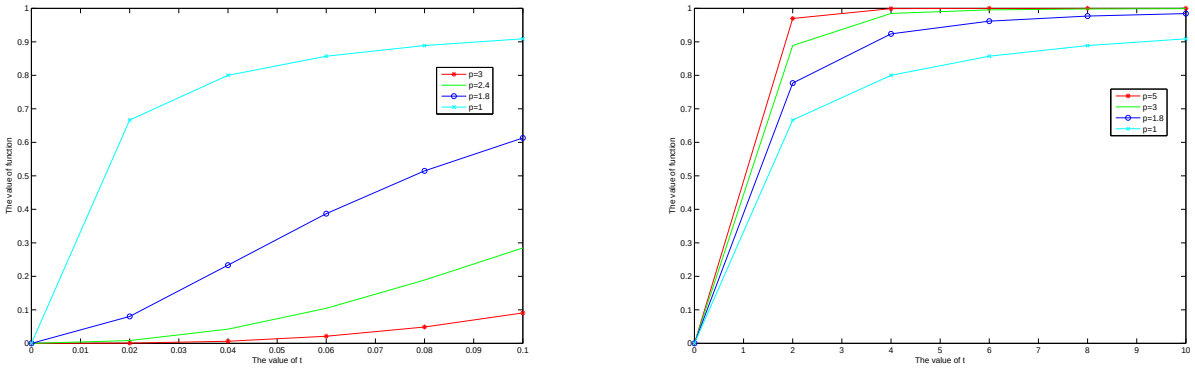

Figure 3: The curves of φ_p corresponding to different p

Clearly, $\psi_p = \sqrt[p]{\phi_p}$. The function $\psi_p(\varepsilon, \cdot)$ is continuously differentiable, concave and globally Lipschitz continuous on $\mathbb{R}_+$. The following figure plots the curves of $\psi_p(\varepsilon, t)$ corresponding to different p at $t \in [0, 0.1]$ with $\varepsilon = 0.01$ and at $t \in [0, 10]$ with $\varepsilon = 0.1$, respectively. We see that ψ_p with a larger p gives a better approximation to θ.

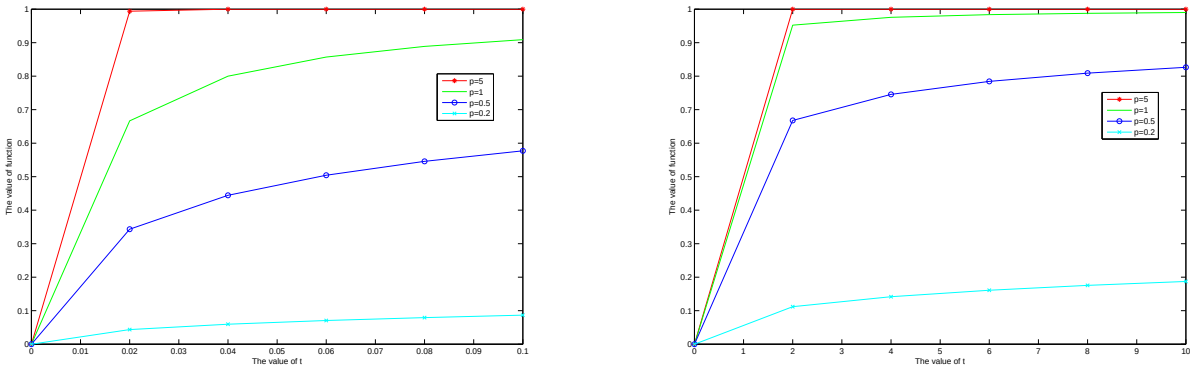

Figure 4: The curves of ψ_p corresponding to different p

Example 2.4. Let $\rho(u) = \begin{cases} 0 & \text{if } u < 0, \\ \exp(-u) & \text{if } u \geq 0 \end{cases}$ and $\vartheta(\varepsilon) \equiv \varepsilon$. Then we have

$$\hat{\delta}(\varepsilon, u) = \begin{cases} 0 & \text{if } u < 0 \\ \frac{1}{\varepsilon}\exp(-\frac{u}{\varepsilon}) & \text{if } u \geq 0 \end{cases} \quad \text{and } \varphi(\varepsilon, t) = \int_{-\infty}^{t} \hat{\delta}(\varepsilon, u)du = 1 - \exp\left(-\frac{t}{\varepsilon}\right) \quad \forall t \geq 0.$$

The function $\varphi(\varepsilon, \cdot)$ is continuously differentiable, concave and globally Lipschitz continuous on $\mathbb{R}_+$. Also, $\varphi(\varepsilon, t)$ with different ε displays a similar performance to $\phi_1(\varepsilon, t)$ and $\psi_p(\varepsilon, t)$. To check their approximation effect, we plot the curves of φ, ϕ_1 and ψ_2 at $t \in [0, 0.01]$ with $\varepsilon = 10^{-3}$, and at $t \in [0, 10]$ with $\varepsilon = 1$, respectively. Figure 5 below

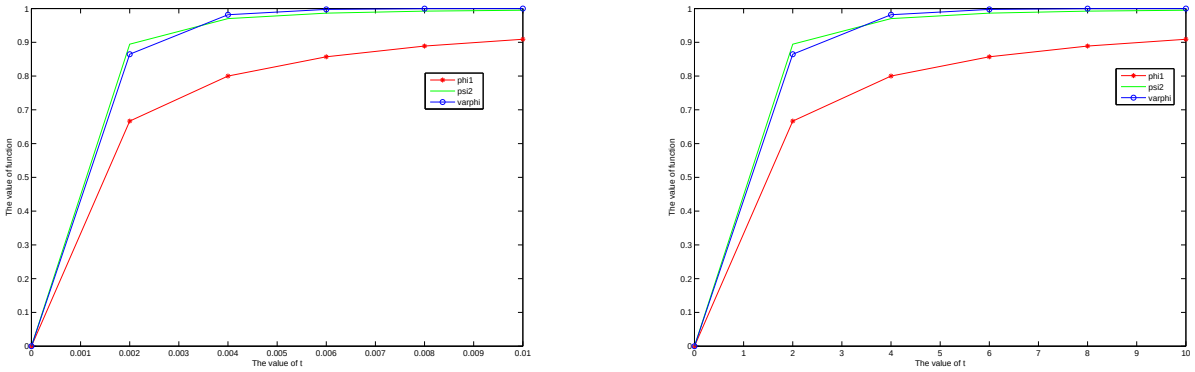

Figure 5: The curves of the functions ϕ_1, ψ_2 and φ

shows that among the three functions, ϕ_1 has the worst approximation; and ψ_2 and φ yield better approximation and their approximation effects are comparable.

Next we focus on $\|\Phi(\varepsilon, X)\|_*$ associated to the approximation functions yielded by (11) with a density function satisfying Assumption (A). The following proposition states the properties of $\|\Phi(\varepsilon, X)\|_*$, which particularly include its approximation to $\mathrm{rank}(\cdot)$.

Proposition 2.2 *Let $\Phi\colon \mathbb{R}_{++} \times \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ be defined by (7) where ϕ is yielded via formula (11) with a density function ρ satisfying Assumption (A). Then, for any given $\varepsilon > 0$ and $X \in \mathbb{R}^{m\times n}$, the following statements hold.*

(a) *For any $\varepsilon' \geq \varepsilon$, it holds that $\|\Phi(\varepsilon, X)\|_* \geq \|\Phi(\varepsilon', X)\|_*$.*

(b) *The function $\Xi(\cdot) \equiv \|\Phi(\varepsilon, \cdot)\|_*$ is directionally differentiable everywhere in $\mathbb{R}^{m\times n}$, and its directional derivative at $X \in \mathbb{R}^{m\times n}$ in the direction $H \in \mathbb{R}^{m\times n}$ is given by*

$$\Xi'(X; H) = \sum_{i=1}^{m} \frac{1}{\vartheta(\varepsilon)} \rho\left(\frac{\sigma_i(X)}{\vartheta(\varepsilon)}\right) \sigma_i'(X; H), \tag{16}$$

where $\sigma_i'(X; H)$ is the directional derivative of $\sigma_i(X)$ at X in the direction H.

(c) *The function $\|\Phi(\varepsilon, \cdot)\|_*$ is globally Lipschitz continuous and semismooth on $\mathbb{R}^{m\times n}$.*

(d) *$0 \leq \mathrm{rank}(X) - \|\Phi(\varepsilon, X)\|_* \leq \kappa\,(1 - \phi(\varepsilon, \sigma_\kappa(X)))$, where κ is the rank of X. Moreover if $\kappa\,(1 - \phi(\varepsilon, \sigma_\kappa(X))) < 1$, it holds that $\mathrm{rank}(X) = \lfloor \|\Phi(\varepsilon, X)\|_* \rfloor$.*

Proof. (a) The result is immediate by equation (8) and Proposition 2.1(b).

(b) By the definition of directional derivative and equation (8), we have that

$$\begin{aligned} \Xi'(X; H) &= \lim_{t\downarrow 0} \frac{\|\Phi(\varepsilon, X + tH)\|_* - \|\Phi(\varepsilon, X)\|_*}{t} \\ &= \sum_{i=1}^{m} \lim_{t\downarrow 0} \frac{\phi(\varepsilon, \sigma_i(X + tH)) - \phi(\varepsilon, \sigma_i(X))}{t}. \end{aligned} \tag{17}$$

By [9, Proposition 2.9] each σ_i is directionally differentiable everywhere, and hence

$$\sigma_i(X+tH) - \sigma_i(X) - t\sigma_i'(X;H) = o(t), \quad i = 1, 2, \ldots, m.$$

In addition, by Proposition 2.1(a), we know that $\phi(\varepsilon, \cdot)$ is continuously differentiable on $\mathbb{R}_+$, and its derivative at $t \geq 0$ is $\frac{1}{\vartheta(\varepsilon)}\rho(\frac{t}{\vartheta(\varepsilon)})$. Then for each $i = 1, \ldots, m$, we have

$$\begin{aligned} \phi(\varepsilon, \sigma_i(X+tH)) - \phi(\varepsilon, \sigma_i(X)) &= \phi(\varepsilon, \sigma_i(X) + t\sigma_i'(X;H) + o(t)) - \phi(\varepsilon, \sigma_i(X)), \\ &= t\frac{1}{\vartheta(\varepsilon)}\rho\left(\frac{\sigma_i(X)}{\vartheta(\varepsilon)}\right)\sigma_i'(X;H) + o(t). \end{aligned}$$

Together with equation (17), we immediately obtain the formula in (16).

(c) By [1, Page 63] and the definition of $\mathcal{B}(X)$ in [9, Eq.(31)], $\sigma_i(\cdot)$ is globally Lipschitz continuous on $\mathbb{R}^{m\times n}$. Along with Proposition 2.1(d) and (8), we get the global Lipschitz continuity of $\|\Phi(\varepsilon, X)\|_*$. In addition, combining [32, Theorem 4.7] with the definition of $\mathcal{B}(X)$, we know that $\sigma_i(\cdot),\ i = 1, 2, \ldots, m$ are strongly semismooth in $\mathbb{R}^{m\times n}$. Since $\tilde{\phi}(\varepsilon, \cdot)$ in (6) is continuously differentiable on $\mathbb{R}$, it follows that $\tilde{\phi}(\varepsilon, \sigma_i(X)),\ i = 1, 2, \ldots, m$ are semismooth in $\mathbb{R}^{m\times n}$. Then the semismoothness of $\|\Phi(\varepsilon, \cdot)\|_*$ follows by (8).

(d) Since the rank of X is equal to κ, we have that

$$\sigma_1(X) \geq \cdots \geq \sigma_\kappa(X) > \sigma_{\kappa+1}(X) = \cdots = \sigma_m(X) = 0.$$

From equations (8) and (11), it then follows that

$$\begin{aligned} 0 \leq \mathrm{rank}(X) - \|\Phi(\varepsilon, X)\|_* &= \sum_{i=1}^{\kappa} [\theta(\sigma_i(X)) - \phi(\varepsilon, \sigma_i(X))] \\ &= \sum_{i=1}^{\kappa} [1 - \phi(\varepsilon, \sigma_i(X))] = \sum_{i=1}^{\kappa} \int_{\frac{\sigma_i(X)}{\vartheta(\varepsilon)}}^{+\infty} \rho(y)dy \\ &\leq \kappa \int_{\frac{\sigma_\kappa(X)}{\vartheta(\varepsilon)}}^{+\infty} \rho(y)dy = \kappa\left(1 - \phi(\varepsilon, \sigma_\kappa(X))\right), \end{aligned} \tag{18}$$

where the second inequality is using $\sigma_1(X) \geq \cdots \geq \sigma_\kappa(X) > 0$. If $\kappa\left(1 - \phi(\varepsilon, \sigma_\kappa(X))\right) < 1$, we have

$$\|\Phi(\varepsilon, X)\|_* \leq \mathrm{rank}(X) \leq \|\Phi(\varepsilon, X)\|_* + \kappa\left(1 - \phi(\varepsilon, \sigma_\kappa(X))\right) < \|\Phi(\varepsilon, X)\|_* + 1, \tag{19}$$

since $\mathrm{rank}(X)$ is an integer, we have $\mathrm{rank}(X) = \lfloor \|\Phi(\varepsilon, X)\|_* \rfloor$. $\square$

Remark 2.1 *For any $\varepsilon > 0$, the nuclear norm $\|\Phi(\varepsilon, \cdot)\|_*$ is not differentiable in $\mathbb{R}^{m\times n}$, though the operator $\Phi(\varepsilon, \cdot)$ is continuously differentiable everywhere. In addition, $\|\Phi(\varepsilon, \cdot)\|_*$ is not necessarily concave in $\mathbb{R}^{m\times n}$ even if $\phi(\varepsilon, \cdot)$ is concave on $\mathbb{R}_+$.*

Remark 2.2 *For any $\varepsilon > 0$, however the operator $\Phi(\varepsilon, \cdot)$ is continuously differentiable in $\mathbb{R}^{m\times n}$, whether the nuclear norm $\|\Phi(\varepsilon, \cdot)\|_*$ is differentiable in $\mathbb{R}^{m\times n}$? In addition, $\phi(\varepsilon, \cdot)$ is concave on $\mathbb{R}_+$, whether $\|\Phi(\varepsilon, \cdot)\|_*$ is concave in $\mathbb{R}^{m\times n}$ or not?*

Remark 2.3 *For the smoothing functions ϕ in Examples 2.1-2.4, from Proposition 2.2(d) it follows that for any $\varepsilon > 0$ and $X \in \mathbb{R}^{m\times n}$ with $\mathrm{rank}(X) = \kappa$,*

$$\begin{aligned} 0 \le \mathrm{rank}(X) - \|\Phi(\varepsilon, X)\|_* &\le \frac{\kappa\varepsilon^p}{(\sigma_\kappa(X))^p + \varepsilon^p} \quad \text{when } \phi = \phi_p, \\ 0 \le \mathrm{rank}(X) - \|\Phi(\varepsilon, X)\|_* &\le \frac{\kappa(\sqrt[p]{(\sigma_\kappa(X))^p + \varepsilon^p} - \sigma_i(X))}{\sqrt[p]{(\sigma_i(X))^p + \varepsilon^p}} \quad \text{when } \phi = \psi_p, \\ 0 \le \mathrm{rank}(X) - \|\Phi(\varepsilon, X)\|_* &\le \kappa \exp\left(-\frac{\sigma_\kappa(X)}{\varepsilon}\right) \quad \text{when } \phi = \varphi. \end{aligned} \tag{20}$$

Particularly, for a given X with $\mathrm{rank}(X) = \kappa$, if $\sigma_\kappa(X) \ge \kappa\varepsilon$ for some $\varepsilon > 0$, then

$$\begin{aligned} \mathrm{rank}(X) - \|\Phi(\varepsilon, X)\|_* &\le \frac{\kappa}{1+\kappa^p} \quad \text{for } \phi = \phi_p, \\ \mathrm{rank}(X) - \|\Phi(\varepsilon, X)\|_* &\le \frac{\kappa(\sqrt[p]{\kappa^p + 1} - \kappa)}{\sqrt[p]{\kappa^p + 1}} \quad \text{for } \phi = \psi_p, \\ \mathrm{rank}(X) - \|\Phi(\varepsilon, X)\|_* &\le \kappa\exp(-\kappa) \quad \text{for } \phi = \varphi. \end{aligned} \tag{21}$$

Remark 2.4 *Remark 2.3 and Proposition 2.2(a)(d) show that the smoothing functions ϕ in Examples 2.1-2.4 have the following favorable property: for any given $X \in \mathbb{R}^{m\times n}$ with $\mathrm{rank}(X) = \kappa$, then for all $0 < \varepsilon \le \frac{\sigma_\kappa(X)}{\kappa}$,*

$$0 \le \mathrm{rank}(X) - \|\Phi(\varepsilon, X)\|_* \le \mathrm{rank}(X) - \|\Phi(\widehat{\varepsilon}, X)\|_* \le c(\kappa),$$

where $c(\kappa)$ is a positive constant defined as follows

$$\begin{aligned} c(\kappa) &= \frac{\kappa}{1+\kappa^p} \quad \text{when} \quad \phi = \phi_p, \\ c(\kappa) &= \frac{\kappa(\sqrt[p]{\kappa^p + 1} - \kappa)}{\sqrt[p]{\kappa^p + 1}} \quad \text{when} \quad \phi = \psi_p, \\ c(\kappa) &= \kappa\exp(-\kappa) \quad \text{when} \quad \phi = \varphi. \end{aligned} \tag{22}$$

Clearly, when $c(\kappa) < 1$, we have that $\mathrm{rank}(X) = \lfloor\|\Phi(\varepsilon, X)\|_*\rfloor$, where $\lfloor\|\Phi(\varepsilon, X)\|_*\rfloor$ denotes the minimum integer greater than $\|\Phi(\varepsilon, X)\|_*$.

Remark 2.4 shows that for those $X \in \mathbb{R}^{m\times n}$ whose smallest nonzero singular value has a large scale, $\|\Phi(\varepsilon, X)\|_*$ is a good approximation to $\mathrm{rank}(X)$ even if ε is not small enough. This motivates us to consider the following approximation problems

$$\begin{aligned} \min \ & f(X) \\ \text{s.t.} \ & \|\Phi(\varepsilon, X)\|_* \le r, \\ & X \in \Omega \end{aligned} \tag{23}$$

and

$$\begin{aligned} &\min \ f(X) + \nu\|\Phi(\varepsilon, X)\|_* \\ &\text{s.t.} \ \ X \in \Omega \end{aligned} \tag{24}$$

where $\Phi(\varepsilon, X)$ is defined by (7) with one of the smoothing functions ϕ_p, ψ_p and φ.

Next we focus on the approximation properties of the two problems. To this end, for any given $\varepsilon > 0$, let $\mathcal{F}_\varepsilon$ and $\mathcal{S}_\varepsilon$ denote the feasible set and the optimal solution set of (23), respectively, and $\mathcal{F}$ and $\mathcal{S}$ denote the feasible set and the optimal solution set of (1), respectively. Using Proposition 2.2, we have the following result for (1) and (23).

Proposition 2.3 *For problem (1) and its approximation (23), the following results hold.*

(a) *For any $0 < \varepsilon_1 \le \varepsilon_2$, it holds that $\mathcal{F}_{\varepsilon_1} \subset \mathcal{F}_{\varepsilon_2}$, and consequently, $\mathcal{F} = \bigcap_{\varepsilon>0} \mathcal{F}_\varepsilon$.*

(b) *For any $0 < \varepsilon_1 \le \varepsilon_2$, we have $f(X^*_{\varepsilon_2}) \le f(X^*_{\varepsilon_1})$ where $X^*_{\varepsilon_2} \in \mathcal{S}_{\varepsilon_2}$ and $X^*_{\varepsilon_1} \in \mathcal{S}_{\varepsilon_1}$.*

(c) *For any $\varepsilon > 0$, if $X^*_\varepsilon \in \mathcal{S}_\varepsilon$ with $\mathrm{rank}(X^*_\varepsilon) \le r$, then $X^*_\varepsilon \in \mathcal{S}$.*

(d) *For any $\varepsilon_k \downarrow 0$, if $X^*_{\varepsilon_k} \in \mathcal{S}_{\varepsilon_k}$ and $X^*_{\varepsilon_k} \to \widetilde{X}^*$, then $\widetilde{X}^*$ is an optimal solution of (1).*

Proof. (a) The result follows directly from Proposition 2.2(a).

(b) By part (a), $\mathcal{F}_{\varepsilon_1} \subset \mathcal{F}_{\varepsilon_2}$. This means that $X^*_{\varepsilon_1} \in \mathcal{F}_{\varepsilon_2}$, and hence $f(X^*_{\varepsilon_2}) \le f(X^*_{\varepsilon_1})$.

(c) Suppose that the solution X^*_ε of (23) satisfies $\mathrm{rank}(X^*_\varepsilon) \le r$. It is clear that X^*_ε is a feasible solution to (1). Thus, to prove the result, it suffices to argue that

$$f(X) \ge f(X^*_\varepsilon) \qquad \forall\, X \in \mathcal{F}. \tag{25}$$

Suppose on the contrary that there is an $X \in \mathcal{F}$ with $f(X) < f(X^*_\varepsilon)$. Since $\mathrm{rank}(X) \le r$, it follows from Proposition 2.2(d) that $\Phi(\varepsilon, X) \le r$ for any $\varepsilon > 0$. This means that X is also a feasible solution to (23). Note that X^*_ε is an optimal solution to (23). Therefore, we have that $f(X^*_\varepsilon) \le f(X)$, which gives a contradiction to $f(X) < f(X^*_\varepsilon)$.

(d) Note that $\mathcal{F}_\varepsilon$ is closed. By part (a), $\mathcal{F}$ is also closed, and consequently, $\widetilde{X}^* \in \mathcal{F}$. Let X^* be an optimal solution of (1). Note that $X^* \in \mathcal{F}_{\varepsilon_k}$ for each k. It then follows that $f(X^*_{\varepsilon_k}) \le f(X^*)$. Taking the limit $k \to \infty$ and using the continuity of f yields that $f(\widetilde{X}^*) \le f(X^*)$. Together with $\widetilde{X}^* \in \mathcal{F}$ and $X^* \in \mathcal{S}$, we get the result. $\square$

For problem (2) and its approximation problem (24), we have the following result.

Proposition 2.4 *For any $\varepsilon > 0$, let X^*_ε be an optimal solution of (24) associated to ε. Then, the following statements hold.*

(a) *For any $0 < \varepsilon_1 \le \varepsilon_2$, it holds that $f(X^*_{\varepsilon_1}) + \nu\|\Phi(\varepsilon_1, X^*_{\varepsilon_1})\|_* \ge f(X^*_{\varepsilon_2}) + \nu\|\Phi(\varepsilon_2, X^*_{\varepsilon_2})\|_*$.*

(b) *If* $\mathrm{rank}(X_\varepsilon^*) = \kappa$ *and* $\sigma_\kappa(X_\varepsilon^*) \geq \kappa\varepsilon$, *then we have the following inequality*

$$f(X_\varepsilon^*) + \nu\mathrm{rank}(X_\varepsilon^*) \leq f(X) + \nu\mathrm{rank}(X) + \nu c(\kappa) \quad \forall X \in \Omega,$$

and particularly, the optimal values of (2) and (24) have the following relation

$$f(X_\varepsilon^*) + \nu\mathrm{rank}(X_\varepsilon^*) \leq f(X^*) + \nu\mathrm{rank}(X^*) + \nu c(\kappa)$$

where X^* *is an optimal solution of (2), and* $c(\kappa)$ *is given by (22).*

(c) *For any* $\varepsilon_k \downarrow 0$, *if* $X_{\varepsilon_k}^* \to \widehat{X}^*$, *then* $\widehat{X}^*$ *is an optimal solution to (2).*

Proof. (a) Note that for problem (24) with $\varepsilon = \varepsilon_2$, $X_{\varepsilon_2}^*$ is an optimal solution, whereas $X_{\varepsilon_1}^*$ is only a feasible one. Therefore, the following inequality holds:

$$f(X_{\varepsilon_1}^*) + \nu\|\Phi(\varepsilon_2, X_{\varepsilon_1}^*)\|_* \geq f(X_{\varepsilon_2}^*) + \nu\|\Phi(\varepsilon_2, X_{\varepsilon_2}^*)\|_*.$$

On the other hand, applying Proposition 2.2(a), we have that

$$f(X_{\varepsilon_1}^*) + \nu\|\Phi(\varepsilon_1, X_{\varepsilon_1}^*)\|_* \geq f(X_{\varepsilon_1}^*) + \nu\|\Phi(\varepsilon_2, X_{\varepsilon_1}^*)\|_*.$$

Combining the last two inequalities, we immediately obtain the result.

(b) Let $X \in \mathbb{R}^{m\times n}$ be an arbitrary feasible solution to (2). Then, it holds that

$$f(X_\varepsilon^*) + \nu\|\Phi(\varepsilon, X_\varepsilon^*)\|_* \leq f(X) + \nu\|\Phi(\varepsilon, X)\|_* \leq f(X) + \nu\mathrm{rank}(X),$$

where the first inequality is using $X \in \Omega$, and the second one is due to Proposition 2.2(d). In addition, from Remark 2.4 we have that

$$f(X_\varepsilon^*) + \nu\mathrm{rank}(X_\varepsilon^*) - \nu c(\kappa) \leq f(X_\varepsilon^*) + \nu\|\Phi(\varepsilon, X_\varepsilon^*)\|_*.$$

From the last two equations, we readily obtain the result.

(c) Let X^* be an optimal solution to (2). For each k, it holds that

$$f(X^*) + \nu\mathrm{rank}(X^*) \geq f(X^*) + \nu\|\Phi(\varepsilon_k, X^*)\|_* \geq f(X_{\varepsilon_k}^*) + \nu\|\Phi(\varepsilon_k, X_{\varepsilon_k}^*)\|_*$$

Taking the limit $k \to \infty$ and noting that $\widehat{X}^* \in \Omega$, we get the result. □

3 Convex relaxation method

Proposition 2.3(c) and (d) show that an optimal solution of (1) can be got by solving a single problem (23) with a suitable small $\varepsilon > 0$ or a sequence of problems (23) with a decreasing $\varepsilon > 0$. Similar results also hold for (2) via its approximation problem (24). However, for a given $\varepsilon > 0$, problems (23) and (24) are still nonconvex for which one expects only a local optimal solution via direct solutions. To seek a favorable local optimal

solution, one often handles their convex relaxation problems. In this section, we propose a convex relaxation method with the approximation functions $\|\Phi(\varepsilon, X)\|_*$ associated with ϕ_p and ψ_p for the rank minimization problems with the PSD cone constraint

$$\begin{array}{ll} \min & f(X) \\ \text{s.t.} & \mathcal{A}X \in b + \mathcal{Q}, \\ & \mathrm{rank}(X) \le r, \\ & X \succeq 0 \end{array} \tag{26}$$

where $\mathcal{A}\colon \mathbb{S}^n \to \mathbb{R}^m$ is a linear operator with its adjoint operator denoted by $\mathcal{A}^*$, and $\mathcal{Q} := \{0\}^{m_1} \times \mathbb{R}^{m_2}_+$ is a polyhedral convex cone with $1 \le m_1 \le m$ and $m_1 + m_2 = m$.

Under the PSD cone constraint $X \succeq 0$, $\|\Phi(\varepsilon, X)\|_*$ reduces to the trace function

$$\begin{aligned} \langle \Psi_1(\varepsilon, X), X\rangle \ \text{ with } \ \Psi_1(\varepsilon, X) &:= \frac{X^{p-1}}{X^p + \varepsilon^p} \quad \text{when } \phi = \phi_p, \\ \langle \Psi_2(\varepsilon, X), X\rangle \ \text{ with } \ \Psi_2(\varepsilon, X) &:= \frac{1}{\sqrt[p]{X^p + \varepsilon^p}} \quad \text{when } \phi = \psi_p. \end{aligned} \tag{27}$$

Note that for a convergent sequence $\{(\varepsilon_k, X^k)\} \subset \mathbb{R}_{++} \times \mathbb{S}^n_+$ with $\varepsilon_k \downarrow 0$ and $X^k \to \widehat{X}$, the sequences $\langle \Psi_1(\varepsilon_k, X^k), X^k\rangle$ and $\langle \Psi_2(\varepsilon_k, X^k), X^k\rangle$ have a similar convergence behavior to $\langle \Psi_1(\varepsilon_k, X^k),\ X^{k+1}\rangle$ and $\langle \Psi_2(\varepsilon_k, X^k),\ X^{k+1}\rangle$, respectively, and particularly, the functions $\langle \Psi_1(\varepsilon_k, X^k), X\rangle$ and $\langle \Psi_2(\varepsilon_k, X^k), X\rangle$ are both linear. This observation motivates us to replace the solution of the kth nonconvex approximate subproblem of (26), that is,

$$\begin{array}{ll} \min\limits_{X \in \mathbb{S}^n} & f(X) \\ \text{s.t.} & \mathcal{A}X \in b + \mathcal{Q}, \\ & \|\Phi(\varepsilon^k, X)\|_* \le r, \\ & X \succeq 0, \end{array}$$

by fixing $\Psi_i(\varepsilon_k, X^k)$ with $i = 1$ or 2 and solving the following convex programming

$$\begin{array}{ll} \min\limits_{X \in \mathbb{S}^n} & f(X) \\ \text{s.t.} & \mathcal{A}X \in b + \mathcal{Q}, \\ & \langle \Psi_i(\varepsilon_k, X^k), X\rangle = r, \\ & X \succeq 0. \end{array} \tag{28}$$

To obtain a solution of (28), it suffices to find a KKT point, i.e., the point satisfying

$$\left\{ \begin{array}{l} \nabla f(X) - \mathcal{A}^* y - s\Psi_i(\varepsilon_k, X^k) - S = 0, \\ \langle \Psi_i(\varepsilon_k, X^k), X\rangle - r = 0, \\ y \in \mathcal{Q}^*, \ \ \mathcal{A}X - b \in \mathcal{Q}, \ \ \langle y, AX - b\rangle = 0, \\ X \succeq 0, \ \ S \succeq 0, \ \ \langle X, S\rangle = 0, \end{array} \right. \tag{29}$$

where $\mathcal{Q}^* = \mathbb{R}^{m_1} \times \mathbb{R}^{m_2}_+$ is the dual cone of $\mathcal{Q}$. The conditions (29) can be rewritten as

$$\begin{cases} R_d(w) := \nabla f(X) - \mathcal{A}^* y - s\Psi_i(\varepsilon_k, X^k) - S = 0, \\ R_p(w) := \langle \Psi_i(\varepsilon_k, X^k), X \rangle - r = 0, \\ R_1(w) := y - \Pi_{\mathcal{Q}^*}[y - (\mathcal{A}X - b)] = 0, \\ R_2(w) := X - \Pi_{\mathbb{S}^n_+}[X - S] = 0, \end{cases} \tag{30}$$

with $w = (X, y, s, S) \in \mathbb{S}^n \times \mathbb{R}^m \times \mathbb{R} \times \mathbb{S}^n$, where $\Pi_{\mathcal{Q}}(\cdot)$ and $\Pi_{\mathbb{S}^n_+}(\cdot)$ are the projection operators onto $\mathcal{Q}$ and $\mathbb{S}^n_+$, respectively. Therefore, in the algorithm we may use

$$|R_p(w^{k+1})| + \|R_d(w^{k+1})\| + \|R_1(w^{k+1})\| + \|R_2(w^{k+1})\|$$

to measure the quality of solution $w^{k+1} = (X^{k+1}, y^{k+1}, s^{k+1}, S^{k+1}) \in \mathbb{S}^n \times \mathbb{R}^m \times \mathbb{R} \times \mathbb{S}^n$ to (28). Now we describe the iteration steps of our convex relaxation methods.

Algorithm 3.1 *(Convex Relaxation Method for (26))*

(S.0) *Given a tolerance sequence $\epsilon_k \downarrow 0$. Choose a starting point $(\varepsilon_0, X^0) \in \mathbb{R}_{++} \times \mathbb{R}^{m \times n}$ and a constant $\rho \in (0, 1)$. Set $k := 0$.*

(S.1) *Compute $\Psi_i(\varepsilon_k, X^k)$ by the eigenvalue decomposition of X^k.*

(S.2) *Find an approximate KKT point $w^{k+1} = (X^{k+1}, y^{k+1}, s^{k+1}, S^{k+1})$ of (28) such that*

$$|R_p(w^{k+1})| + \|R_d(w^{k+1})\| + \|R_1(w^{k+1})\| + \|R_2(w^{k+1})\| \le \epsilon_{k+1}.$$

(S.3) *Set $\varepsilon_{k+1} := \rho\varepsilon_k$ and $k := k + 1$. Then, go to Step (S.1).*

In Algorithm 3.1, the explicit form of dual variables $(y^{k+1}, s^{k+1}, S^{k+1})$ is not required, but their information plays an important role in the following convergence analysis. In order to study the convergence of Algorithm 3.1, we need the following two lemmas.

Lemma 3.1 *Let $\{X^k\}$ be a convergent sequence in $\mathbb{R}^{m \times n}$ with $X^k \to X^*$. Then, there exists $\bar{k}$ such that for all $k \ge \bar{k}$,* $\mathrm{rank}(X^k) \ge \mathrm{rank}(X^*)$.

Proof. Without loss of generality, we assume that $\mathrm{rank}(X^*) = \kappa$ and X^* has the SVD

$$U^T X^* V = \begin{bmatrix} \Sigma^*_\kappa & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

where $\Sigma^*_\kappa = \mathrm{Diag}(\sigma_1(X^*), \ldots, \sigma_\kappa(X^*))$ with $\sigma_i(X^*) > 0,\ i = 1, \ldots, \kappa$. For each k, let

$$U^T X^k V = \begin{bmatrix} \widetilde{X}^k_{11} & \widetilde{X}^k_{12} & \widetilde{X}^k_{13} \\ \widetilde{X}^k_{21} & \widetilde{X}^k_{22} & \widetilde{X}^k_{23} \end{bmatrix} \quad \text{with} \quad \widetilde{X}^k_{11} \in \mathbb{R}^{\kappa \times \kappa}.$$

Since $U^T X^k V \to U^T X^* V$, we have $\widetilde{X}^k_{11} \to \Sigma^*_\kappa$. This together with the nonsingularity of Σ^*_κ implies that $\widetilde{X}^k_{11}$ is nonsingular for all sufficiently large k, and then $\mathrm{rank}(\widetilde{X}^k_{11}) = \kappa$. Combining with the expression of $U^T X^k V$, we obtain the desired result. $\square$

Lemma 3.2 *For any $\varepsilon > 0$ and $X \in \mathbb{S}^n$, let $\Psi_i(\varepsilon, X)$ be defined by (27). Then, for any $\{(\varepsilon_k, X^k)\} \subset \mathbb{R}_{++} \times \mathbb{R}^{m\times n}$ with $\varepsilon_k \downarrow 0$ and $X^k \to X^*$, the following inequalities hold*

$$\operatorname{rank}(X^*) \le \liminf_{k\to\infty} \langle \Psi_i(\varepsilon_k, X^k), X^k\rangle \le \liminf_{k\to\infty} \operatorname{rank}(X^k), \tag{31}$$

$$\operatorname{rank}(X^*) \le \liminf_{k\to\infty} \langle \Psi_i(\varepsilon_k, X^k), X^{k+1}\rangle. \tag{32}$$

Proof. Assume that $i = 1$ and $\operatorname{rank}(X^*) = \kappa$. From Proposition 2.2(d), it follows that

$$\langle \Psi_1(\varepsilon_k, X^k), X^k\rangle = \|\Phi(\varepsilon_k, X^k)\|_* \le \operatorname{rank}(X^k) \quad \text{for each } k.$$

This implies that $\liminf_{k\to\infty}\langle \Psi_1(\varepsilon_k, X^k), X^k\rangle \le \liminf_{k\to\infty} \operatorname{rank}(X^k)$. Now assume that X^k has the spectral decomposition $X^k = P^k\Lambda^k(P^k)^T$, where P^k is an $n \times n$ orthogonal matrix, and $\Lambda^k = \operatorname{Diag}(\lambda_1(X^k), \ldots, \lambda_n(X^k))$ with $\lambda_1(X^k) \ge \cdots \ge \lambda_n(X^k) \ge 0$. Then,

$$\big\langle \Psi_1(\varepsilon_k, X^k), X^k\big\rangle = \sum_{i=1}^{n} \frac{[\lambda_i(X^k)]^p}{[\lambda_i(X^k)]^p + (\varepsilon_k)^p} \ge \sum_{i=1}^{\kappa} \frac{[\lambda_i(X^k)]^p}{[\lambda_i(X^k)]^p + (\varepsilon_k)^p}.$$

Since $\lim_{k\to\infty} \lambda_i(X^k) = \lambda_i(X^*) > 0$ for $i = 1, \ldots, \kappa$, from the last inequality we have

$$\liminf_{k\to\infty} \big\langle \Psi(\varepsilon_k, X^k), X^k\big\rangle \ge \liminf_{k\to\infty} \sum_{i=1}^{\kappa} \frac{[\lambda_i(X^k)]^p}{[\lambda_i(X^k)]^p + (\varepsilon_k)^p} = \kappa.$$

Thus, we complete the proof of (31). Next we prove the inequality (32). By Lemma 3.1, $\operatorname{rank}(X^k) \ge \kappa$ for all sufficiently large k, and then an elementary calculation yields that

$$\begin{aligned}
& \langle \Psi(\varepsilon_k, X^k), X^{k+1}\rangle \\
= \;& \big\langle P^k(\Lambda^k)^{p-1}[(\Lambda^k)^p + (\varepsilon_k)^p I]^{-1}(P^k)^T, P^{k+1}\Lambda^{k+1}(P^{k+1})^T\big\rangle \\
= \;& \big\langle (P^{k+1})^T P^k(\Lambda^k)^{p-1}[(\Lambda^k)^p + (\varepsilon_k)^p I]^{-1}(P^k)^T P^{k+1}, \Lambda^{k+1}\big\rangle \\
\ge \;& \left\langle (P^{k+1})^T P^k(\Lambda^k)^{p-1}[(\Lambda^k)^p + (\varepsilon_k)^p I]^{-1}(P^k)^T P^{k+1}, \begin{bmatrix} \Lambda^{k+1}_{\kappa\kappa} & 0 \\ 0 & 0 \end{bmatrix}\right\rangle \\
\ge \;& \left\langle (P^{k+1})^T P^k \begin{bmatrix} (\Lambda^k_{\kappa\kappa})^{p-1}(\Lambda^k_{\kappa\kappa} + \varepsilon_k I)^{-1} & 0 \\ 0 & 0 \end{bmatrix} (P^k)^T P^{k+1}, \begin{bmatrix} \Lambda^{k+1}_{\kappa\kappa} & 0 \\ 0 & 0 \end{bmatrix}\right\rangle \\
= \;& \left\langle \begin{bmatrix} (\Lambda^k_{\kappa\kappa})^{p-1}(\Lambda^k_{\kappa\kappa} + \varepsilon_k I)^{-1} & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} \Lambda^{k+1}_{\kappa\kappa} & 0 \\ 0 & 0 \end{bmatrix}\right\rangle + o(1),
\end{aligned}$$

where the first inequality is using $\Lambda^k \succeq 0$ and $\Lambda^{k+1} \succeq 0$, the second is due to $\operatorname{rank}(X^k) \ge \kappa$, and the last equality is using $(P^{k+1})^T P^k = I + o(1)$. From the last inequality, we get

$$\liminf_{k\to\infty}\langle \Psi(\varepsilon_k, X^k), X^{k+1}\rangle \ge \liminf_{k\to\infty} \sum_{i=1}^{\kappa} \left[\frac{[\lambda_i(X^k)]^{p-1}\lambda_i(X^{k+1})}{[\lambda_i(X^k)]^p + (\varepsilon_k)^p} + o(1)\right] = \kappa.$$

When $i = 2$, using the same arguments as above yields the result. $\square$

Let $\{X^k\}$ be a sequence generated by Algorithm 3.1. Assume that $\{X^k\}$ is bounded and X^* is an arbitrary limit point. Without loss of generality, we may assume that $\{X^k\}$ converges to X^*. Then, from Lemma 3.2 and Step 2 of Algorithm 3.1, it follows that

$$\mathrm{rank}(X^*) \le \liminf_{k\to\infty} \langle \Psi(\varepsilon_k, X^k), X^{k+1}\rangle \le \liminf_{k\to\infty} \epsilon_{k+1} + r = r,$$

which means that X^* must be a feasible solution to problem (26). In fact, by letting

$$X^* = P\Lambda^* P^T \ \text{ and } \ U^* = P_1 P_1^T \tag{33}$$

where $\Lambda^* = \mathrm{Diag}(\lambda_1^*, \ldots, \lambda_n^*)$ with $\lambda_1^* \ge \cdots \ge \lambda_n^*$, P is an $n \times n$ orthogonal matrix, and P_1 is the submatrix of P consisting of the first rth column, Theorem 3.1 below shows that (X^*, U^*) is a stationary point of the following optimization problem

$$\begin{aligned} \min_{(X,U)\in\mathbb{S}^n\times\mathbb{S}^n} \ & f(X) \\ \text{s.t.} \ \ & \mathcal{A}X \in b + \mathcal{Q}, \\ & \langle X, U - I\rangle = 0, \ \langle I, U\rangle = r \\ & I - U \succeq 0, \ \ U \succeq 0, \ \ X \succeq 0. \end{aligned} \tag{34}$$

Note that (34) is equivalent to problem (26) by the arguments of [17, Theorem 1.1].

Theorem 3.1 *Let $\{(X^k, \varepsilon_k, y^k, s^k, S^k)\}$ be a bounded sequence generated by Algorithm 3.1. Suppose that $(X^*, 0, y^*, s^*, S^*)$ is an arbitrary limit point with X^* having the spectral decomposition (33). Then (X^*, U^*) with U^* defined as above is a stationary point of (34).*

Proof. Let $y \in \mathbb{R}^m$ be the Lagrange multiplier of constraint $\mathcal{A}X \in b + \mathcal{Q}$, $\mu \in \mathbb{R}$ and $\upsilon \in \mathbb{R}$ are the multipliers associated with constraints $\langle X, U - I\rangle = 0$ and $\langle I, U\rangle = r$, respectively, and $R, V, S \in \mathbb{S}^n$ are the multipliers associated with the three PSD conic constraints, respectively. Then, the KKT conditions of problem (34) take the form

$$\left\{ \begin{array}{l} \nabla f(X) - \mathcal{A}^* y - \mu U - S = 0, \\ -\mu X - \upsilon I + R - V = 0, \\ y \in \mathcal{Q}^*, \ \ \mathcal{A}X - b \in \mathcal{Q}, \ \ \langle y, \mathcal{A}X - b\rangle = 0, \\ \langle X, I - U\rangle = 0, \ \ \langle I, U\rangle = r, \\ R \succeq 0, \ I - U \succeq 0, \ \langle R, I - U\rangle = 0, \\ U \succeq 0, \ V \succeq 0, \ \langle U, V\rangle = 0, \\ X \succeq 0, \ S \succeq 0, \ \langle X, S\rangle = 0. \end{array} \right. \tag{35}$$

Recall that (X, U) is called a stationary point of problem (34) if there exists a Lagrange multiplier $(y, S, \mu, \upsilon, R, V)$ such that $(X, U, y, S, \mu, \upsilon, R, V)$ satisfies (35).

Now without loss of generality, we suppose that the sequence $\{(X^k, \varepsilon_k, y^k, s^k, S^k)\}$ converges to $(X^*, 0, y^*, s^*, S^*)$. From Step 2 of Algorithm 3.1 and equation (30), we have

$$\begin{aligned} & \|\nabla f(X^{k+1}) - \mathcal{A}^* y^{k+1} - s^{k+1}\Psi_i(\varepsilon_k, X^k) - S^{k+1}\| \le \epsilon_{k+1}, \\ & |\langle \Psi_i(\varepsilon_k, X^k), X^{k+1}\rangle - r| \le \epsilon_{k+1}, \\ & \|y^{k+1} - \Pi_{\mathcal{Q}^*}[y^{k+1} - (\mathcal{A}X^{k+1} - b)]\| \le \epsilon_{k+1}, \\ & \|X^{k+1} - \Pi_{\mathbb{S}^n_+}[X^{k+1} - S^{k+1}]\| \le \epsilon_{k+1}. \end{aligned}$$

Therefore,

$$\begin{aligned}
&\nabla f(X^*) - \mathcal{A}^* y^* - \lim_{k\to\infty} s^{k+1}\Psi_i(\varepsilon_k, X^k) - S^* = 0,\\
&\lim_{k\to\infty} \langle \Psi_i(\varepsilon_k, X^k), X^{k+1}\rangle - r = 0,\\
&y^* - \Pi_{\mathcal{Q}^*}[y^* - (\mathcal{A}X^* - b)] = 0,\\
&X^* - \Pi_{\mathbb{S}^n_+}[X^* - S^*] = 0, \quad s^* = 0.
\end{aligned} \tag{36}$$

Let $\overline{S} := \lim_{k\to\infty} s^{k+1}\Psi_i(\varepsilon_k, X^k) + S^*$. Then $\overline{S} \succeq 0$. Also, by using $\langle S^*, X^*\rangle = 0$ and $s^* = 0$, we have $\langle \overline{S}, X^*\rangle = 0$. Together with (36) and Lemma 3.2, it follows that

$$\left\{\begin{array}{l}
\nabla f(X^*) - \mathcal{A}^* y^* - \overline{S} = 0,\\
\mathrm{rank}(X^*) \le r,\\
y^* - \Pi_{\mathcal{Q}^*}[y^* - (\mathcal{A}X^* - b)] = 0,\\
X^* - \Pi_{\mathbb{S}^n_+}[X^* - \overline{S}] = 0.
\end{array}\right. \tag{37}$$

Assume that X^* has the spectral decomposition as in (33) and U^* is given as in (33). Then, $X^* \succeq 0$ and $\mathrm{rank}(X^*) \le r$ imply that $\langle I, U^*\rangle = r$ and $\langle X^*, U^* - I\rangle = 0$. Note that the fourth condition of (37) means $X^* \succeq 0$. From equation (37) it then follows that

$$\left\{\begin{array}{l}
\nabla f(X^*) - \mathcal{A}^* y^* - \overline{S} = 0,\\
\langle X^*, U^* - I\rangle = 0, \quad \langle I, U^*\rangle = r,\\
y^* - \Pi_{\mathcal{Q}^*}[y^* - (\mathcal{A}X^* - b)] = 0,\\
X^* - \Pi_{\mathbb{S}^n_+}[X^* - \overline{S}] = 0.
\end{array}\right. \tag{38}$$

It is easy to see that $(X^*, U^*, y^*, \overline{S}, 0, 0, 0, 0)$ satisfies the KKT conditions (35). Hence, (X^*, U^*) is a stationary point of problem (26). The proof is completed. $\square$

To close this section, we remark that if the constraints of rank minimization problem (2) involves the cone constraint $X \succeq 0$, then one may consider the convex relaxation method by solving a sequence of subproblems with the following forms

$$\begin{aligned}
\min_{X\in\mathbb{S}^n} \;& f(X) + \nu\langle \Psi_i(\varepsilon_k, X^k), X\rangle\\
\text{s.t.} \;& X \in \Omega.
\end{aligned} \tag{39}$$

Note that Fazel, Hindi and Boyd [11] proposed a heuristic method via the solution of a sequence of subproblems (39) involving Ψ_1 with $p = 1$. Different from our method, their convex relaxation method was proposed by linearizing the log-det function. In addition, they applied their method for finding the minimum-order system realization with time-domain constraints and the lowest-dimension embedding of points in a Euclidean space. In the next section, we apply Algorithm 3.1 for the nearest low-rank correlation matrix.

4 Nearest low-rank correlation matrix

The problem of computing the nearest low-rank correlation matrix has the form of (26) with $f(X) := \frac{1}{2}\|X - C\|^2$ and $\mathcal{A}(X) := \mathrm{diag}(X) - e$, where C is a given matrix in $\mathbb{S}^n$, and $e \in \mathbb{R}^n$ is the vector of all ones. That is, this problem is described as

$$\begin{array}{ll} \min\limits_{X \in \mathbb{S}^n} & \dfrac{1}{2}\|X - C\|^2 \\ \text{s.t.} & \mathrm{diag}(X) = e, \\ & \mathrm{rank}(X) \le r, \\ & X \succeq 0. \end{array} \tag{40}$$

The nearest low-rank correlation matrix has an important application in finance (see, for example, [28, 29, 24, 14, 35, 37]). There have proposed several solution methods which include the Lagrangian method [37], the geometric optimization methods [14], and the majorization method [24]. These methods are all designed by using the Gramian representation $X = RR^T$ of $X \succeq 0$ where $R \in \mathbb{R}^{n \times r}$. Recently, by using that

$$X \succeq 0,\ \mathrm{rank}(X) \le r \iff \mathrm{tr}(X) = \sum_{i=1}^{r} \lambda_i(X),$$

Li and Qi [17] propose a sequential semismooth Newton method, and Gao and Sun [31] develop a majorized penalty approach. To our best of knowledge, the two methods are the most effective which can deal with problems of $n \ge 1000$. In this section, we will apply the proposed convex relaxation method for solving this class of problems.

4.1 Convex relaxation method for (40)

When applying Algorithm 3.1 for (40), the subproblem (28) in the kth iteration becomes

$$\begin{array}{ll} \min\limits_{X \in \mathbb{S}^n} & \dfrac{1}{2}\|X - C\|^2 \\ \text{s.t.} & \mathrm{diag}(X) = e, \\ & \langle \Psi(\varepsilon_k, X^k),\ X \rangle = r, \\ & X \succeq 0, \end{array} \tag{41}$$

where $\Psi = \Psi_1$ or Ψ_2. An elementary computation yields the dual of (41) has the form

$$\min_{y \in \mathbb{R}^{n+1}} \ \Theta_k(y) \tag{42}$$

with

$$\Theta_k(y) := \frac{1}{2}\left\|[C + \mathcal{B}_k^*(y)]_+\right\|^2 - y^T \begin{pmatrix} e \\ r \end{pmatrix}, \tag{43}$$

where the linear operator $\mathcal{B}_k\colon \mathbb{S}^n \to \mathbb{R}^{n+1}$ and its adjoint $\mathcal{B}_k^*\colon \mathbb{R}^{n+1} \to \mathbb{S}^n$ are defined by

$$\mathcal{B}_k(X) := \begin{bmatrix} \mathrm{diag}(X) \\ \langle \Psi(\varepsilon_k, X^k),\ X\rangle \end{bmatrix} \quad \forall X \in \mathbb{S}^n$$

and

$$\mathcal{B}_k^*(y) := \mathrm{Diag}(y_1, \ldots, y_n) + y_{n+1}\Psi(\varepsilon_k, X^k) \quad \forall y \in \mathbb{R}^{n+1},$$

respectively, and y_i for $i = 1, 2, \ldots, n+1$ denotes the ith component of the vector y.

Compared with the primal problem (41), the dual (42) has much less variables. Also, by [17, Proposition4.1], its objective function $\Theta_k(y)$ has the following properties.

Lemma 4.1 *For each k, let $\Theta_k\colon \mathbb{R}^{n+1} \to \mathbb{R}$ be the function defined by (43). Then,*

(a) *Θ_k is convex and continuously differentiable everywhere in $\mathbb{R}^n \times \mathbb{R}$, and*

$$\nabla\Theta_k(y) = \mathcal{B}_k\left([C + \mathcal{B}_k^*(y)]_+\right) - \begin{pmatrix} e \\ r \end{pmatrix}.$$

(b) *The gradient mapping $\nabla\Theta_k : \mathbb{R}^{n+1} \to \mathbb{R}^{n+1}$ is strongly semismooth everywhere.*

(c) *Θ_k is coercive, i.e., $\Theta_k(y) \to +\infty$ as $\|y\| \to +\infty$.*

(d) *$\overline{y} \in \mathbb{R}^{n+1}$ is an optimal solution of (42) if and only if $\nabla\Theta_k(\overline{y}) = 0$. Furthermore,*

$$\overline{X} := [C + \mathcal{B}_k^*(\overline{y})]_+ \tag{44}$$

is the unique optimal solution of (41).

Lemma 4.1 and the good feature of dual problem (42) motivate us to get a solution of (41) by solving its dual (42). Like [17], we achieve this by finding a root to

$$\nabla\Theta_k(y) = 0, \tag{45}$$

with the globally convergent semismooth Newton method developed by Qi and Sun [30]. For other effective methods for (42), we refer to [21, 2, 33]. Since the Clarke generalized Jacobian $\partial(\nabla\Theta_k)$ of $\nabla\Theta_k$ is difficult to characterize, the following alternative

$$\widehat{\partial}^2\Theta_k(y) := \mathcal{B}_k\partial\Pi_{\mathbb{S}^n_+}(C + \mathcal{B}_k^*(y))\mathcal{B}_k^* \tag{46}$$

is adopted in the method, where $\partial\Pi_{\mathbb{S}^n_+}(C + \mathcal{B}_k^*(y))$ is the Clarke generalized Jacobian of $\Pi_{\mathbb{S}^n_+}(\cdot)$ at $C + \mathcal{B}_k^*(y)$. In the following, we describe the semismooth Newton method [30].

Algorithm 4.1 *(Semismooth Newton Algorithm for (45))*

(S.0) *Choose $y^0 \in \mathbb{R}^{n+1}$, and constants $\eta, \varrho, \delta, \kappa_1 \in (0,1)$ and $\kappa_2, \kappa_3 \in [1, \infty)$. Set $j := 0$.*

(S.1) *Select an element* $\mathcal{V}_j \in \widehat{\partial}^2\Theta_k(y^j)$, *compute* $t_j := \min\{\kappa_1, \kappa_2\|\nabla\Theta_k(y^j)\|\}$, *and apply the conjugate gradient method [16] for the linear system*

$$(\mathcal{V}_j + t_j\mathcal{I})\triangle y^j = -\nabla\Theta_k(y^j) \tag{47}$$

to find a search direction $\triangle y^j$ *such that*

$$\|\nabla\Theta_k(y^j) + (\mathcal{V}_j + t_j\mathcal{I})\triangle y^j\| \le \eta_j,$$

where $\eta_j := \min\{\eta, \kappa_3\|\nabla\Theta_k(y^j)\|\}$.

(S.2) *Let* l_j *be the smallest nonnegative integer* l *such that*

$$\Theta_k(y^j + \delta^l(\triangle y^j)) - \Theta_k(y^j) \le \varrho\delta^l\langle\nabla\Theta_k(y^j), \triangle y^j\rangle. \tag{48}$$

Set $\tau_j := \delta^{l_j}$ *and* $y^{j+1} := y^j + \tau_j\triangle y^j$.

(S.3) *Set* $j \leftarrow j+1$, *and go to Step (S.1).*

Remark 4.1 *(i) From [8] we have* $\partial(\nabla\Theta_k)(y)d \subseteq \widehat{\partial}^2\Theta_k(y)d$ *for any* $d \in \mathbb{R}^{n+1}$, *which means that if every element of* $\widehat{\partial}^2\Theta_k(y)$ *is positive definite, so is every element in* $\partial(\nabla\Theta_k)$.

(ii) Proposition of [17] shows that every element of $\widehat{\partial}^2\Theta_k$ *at an optimal solution of (42) is positive definite. However, this can not guarantee the positive definiteness of* $\widehat{\partial}^2\Theta_k$ *at the initial iterations unless the starting point is chosen close to the set of solutions. Therefore, in (S.1) we solve system (47), instead of system* $\mathcal{V}_j\triangle y^j = -\nabla\Theta_k(y^j)$, *so that the conjugate gradient method [16] may succeed.*

For the locally quadratic convergence analysis of Algorithm 4.1, please refer to [30, Theorem 5.3] or [17, Prop. 4.4]. Next we take a look at the computation of the elements in $\widehat{\partial}^2\Theta_k(y)$. For a given $y \in \mathbb{R}^{n+1}$, let $C + \mathcal{B}_k^*(y) = P\mathrm{Diag}(\lambda_1(y), \ldots, \lambda_n(y))P^T$ be the spectral decomposition of the symmetric matrix $C+\mathcal{B}_k^*(y)$, where P is an $n\times n$ orthogoal matrix, and $\lambda_1(y) \ge \lambda_2(y) \ge \cdots \ge \lambda_n(y)$ are the eigenvalues of $C + \mathcal{B}_k^*(y)$ arranged in nonincreasing order. Associated with $\lambda_i(y)$, we define three disjoint index sets

$$\alpha := \{i \mid \lambda_i(y) > 0\}, \quad \beta := \{i \mid \lambda_i(y) = 0\}, \quad \gamma := \{i \mid \lambda_i(y) < 0\},$$

and write $\bar{\alpha} := \{1, \ldots, n\}\backslash\alpha = \beta \cup \gamma$. Define the linear operator $\mathcal{W} : \mathbb{S}^n \to \mathbb{S}^n$ by

$$\mathcal{W}(H) := P\left(\Xi \circ (P^THP)\right)P^T \quad \forall H \in \mathbb{S}^n, \tag{49}$$

where

$$\Xi := \begin{bmatrix} E_{\alpha\alpha} & W_{\alpha\bar{\alpha}} \\ W_{\alpha\bar{\alpha}}^T & 0 \end{bmatrix}$$

with $E_{\alpha\alpha} \in \mathbb{S}^{|\alpha|}$ being the matrix of ones, $W_{ij} = \frac{\lambda_i(y)}{\lambda_i(y)-\lambda_j(y)}$ for all $i \in \alpha$ and $j \in \bar{\alpha}$. From [25, Lemma 11] it follows that $\mathcal{W} \in \partial\Pi_{\mathbb{S}^n_+}(C + \mathcal{B}_k^*(y))$. Define $\mathcal{V} : \mathbb{R}^{n+1} \to \mathbb{S}^n$ by

$$\mathcal{V}(h) := \mathcal{B}_k\mathcal{W}(\mathcal{B}_k^*(h)) \quad \forall h \in \mathbb{R}^{n+1}. \tag{50}$$

Then, together with equation (46), it follows that $\mathcal{V} \in \widehat{\partial}^2\Theta_k(y)$.

To sum up, our convex relaxation method for (40) works as follows: Algorithm 3.1 is used to solve (40) with each subproblem (41) being solved by applying Algorithm 4.1 to its dual problem (42) for an output $\overline{y}$. The next iterate X^{k+1} is taken to be $\overline{X}$ in (44).

4.2 Numerical experiments

4.2.1 Termination criteria

In the numerical implementation, we use the following stopping rule for Algorithm 3.1

$$|\sum_{i=1}^{r}\lambda_i^k - n| \le \tau_1,\ |\langle \Psi(\varepsilon_k, X^k) - \Psi(\varepsilon_{k-1}, X^{k-1}), X^k\rangle| \le \tau_2,\ \mathrm{rank}(X^k) - r \le \tau_3 \quad (51)$$

where λ_i^k is the ith largest eigenvalue of X^k, $\tau_1 > 0, \tau_2 > 0$ and $\tau_3 > 0$ are given tolerances to control the rank of the iterate X^k, and adopt the following rule for Algorithm 4.1:

$$\|\nabla\Theta_k(\overline{y})\| \le \epsilon \ \text{ and } \ j \le k_{\max} \quad (52)$$

where $\overline{y}$ is the final iterate given by Algorithm 4.1 for the dual of the kth subproblem, $\epsilon > 0$ is a given tolerance, and $k_{\max}$ is the maximum number of iteration.

The first stopping rule in (56) implies that the dual problem (42), and so the primal problem (41), is solved approximately. Indeed, let $\overline{X} \succeq 0$ be defined by (44). Then

$$\|\nabla\Theta_k(\overline{y})\| \le \epsilon \iff \|\mathrm{diag}(\overline{X}) - e\| \le \epsilon \text{ and } |\langle \Psi(\varepsilon_k, X^k), \overline{X}\rangle - r| \le \epsilon. \quad (53)$$

Let

$$\overline{S} := \overline{X} - \left[C + \mathrm{Diag}(\overline{y}_1, \ldots, \overline{y}_n) + \overline{y}_{n+1}\Psi(\varepsilon_k, X^k)\right] = \overline{X} - \left[C + \mathcal{B}_k^*(\overline{y})\right]. \quad (54)$$

By the definition of $\overline{X}$, it is easy to verify that $\overline{S} \succeq 0$ and $\langle \overline{X}, \overline{S}\rangle = 0$. This, along with (53) and (54), implies that $\overline{w} := (\overline{X}, \overline{y}, \overline{S})$ is an approximate KKT point of (41) satisfying $\|R_p(\overline{w})\| + \|R_d(\overline{w})\| + \|R_2(\overline{w})\| = O(\epsilon)$. This shows that under this case Algorithm 4.1 provides an approximate solution to (41) as required in (S.2) of Algorithm 3.1.

4.2.2 Diagonal preconditioner for conjugate gradient

To speed up the convergence of the conjugate gradient solver for (47), we shall introduce a diagonal preconditioner D_j at the iterate y^j. Define the vector $d \in \mathbb{R}^{n+1}$ as the diagonal part of the matrix $\mathcal{V}_j \in \widehat{\partial}^2\Theta_k(y^j)$, i.e., $d := \mathrm{diag}(\mathcal{V}_j)$. Using equations (50) and (53) and the definition of $\mathcal{B}_k$, we have that $d_l,\ l = 1, 2, \ldots, n$ and d_{n+1} are given by

$$\begin{aligned}
d_l &= [\mathrm{diag}(\mathcal{V}_j)]_{ll} = [\mathcal{V}_j(e_l)]_l = \langle e_l, \mathcal{B}_k\mathcal{W}(\mathcal{B}_k^*(e_l))\rangle = \langle \mathcal{B}_k^*(e_l), \mathcal{W}(\mathcal{B}_k^*(e_l))\rangle \\
&= \langle \mathrm{Diag}(\widetilde{e}_l), \mathcal{W}(\mathrm{Diag}(\widetilde{e}_l))\rangle = \langle P^T\mathrm{Diag}(\widetilde{e}_l)P, \Xi \circ (P^T\mathrm{Diag}(\widetilde{e}_l)P)\rangle, \\
d_{n+1} &= \langle e_{n+1}, \mathcal{B}_k\left(\mathcal{W}(\mathcal{B}_k^*(e_{n+1}))\right)\rangle = \langle \mathcal{B}_k^*(e_{n+1}), \mathcal{W}(\mathcal{B}_k^*(e_{n+1})\rangle \\
&= \left\langle \Psi_k(\varepsilon_k, X^k),\ P\left[\Xi \circ (P^T\Psi_k(\varepsilon_k, X^k)P)P^T\right]\right\rangle \\
&= \left\langle P^T\Psi_k(\varepsilon_k, X^k)P,\ \Xi \circ (P^T\Psi_k(\varepsilon_k, X^k)P)\right\rangle,
\end{aligned}$$

where e_l is the lth column of the $(n+1)\times(n+1)$ identity matrix, and $\widetilde{e}_l$ is the lth column of the $n \times n$ identity matrix. Take $D_j = \mathrm{Diag}(d)$. We see that such a preconditioner is different from the one used in [17] for the last diagonal entry of D_j.

4.2.3 Test problems

We apply our method for solving some test problems that are widely used in the literature (see, e.g., [24, 28, 3, 17, 31]). The matrix $C \in \mathbb{S}^n$ of these test problems are given below.

(P1) [3] Set $G_{ij} = 0.5+0.5\exp(-0.05|i-j|)$ for all i, j, and $C = G+G'-\mathrm{Diag}(\mathrm{diag}(G))$.

(P2) [3] Set $G_{ij} = \exp(-|i-j|)$ for all i, j, and let $C = G + G' - \mathrm{Diag}(\mathrm{diag}(G))$.

(P3) [37] Set $G_{ij} = \mathrm{LongCorr}+(1-\mathrm{LongCorr})\exp(\kappa|t_i-t_j|)$ with $\kappa = d_1-d_2\max(t_i, t_j)$, where $\mathrm{LongCorr} = 0.3$, $d_1 = -0.12$, $d_2 = 0.005$, and $t_i = i/2$, and then let $C = G + G' - \mathrm{Diag}(\mathrm{diag}(G))$.

(P4) [28] C_{ij} has the same form as in (P3) with $\mathrm{LongCorr} = 0.6$, $\kappa = -0.1$ and $t_i = i$.

(P5) [15] $C = E + 0.1R$, where E is generated by Matlab's gallery('randcorr',n), and R is a random $n \times n$ symmetric matrix with $R_{ij} \in [-1, 1]$ for $i, j = 1, 2, \ldots, n$.

(P6) [21] C is a random $n \times n$ symmetric matrix with $C_{ij} \in [-1, 1], C_{ii} = 1$ for $i, j = 1, \ldots, n$, which is generated by: $C = \mathrm{rand}(n)$; C=(C+C')/2; $C = 1 - 2 * C$; $C = C - \mathrm{diag}(\mathrm{diag}(C)) + \mathrm{eye}(n)$.

(P7) [38] $C = E+0.05R$ where E and R are generated by: d=10.^(4*[−1 : 1/(n−1) : 0]); E=gallery('randcorr', $n * d/\mathrm{sum(d)}$); R =$2 * \mathrm{rand}(n) - 1$; R=triu(R)+triu(R,1)'; G = $E + 0.05 * R$.

(P8) C is generated by C=2.0∗rand(n,n)-ones(n,n); C=triu(C)+triu(C,1)'; C(i,i)=1 for $i = 1, 2, \ldots, n$.

(P9) [30] $C = 0.9E + 0.1R$ where E and R are given by: d=10.^(4 * [−1 : 1/(n −1) : 0]); E=gallery('randcorr', $n * d/\mathrm{sum(d)}$); R = $2 * \mathrm{rand}(n) - 1$; R=triu(R)+triu(R,1)'; R=(R+R')/2.

4.2.4 Numerical results

We write the code for Algorithm 3.1 in Matlab 7.5, and run it on a desktop (Pentium(R) 4, CPU 2.8GHz and RAM 480Mb). During the testing, we chose $\|\Phi(\varepsilon, X)\|_*$ defined with ϕ_1 and ψ_2 as the approximation of $\mathrm{rank}(X)$, for which Ψ_1 and Ψ_2 in (27) has the form

$$\Psi_1(\varepsilon, X) := \frac{1}{X + \varepsilon} \quad \text{and} \quad \Psi_2(\varepsilon, X) := \frac{1}{\sqrt{X^2 + \varepsilon^2}}.$$

Let $\widehat{X}^0$ be the solution of (40) without $\mathrm{rank}(X) \le r$ yielded by CorNewton.m [30]. Our starting point X^0 was chosen as the output given by applying the modified principal

component analysis to $\widehat{X}^0$ (see [17]). The initial ε_0 was chosen by the value of n/r and $\zeta_0 = |\sum_{i=1}^r \widehat{\lambda}_i^0 - n|$, where $\widehat{\lambda}_1^0 \geq \cdots \geq \widehat{\lambda}_n^0$ are the eigenvalues of $\widehat{X}^0$. When $\Psi = \Psi_1$,

$$\begin{aligned}
&\text{if } \zeta_0 > 100, \text{ then } \varepsilon_0 = \begin{cases} 10 & \text{if } \frac{n}{r} > 100, \\ 5 & \text{if } \frac{n}{r} \in [25, 100], \\ 1 & \text{if } \frac{n}{r} < 25; \end{cases} \\
&\text{if } 5 \leq \zeta_0 \leq 100, \text{ then } \varepsilon_0 = \begin{cases} 0.5 & \text{if } \frac{n}{r} \leq 8, \\ 1 & \text{if } \frac{n}{r} > 8; \end{cases} \\
&\text{if } \zeta_0 < 5, \quad \text{then } \varepsilon_0 = 0.01;
\end{aligned}$$

and when $\Psi = \Psi_2$, the parameter ε_0 is set as follows:

$$\begin{aligned}
&\text{if } \zeta_0 > 200, \text{ then } \varepsilon_0 = \begin{cases} 10 & \text{if } \frac{n}{r} \geq 50, \\ 5 & \text{if } \frac{n}{r} \in (15, 50), \\ 3 & \text{if } \frac{n}{r} \leq 15; \end{cases} \\
&\text{if } 5 \leq \zeta_0 \leq 200, \text{ then } \varepsilon_0 = \begin{cases} 1 & \text{if } \frac{n}{r} \leq 50, \\ 5 & \text{if } \frac{n}{r} > 50; \end{cases} \\
&\text{if } \zeta_0 < 5, \quad \text{then } \varepsilon_0 = 0.1.
\end{aligned}$$

The ratio factor ρ is chosen by the value of $\omega_k \equiv \langle \Psi(\varepsilon_k, X^k) - \Psi(\varepsilon_{k-1}, X^{k-1}), X^k \rangle$, i.e.,

$$\rho = \begin{cases} 0.95 & \text{if } \omega_k \geq 3, \\ 0.9 & \text{if } \omega_k \in [0.03, 3), \\ 0.3 & \text{if } \omega_k < 0.03. \end{cases}$$

The termination parameters τ_1, τ_2 and τ_3 of Algorithm 3.1 were set as: $\tau_1 = 10^{-2}$, $\tau_2 = 3 \times 10^{-2}$, $\tau_3 = 10^{-3}$. The parameters involved in Algorithm 4.1 were set to be

$$\begin{aligned}
&\epsilon = 10^{-3}, \ k_{\max} = 10, \ \kappa_1 = 10^{-5}, \ \kappa_2 = 1, \ \eta = 10, \\
&\kappa_3 = \min(0.5, \max(10^{3-k}, 10^{-2})), \ \varrho = 10^{-4}, \quad \delta = 0.5.
\end{aligned} \tag{55}$$

The maximum number of GG steps at each iteration is set as 150.

We compared our method with the convex relaxation method proposed by Li and Qi [17]. In the code of Li and Qi's method, their Algorithm 3.1 was stopped whenever

$$|\sum_{i=1}^r \lambda_i^k - n| \leq \tau_1 \ \text{ and } \ |\langle U^{k+1} - U^k, X^{k+1} \rangle| \leq \tau_2,$$

where $\lambda_1^k \geq \cdots \geq \lambda_n^k$ are the eigenvalues of iterate X^k, and τ_1 and τ_2 were taken to be

$$\tau_1 = 10^{-3} \times \zeta_0 + 2.0 \times 10^{-4}, \quad \tau_2 = \min(0.03, 10^{-3}\varpi + 0.03)$$

where ζ_0 is defined as above, and $\varpi := |\langle U^1 - U^0, X^1 \rangle|$. The initial point X^0 was chosen in the same way as above, and other parameters in Algorithm 3.1 of [17] were set as $c_0 = 1$, $\mu_0 = 0$, $\rho = \frac{1}{0.79}$. The multiplier μ and the penalty parameter c are modified by

$$\mu_{k+1} := 2\max(0, \mu_k - c_k(\langle X^{k+1}, U^k \rangle)), \quad c_{k+1} := \rho c_k$$

when $|\sum_{i=1}^{r}\lambda_i^k - n| \le 100\tau_1$ or $|\langle U^{k+1}-U^k, X^{k+1}\rangle| \le 100\tau_2$; and otherwise

$$\mu_{k+1} := \max(0, \mu_k - c_k(\langle X^{k+1}, U^k\rangle)), \quad c_{k+1} := \rho c_k.$$

The parameters in Algorithm 4.3 of [17], which is same as Algorithm 4.1, were set as

$$\begin{aligned} &\epsilon = 10^{-8},\ k_{\max} = 100,\ \kappa_1 = 10^{-9},\ \kappa_2 = 0.1,\ \eta = 10,\\ &\kappa_3 = \min(0.5, \max(10^{2-k}, 10^{-4})),\ \varrho = 10^{-4},\quad \delta = 0.5. \end{aligned} \tag{56}$$

During the tests, we applied the modified principal component analysis to the final iterate X^f and $\widehat{X}^f$ generated by our method and Li and Qi's method, respectively, and denote the corresponding output solution by X_r and $\widehat{X}_r$.

To examine the solution quality yielded by our method, we first run Algorithm 3.1 and Li and Qi's algorithm [17], respectively, for all test problems of dimension 100 with r ranging from 1 to 50, and plot the relative difference curve of the objective values generated by the two methods in Figures 6 and 7. In these figures, the red solid curve with a circle denotes the relative difference between the objective values given by Algorithm 3.1 with $\Psi = \Psi_1$ and those of [17], and the blue solid curve with a star denotes the relative difference between the objective values given by Algorithm 3.1 with $\Psi = \Psi_2$ and those of [17]. The relative difference, denoted by **relgap**, is defined by

$$\text{relgap} := \frac{\|X_r - C\| - \|\widehat{X}_r - C\|}{\|\widehat{X}_r - C\|}, \tag{57}$$

where X_r and $\widehat{X}_r$ are the output solutions of our method and that of [17], respectively.

From Figure 6-7, we see that most of the objective values yielded by Algorithm 3.1 with $\Psi = \Psi_2$ have positive relative difference from those given by Li and Qi's method, which are within 5%, and there are 5 problems that have positive relative difference in the range of $[10\%, 15\%]$; the objective values yielded by Algorithm 3.1 with $\Psi = \Psi_1$ are very close to those given by Li and Qi's method, and the largest positive relative difference is within 3.5%. This shows that Algorithm 3.1 with $\Psi = \Psi_1$ yields better objective values which are comparable to those given by Li and Qi's method.

We also tested the performance of our method for solving those problems of large scale ($n \ge 500$). Tables 1 and 2 report the numerical results yielded by Algorithm 3.1 with $\Psi = \Psi_1$ and $\Psi = \Psi_2$, and those given by Li and Qi's method. In these two tables, the first column provides the information of test problems (for example, "P2n500r10" denotes problem (P2) with $n = 500$ and $r = 10$); the second column reports the results of Algorithm 3.1 with $\Psi = \Psi_1$, the third one lists the results given by Algorithm 3.1 of [17], and the fourth column reports the results of Algorithm 3.1 with $\Psi = \Psi_2$. In each column, **It** denotes the number of iterations, i.e., the number of subproblems solved; **fval** means the output objective value, i.e., the value of $\|X_r - C\|$ for our method, and the

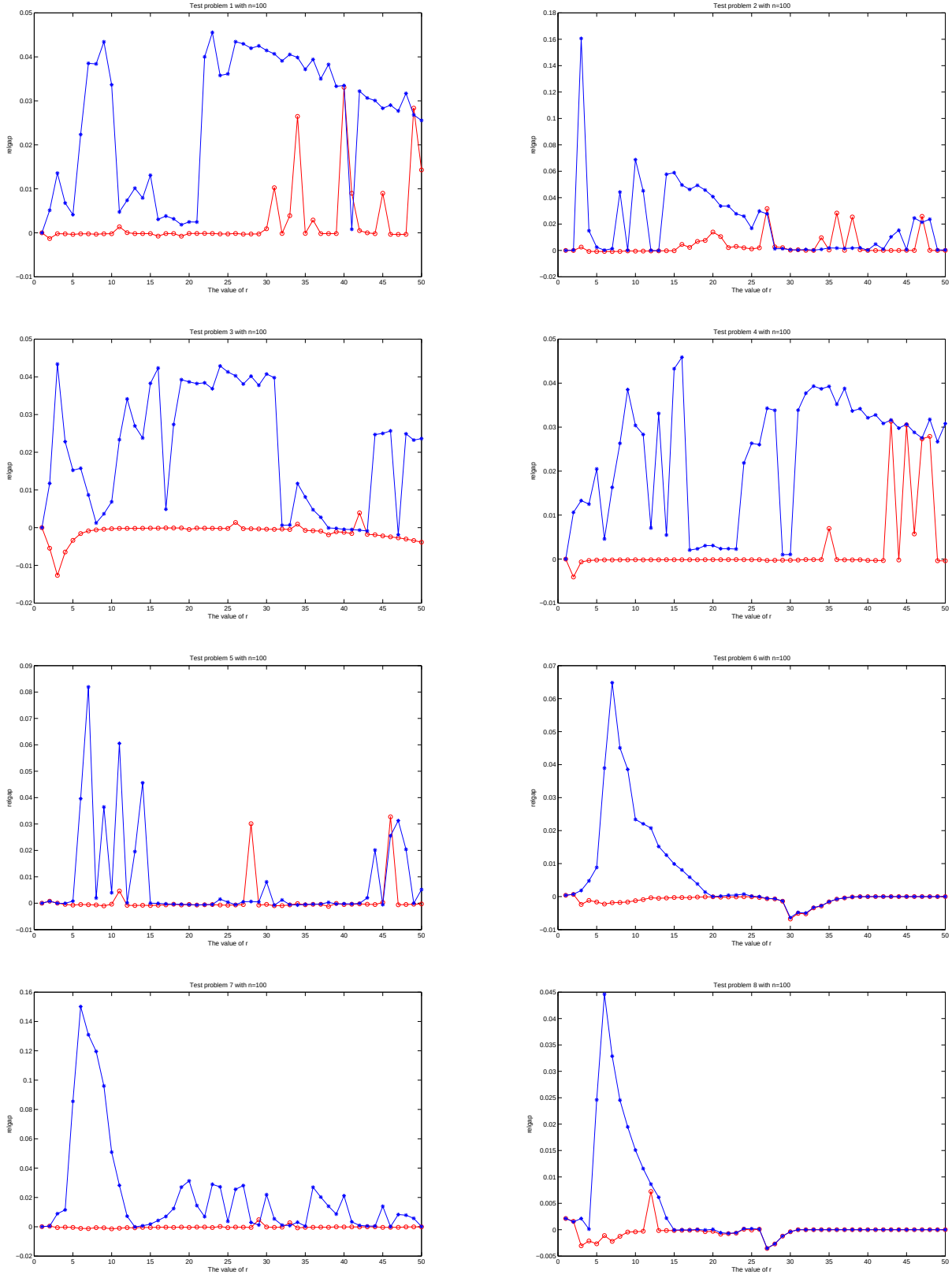

Figure 6: The relative difference between the objective values for P1-P8

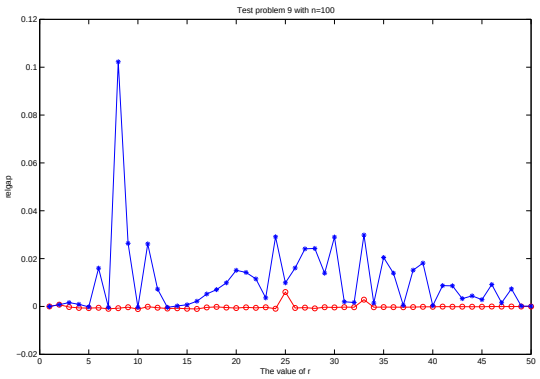

Figure 7: The relative difference between the objective values for P9

value of $\|\widehat{X}_r - C\|$ for Li and Qi's method; and **Relgap** denotes the relative difference of objective values given by our method from those of [17], determined by formula (57).

From Tables 1 and 2 in appendix, we see that the objective values yielded by Algorithm 3.1 with $\Psi = \Psi_1$ are all (except 5 test problems) smaller than those given by Li and Qi's method, and moreover, for problem (P3), the improved rate of objective values (i.e., the value of $|\text{Relgap}| \times 100\%$) is even over 30%; the objective values yielded by Algorithm 3.1 with $\Psi = \Psi_2$ are worse than those given by Algorithm 3.1 with $\Psi = \Psi_1$, but except "P3n500r2" and "P4n500r5" the relative difference from those given by Li and Qi's method within 4%, and for problem (P3), the improved rate of objective values is also over 30%. We also note that the CPU time used by Li and Qi's method is less than that of our method, and the CPU time used by Algorithm 3.1 with $\Psi = \Psi_2$ is less than that of Algorithm 3.1 with $\Psi = \Psi_1$.

We also compared the performance of Algorithm 3.1 with $\Psi = \Psi_2$ with Li and Qi's method by solving problems (P1)-(P9) with $n = 1000$ and six different r for each test instance. Tables 3 and 4 in appendix report the numerical results. We note that for Problems (P1), (P4) and (P8), the objective values of Algorithm 3.1 is worse than those of Li and Qi's method, but the relative difference is within 10%. For the other several problems, the objective values yielded by Algorithm 3.1 is comparable even superior to those given by Li and Qi's method. Particularly, for problem (P3) with five different r, the improved rate of objective values attains 50%. Of course, we should point out that the CPU time used by Algorithm 3.1 is a little more than that of Li and Qi's method.

5 Conclusion

We have established a unified framework to construct the approximation functions of rank$(\cdot)$, and studied the excellent approximation properties of these functions, which

opens a new way to deal with the discontinuous and nonconvex rank function. We also proposed a convex relaxation method with the approximation functions associated to ϕ_p and ψ_p for the rank minimization problem with PSD cone constraint, and showed that each accumulation point of the sequence generated is a stationary point of its equivalent problem. The method was employed to compute the nearest low-rank correlation matrix, and numerical comparison with Li and Qi's [17] shows that it tends to generate a better local optimal solution although a little more time is required. This further demonstrates the feasibility to deal with the nonconvexity of rank function via these functions.

There are several research topics that are worthwhile to pay attention to:

- To apply the convex relaxation method proposed in Section 3 for solving other types of rank minimization problems, and improve its numerical efficiency further.
- To study convergence and numerical performance of the convex relaxation method mentioned in the end of Section 3 for the problem (2) with the PSD cone constraint.
- To explore other effective convex relaxation methods by making use of the special structure of some approximation functions.

Acknowledgements. We would like to thank Dr. Li Qingna for sending us their code, which gives us a great help for the numerical experiments of this paper.

References

[1] R. Bhatia, *Matrix Analysis*, Springer-Verlag, New York, 1997.

[2] S. Boyd and L. Xiao, *Least-squares covariance matrix adjustment*, SIAM Journal on Matrix Analysis and Applications, vol. 27, pp. 532-546, 2005.

[3] D. Brigo, *A note on correlation and rank reduction*, 2002 (www.damianobrigo.it).

[4] S. Burer, R. D. C. Monteiro and Y. Zhang, *Maximum stable set formulations and heuristics based on continuous optimization*, Mathematical Programming, vol. 94, pp. 137-166, 2002.

[5] C. Chen and O. L. Mangasarian, *A class of smoothing functions for nonlinear and mixed complementarity problems*, Computational Optimization and Applications, vol. 5, pp. 97-138, 1996.

[6] E. J. Candès and B. Recht, *Exact matrix completion via convex optimization*, Foundations of Computational Mathematics, vol. 9, pp. 717-772, 2009.

[7] J.-F. Cai, E. J. Candès and Z. Shen, *A singular value thresholding algorithm for matrix completion*, Technical report, 2008.

[8] F. H. Clarke, *Optimization and Nonsmooth Analysis*, John Wiley & Sons, New York, 1983 (reprinted by SIAM, Philadelphia, PA, 1990).

[9] D. Ghao, D. F. Sun and K-C. Toh, *An introduction to a class of matrix cone programming*, technical report, 2010.

[10] M. Fazel, *Matrix Rank Minimization with Applications*, PhD thesis, Stanford University, 2002.

[11] M. Fazel, H. Hindi and S. Boyd, *Log-det heuirstic for matrix rank minimization with applicatios to Hankel and Euclidean distance matrices*, ACC03-IEEE06884, 2003.

[12] M. Fazel, H. Hindi and S. Boyd, *Rank minimization and applications in system theory, in Proceeding of the American Control Conference*, vol. 4, pp. 3273-3278, 2004.

[13] M. X. Goemans and D. P. Williamson, 0.878-*approximation algorithms for MAX CUT and MAX 2SAT*, Lecture Notes Computer Science, pp. 422-431, 1994.

[14] I. Grubiŝić and R. Pietersz, *Efficient rank reduction of correlation matrices*, Linear Algebra and Application, vol. 422, pp. 629-653, 2007.

[15] N. J. Higham, *Computing the nearest correlation matrix-a problem from finance*, IMA Journal on Numerical Analysis, vol. 22, pp. 329-343, 2002.

[16] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, Journal of Research of the National Bureau of Standards, vol. 49, pp. 409-436, 1952.

[17] Q. N. Li and H. D. Qi, *A sequential semismooth Newton method for the nearest low-rank correlation matrix problem*, technical report, University of Southampton, September, 2009.

[18] Z. Liu and L. Vandenberghe, *Interior-point method for nuclear norm approximation with application to system identification*, submitted to Mathematical Programming Series, 2008.

[19] Y.-J. Liu, D. F. Sun and K. C. Toh, *An implementable proximal point algorithmic framework for nuclear norm minimization*, 2009. Preprint availabe at http://www. optimization-online.org.

[20] M. Mesbahi, *On the rank minimization problem and its control applications*, Systems & Control Letters, vol. 33, pp. 31-36, 1998.

[21] J. Malick, *A dual approach to semidefinite least-squares problems*, SIAM Journal on Matrix Analysis and Applications, vol. 26, pp. 272-284, 2004.

[22] S. Ma, D. Goldfarb and L. Chen, *Fixed point and Bregman iterative methods for matrix rank minimization*, Mathematical Programming, to appear 2009.

[23] J. Malick, J. Povh, F. Rendl and A. Wiegele, *Regularization methods for semidefinite programming*, SIAM Journal on Optimization, vol. 20, pp. 336-356, 2009.

[24] R. Pietersz and I. Grubiŝić, *Rank reduction of correlation matrices by majorization*, Quant. Financ., vol. 4, pp. 649-662, 2004.

[25] J.-S. Pang, D. F. Sun and J. Sun, *Semismooth homemorphisms and strong stability of semidefinite and Lorentz complementarity problems*, Mathematics of Operations Research, vol. 28, pp. 39-63, 2003.

[26] B. Recht, M. Fazel and P. Parrilo, *Guaranteed minimum rank solutions of matrix equations via nuclear norm minimization*, SIAM Review, vol. 52, pp. 471-501, 2010.

[27] B. Recht, W. Xu and B. Hassibi, *Null space conditions and thresholds for rank minimization*, Technical report, California Institute of Technology, April 2009.

[28] R. Rebonato, *Modern Princing and Interest-Rate Derivatives*, Princeton University Press, New Jersey, 2002.

[29] R. Rebonato, *Interest-rate term-structure pricing models: a review*, Proc. Royal Soc. London Ser. A, vol. 460, pp. 667-728, 2004.

[30] H. D. Qi and D. F. Sun, *A quadratically convergent Newton method for computing the nearest correlation matrix*, SIAM Journal on Matrix Analysis and Applications, vol. 28, pp. 360-385, 2006.

[31] Y. Gao and D. F. Sun, *A majorized penalty approach for calibrating rank constrained correlation matrix problems*, technical report, 2010.

[32] D. F. Sun and J. Sun, *Strong semismoothness of eigenvalues of symmetric matrices and its application to inverse eigenvalue problems*, SIAM Journal on Numerical Analysis, vol. 40, pp. 2352-2367, 2003.

[33] K. C. Toh, *An inexact path-following algorithm for convex quadratic SDP*, Mathematical Programming, vol. 112, pp. 221-254, 2008.

[34] K. C. Toh and S. Yun, *An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems*, Accepted in Pacific Journal of Optimization.

[35] L. Wu, *Fast at-the-money calibration of the LIBOR market model using Lagrangian multipliers*, J. Comput. Finance, vol. 6, pp. 39-77, 2003.

[36] Z. Wen, W. Yin and Y. Zhang, *Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm*, TR10-07, Rice University, March 2010.

[37] Z. Zhang and L. Wu, *Optimal low-rank approximation to a correlation matrix*, Linear Algebra and Applications, vol. 364, pp. 161-187, 2003.

[38] X. Y. Zhao, D. F. Sun and K. C. Toh, *A Newton-CG augmented Lagrangian method for semidefinite programming*, SIAM Journal on Optimization, vol. 20, pp. 1737-1765, 2010.

[39] Y. B. Zhao, *Approximation theory of matrix rank minimization and its application to quadratic equations*, Optimization online, 2010.

Appendix

Table 1: Numerical comparisons of Algorithm 3.1 with that of [17]

	Algorithm 3.1 with Ψ_1				Li and Qi's method			Algorithm 3.1 with Ψ_2			
Problem	**It**	**fval**	**Relgap**	**t(s)**	**It**	**fval**	**t(s)**	**It**	**fval**	**Relgap**	**t(s)**
P1n500r2	6	156.9509	-6.60e-3	48.5	4	157.9984	15.4	2	163.3123	3.36e-2	12.1
P1n500r5	9	78.8360	-7.17e-4	90.8	4	78.8926	14.6	4	80.3976	1.91e-2	20.1
P1n500r10	24	38.6842	-9.74e-5	132.9	4	38.6880	13.6	3	39.7885	2.84e-2	15.0
P1n500r20	26	15.7074	-4.49e-5	89.2	4	15.7081	14.7	8	15.7312	1.50e-3	23.6
P1n500r35	35	7.0188	-5.74e-5	91.5	4	7.0192	12.2	11	7.1938	2.49e-2	28.6
P1n500r50	50	4.1392	-4.93e-5	110.6	5	4.1394	14.5	28	4.1420	6.41e-4	70.8
P2n500r2	10	351.4199	-2.85e-6	134.4	4	351.4209	12.9	3	351.4270	1.73e-5	18.7
P2n500r5	8	220.2287	-1.19e-3	56.7	4	220.4906	14.1	3	220.5427	2.36e-4	18.2
P2n500r10	14	153.2989	-1.24e-3	78.1	4	153.4895	14.6	7	153.3513	-9.01e-4	88.0
P2n500r20	33	104.9269	-6.62e-4	266.4	4	104.9964	11.7	10	104.9231	-6.98e-4	101.1
P2n500r35	34	75.3059	-2.19e-4	274.7	4	75.3224	12.2	13	75.2984	-3.19e-4	117.8
P2n500r50	40	59.6496	-1.53e-5	273.8	4	59.6487	13.0	19	59.6387	-1.67e-4	139.8
P3n500r2	12	**253.0254**	-1.13e-2	47.5	7	255.9157	30.6	4	304.7932	1.91e-1	22.2
P3n500r5	27	**159.0494**	-1.39e-2	95.0	8	184.6605	42.8	21	**164.6878**	-1.08e-1	131.7
P3n500r10	28	**99.0853**	-3.18e-1	119.4	8	145.2971	50.3	22	**103.9178**	-2.85e-1	109.7
P3n500r20	42	**62.3163**	-3.43e-1	163.5	8	94.8779	51.4	20	**63.7044**	-3.29e-1	91.3
P3n500r35	38	**39.8079**	-2.75e-1	160.7	6	54.9002	38.4	22	**39.8432**	-2.74e-1	97.3
P3n500r50	44	**28.1759**	-5.11e-2	154.4	5	29.6935	25.8	17	**28.8144**	-2.96e-2	58.8
P4n500r2	7	133.6817	-1.11e-2	36.8	4	135.1773	16.4	3	140.0501	3.61e-2	19.4
P4n500r5	10	75.7594	-3.23e-3	85.1	4	76.0048	16.4	12	83.9201	1.04e-1	49.5
P4n500r10	23	44.3130	-4.94e-4	160.4	4	44.3349	14.4	2	45.3653	2.32e-2	12.9
P4n500r20	30	21.66740	-4.11e-6	148.4	4	21.6673	12.9	8	21.8199	7.04e-3	24.6
P4n500r35	31	10.5812	-1.62e-5	103.6	4	10.5814	13.2	8	10.6396	5.50e-3	31.6
P4n500r50	50	6.4209	-3.07e-5	139.8	5	6.4210	15.7	16	6.5737	2.38e-2	47.5

Table 2: Table 1 (continue)

	Algorithm 3.1 with Ψ_1				Li and Qi's method			Algorithm 3.1 with Ψ_2			
Problem	**Iter**	**fval**	**Relgap**	**t(s)**	**Iter**	**fval**	**t(s)**	**Iter**	**fval**	**Relgap**	**t(s)**
P5n500r2	10	351.3317	-3.82e-5	127.7	2	351.3451	10.1	3	347.5722	-4.57e-6	19.2
P5n500r5	9	219.5859	-7.34e-5	99.4	3	219.6021	11.5	4	212.4129	3.29e-5	17.5
P5n500r10	14	152.2052	-1.72e-4	113.3	4	152.2314	15.4	8	141.6951	-9.62e-5	16.1
P5n500r20	25	103.3847	-3.34e-4	181.2	4	103.4192	13.6	10	89.6403	-2.25e-4	19.7
P5n500r35	24	73.5938	-1.06e-4	201.3	4	73.6016	13.7	13	58.2648	-2.72e-4	13.2
P5n500r50	43	58.5827	1.26e-2	192.7	4	57.8547	13.6	19	43.0489	-2.57e-4	16.7
P6n500r2	6	389.5932	-1.79e-4	57.3	3	389.6628	22.3	5	389.7384	1.94e-4	40.0
P6n500r5	12	273.5679	-1.24e-3	67.8	4	273.9072	21.2	5	273.8473	-2.19e-4	51.4
P6n500r10	14	222.3986	-7.17e-4	95.8	4	222.5581	18.2	9	222.4688	-4.01e-4	110.2
P6n500r20	17	193.2049	-3.97e-4	143.3	4	193.2817	17.7	17	193.8198	-2.78e-3	98.1
P6n500r35	26	180.5804	-5.90e-5	153.4	4	180.5910	19.1	7	180.5862	-2.68e-5	45.6
P6n500r50	31	176.3348	-5.08e-5	183.3	5	176.3438	21.7	15	176.4469	5.85e-4	67.5
P7n500r2	8	346.3982	-4.94e-5	112.1	4	346.4153	21.8	3	346.4248	2.74e-5	35.0
P7n500r5	9	210.3147	-1.69e-4	83.7	4	210.3502	18.3	6	211.8015	6.90e-3	122.4
P7n500r10	13	138.8576	-3.23e-4	106.7	4	138.9025	15.3	12	141.3051	1.73e-2	97.9
P7n500r20	28	85.4387	-4.14e-4	284.4	4	85.4740	16.3	18	87.4528	2.32e-2	115.9
P7n500r35	27	52.5217	-4.81e-5	148.8	4	52.5192	15.8	19	53.1404	1.18e-2	86.0
P7n500r50	35	36.0001	-9.30e-2	167.5	4	36.0035	16.7	18	36.0040	1.52e-5	86.4
P8n500r2	6	433.1553	1.41e-4	53.2	4	433.0941	23.2	4	433.2293	3.12e-4	38.4
P8n500r5	12	331.1695	-9.57e-4	84.1	5	331.4866	22.6	5	331.6316	4.38e-4	41.4
P8n500r10	14	290.4906	-6.06e-4	108.7	4	290.6668	18.2	14	292.8695	7.58e-3	140.7
P8n500r20	29	269.0100	-2.19e-4	204.6	4	269.0690	21.2	14	269.9173	3.15e-3	84.1
P8n500r35	34	260.7271	-5.47e-5	152.6	5	260.7413	20.8	20	260.9724	8.86e-4	94.1
P8n500r50	41	258.2672	-2.00e-3	139.8	5	258.7853	22.0	15	258.3156	-1.82e-3	64.8
P9n500r2	8	347.5538	-5.31e-5	93.7	3	347.5723	19.2	4	347.5792	1.99e-4	37.3
P9n500r5	9	347.5791	-2.14e-4	86.0	3	212.4129	17.5	4	212.3989	-6.60e-5	38.1
P9n500r10	25	141.6239	-5.03e-4	175.7	4	141.6951	16.1	8	141.6649	-2.13e-4	104.8
P9n500r20	39	89.7684	1.43e-3	262.6	5	89.6403	19.7	6	89.6729	3.64e-4	58.5
P9n500r35	34	58.6687	6.93e-3	134.7	3	58.2648	13.2	13	58.2538	-1.90e-4	96.0
P9n500r50	44	43.7571	1.65e-2	148.8	4	43.0489	16.7	18	43.0454	-8.20e-5	103.3

Table 3: Numerical results of Algorithm 3.1 for (40)

	Algorithm 3.1 with Ψ_1				Li and Qi's method		
Problem	**Iter**	**fval**	**Relgap**	**t(s)**	**Iter**	**fval**	**t(s)**
P1n1000r3	3	287.0060	8.50e-2	113.7	5	264.5318	113.2
P1n1000r13	15	86.8880	3.53e-3	323.7	5	86.5826	101.9
P1n1000r32	8	30.2101	1.15e-2	145.7	4	29.8657	69.9
P1n1000r53	18	14.7734	4.47e-4	457.6	5	14.7668	82.7
P1n1000r68	13	10.3822	9.49e-3	215.5	4	10.2847	66.7
P1n1000r81	27	7.9757	2.80e-3	397.8	4	7.9535	73.1
P2n1000r2	3	705.0029	3.68e-5	241.8	4	704.9769	128.1
P2n1000r14	3	261.8745	-1.13e-4	163.4	5	261.9042	142.7
P2n1000r31	6	171.1048	-3.62e-4	569.1	4	171.1667	118.9
P2n1000r47	16	135.2588	-3.32e-4	890.6	4	135.3037	119.8
P2n1000r65	21	111.4771	-2.39e-4	1130.1	3	111.5037	97.6
P2n1000r79	19	98.6160	-1.25e-4	906.1	6	98.6284	164.3
P3n1000r4	32	**399.5636**	-1.65e-2	1149.6	9	406.2826	272.0
P3n1000r15	30	**177.7606**	-4.59e-1	1040.0	9	328.3421	350.3
P3n1000r33	31	**110.2742**	-5.59e-1	993.7	9	249.9521	429.7
P3n1000r49	26	**84.3557**	-5.89e-1	658.9	8	205.1194	383.2
P3n1000r62	20	**72.5765**	-5.85e-1	686.4	8	174.9802	400.8
P3n1000r84	19	**57.9929**	-5.60e-1	630.3	8	131.8983	408.0
P4n1000r5	22	187.3143	1.30e-1	657.0	6	165.7712	135.9
P4n1000r16	14	75.5865	9.87e-4	460.8	4	75.5120	81.2
P4n1000r34	9	37.0021	6.87e-3	173.7	4	36.7495	71.3
P4n1000r48	14	24.4259	1.41e-4	210.3	4	24.4225	71.3
P4n1000r63	9	17.2064	2.23e-3	229.6	4	17.1682	65.8
P4n1000r87	16	11.1999	1.66e-2	267.1	4	11.0166	68.0
P5n1000r3	3	574.1612	-2.42e-6	116.1	3	574.1626	94.3
P5n1000r12	6	281.2724	6.14e-5	446.4	5	281.2552	94.1
P5n1000r28	7	177.6802	-6.37e-5	634.1	4	177.6915	84.3
P5n1000r41	14	142.6505	-2.09e-4	907.0	4	142.6803	84.8
P5n1000r60	19	113.0447	-2.08e-4	1153.5	4	113.0682	85.7
P5n1000r72	17	100.4523	-1.69e-4	804.9	4	100.4693	103.7

Table 4: Numerical results of Algorithm 3.1 for (40)

	Algorithm 3.1 with Ψ_1				Li and Qi's method		
Problem	**Iter**	**fval**	**Relgap**	**t(s)**	**Iter**	**fval**	**t(s)**
P6n1000r2	5	791.2449	5.21e-4	250.6	3	790.8328	130.1
P6n1000r11	7	459.5145	5.80e-3	623.9	4	456.8654	96.9
P6n1000r27	18	395.4773	3.15e-3	647.8	4	394.2367	103.0
P6n1000r43	15	377.8550	-7.77e-5	795.0	4	377.8843	110.7
P6n1000r69	23	368.5132	-4.04e-5	565.3	5	368.5281	118.0
P6n1000r88	16	365.8449	-8.88e-4	352.7	5	366.1700	118.8
P7n1000r5	6	434.2943	2.45e-4	419.6	4	434.1879	93.0
P7n1000r17	16	222.6896	2.41e-2	731.3	5	217.4522	120.0
P7n1000r35	11	134.4941	-1.75e-4	390.4	4	134.5177	88.1
P7n1000r55	18	94.1228	-7.71e-4	680.9	4	94.1954	90.1
P7n1000r67	16	79.0870	-7.11e-4	645.5	5	79.1433	105.6
P7n1000r95	15	56.3294	1.84e-3	433.4	4	56.2259	95.4
P8n1000r3	4	777.1405	3.19e-4	232.7	4	776.8924	147.1
P8n1000r10	7	605.7385	4.39e-3	685.3	4	603.0904	102.4
P8n1000r25	11	553.0403	2.06e-3	640.8	4	551.9055	124.1
P8n1000r44	21	537.9683	2.18e-4	575.4	6	537.8513	161.1
P8n1000r60	14	533.7283	-5.69e-4	381.0	5	534.0324	122.0
P8n1000r82	15	531.2797	-3.39e-3	394.5	6	533.0890	185.3
P9n1000r4	4	491.2534	1.30e-4	218.7	4	491.1895	118.2
P9n1000r18	8	215.7252	-1.30e-4	786.8	4	215.7533	88.0
P9n1000r37	12	137.7603	-1.69e-4	1104.8	4	137.7835	89.2
P9n1000r42	14	126.4726	-1.58e-4	925.7	4	126.4926	89.3
P9n1000r65	21	93.2817	-1.28e-4	1104.8	4	93.2936	95.9
P9n1000r77	18	82.5107	-1.10e-4	925.7	3	82.5198	78.4