NUMERICAL EXPERIMENTS WITH UNIVERSAL BARRIER FUNCTIONS FOR CONES OF CHEBYSHEV SYSTEMS

L. FAYBUSOVICH[*], T. MOUKTONGLANG[†], T. TSUCHIYA [‡]

Abstract. Based on previous explicit computations of universal barrier functions, we describe numerical experiments for solving certain classes of convex optimization problems. The comparison is given of the performance of the classical affine-scaling algorithm with the similar algorithm built upon the universal barrier function.

Key words. Affine-scaling algorithm, approximations of universal barrier functions

AMS subject classifications. 90C51,90C25,90C24

1. Introduction. By now there is a deep and complete theory of interior-point algorithms for problems involving symmetric cones [1, 4, 13]. The software has been developed for semidefinite and second-order programming problems and some of the packages [14] cover all types of symmetric programming problems. The method of choice is usually a version of primal-dual algorithms. There are several tentative venues for possible extension of the domain of applicability of interior-point algorithms. One possible direction is to move to infinite-dimensional problems involving symmetric cones [6, 7]. This venue shows some promising results. A very popular direction is to develop primal-dual algorithms for nonconvex nonlinear programming problems. (see e.g. [17] and references there in). If we wish to stay within the class of convex optimization problems, one tantalizing tool is the universal barrier function [12]. It seems to open an opportunity to solve more or less arbitrary finite-dimensional convex optimization problem using polynomial-time interior-point algorithms. In practice, however, a lot of obstacles arise. To begin with the universal barrier function is in general described as a complicated multi-dimensional integral. In vast majority of cases it is simply impossible to reduce this function to a "computable" form. Thus, the first important issue is to understand what is the right way to approximately evaluate the universal barrier. This problem is to significant extent equivalent to the following question: how the most popular interior-point algorithms behave if the data necessary to run such algorithms is known only approximately (but with a controlled error). The second issue is related to the fact that it is difficult to create primal-dual algorithms based on a reasonably complicated universal barrier function. As a minimum prerequisite we at least should be able to compute the Legendre dual of a universal barrier which is typically a difficult problem itself. The class of symmetric cones is a notable exception. Here the Legendre dual of the universal barrier essentially coincides with the barrier itself. This predetermines the possibility of constructing efficient primal-dual algorithms. Thus, we need to return to classes of primal (resp. dual) only algorithms for more general situations.

The third issue is that one iteration of an interior-point algorithm based on a

[*]This author was supported in part by National Science Foundation Grant NO. DMS04-02740. Department of Mathematics University of Notre Dame, Notre Dame, IN 46556 USA

[†]This author was supported in part by National Science Foundation Grant NO. DMS04-02740. Chiang Mai University , Mathematics Dapertment, Faculty of Science, Chiang Mai, Thailand

[‡]This author was supported in part by the Grant-in-aid for scientific Research (C) 15510144 of The Ministry of Education, Culture, Sport, Science, and Technology of Japan. The Institute of Statistical Mathematics, 4-6-7 Minami-Azabu, Minato-Ku, Tokyo, 106-8569, Japan.

reasonably complicated universal barrier will be much more complicated than usual one. The theory promises that the number of such iterations maybe significantly lower in comparison with interior-point algorithms based on more traditional approaches. However, this third problem seem to open new computational possibilities. Indeed, a typical optimization algorithm applied to a large-scale problem will run a sizable number of relatively simple iterations. Hence the possibilities of using parallel computations on supercomputer seem to be quite limited. On the other hand, the operations involved in one iteration of an algorithm based on reasonably complicated universal barrier are highly parallelizable and hence, ideal for the use of multiprocessor supercomputers.

In the present paper we address first two issues for a special class of optimization problems in the conic form.

Let $u_0, \cdots, u_n$ be a Chebyshev system (see [3, 5, 10]) of continuous differentiable functions defined on an interval $[a, b]$. Consider the cone

$$\mathcal{K} = \{(x_0, \cdots, x_n)^T \in \mathbf{R}^{n+1} : \sum_{i=0}^{n} x_i u_i(t) \geq 0, \forall t \in [a, b]\}.$$

We will consider problems of the following form:

$$\sum_{i=0}^{n} c_i x_i \to \min,$$

$$Ax = b, x \in \mathcal{K}.$$

Here A is $l \times (n+1)$ matrix, $b \in \mathbf{R}^l$.

The universal barrier function for the cone $\mathcal{K}$ has been computed in [3]. For simplicity, we present it here only the case $n = 2\nu - 1$. The general case is discussed in [3]. Let $x \in \text{int}(\mathcal{K})$. Then the universal barrier $F(x)$ looks like this:

$$F(x) = \frac{1}{2} \ln \det \mathcal{D}(x),$$

where $\mathcal{D}(x)$ is a skew-symmetric matrix of the size $2\nu \times 2\nu$. The $(i, j)-$th entry of this matrix $(i, j = 1, \cdots, 2\nu)$ has the form:

$$\int_a^b \frac{u_i(t)u_j'(t) - u_j(t)u_i'(t)}{p(t)^2}\, dt. \tag{1.1}$$

Here $p(t) = \sum_{i=0}^{n} x_i u_i(t)$. Here $u_j'(t)$ is the derivative of u_j with respect to t.

Notice that the evaluation of the gradient and the Hessian of F requires the computation of a significant number of one-dimensional integrals of type (1.1). Let $a \leq t_1 < t_2 < \cdots < t_m \leq b$ be a finite-subset of the interval $[a, b]$. Consider vectors

$$v_i = \begin{bmatrix} u_0(t_i) \\ \vdots \\ u_n(t_i) \end{bmatrix}, i = 1, 2, \cdots m.$$

The system of vectors $v_i \in \mathbf{R}^{n+1}, i = 1, 2, \cdots m$ form a finite Chebyshev system (a Chebyshev system over a finite set). See [3, 5, 10]. We can introduce the cone $\mathcal{K}_m$

(1.2) $$\mathcal{K}_m = \{x \in \mathbf{R}^{n+1} : \langle v_i, x\rangle \geq 0, i = 1, 2, \cdots m\}.$$

Here $\langle \cdot, \cdot \rangle$ is the standard scalar product in $\mathbf{R}^{n+1}$.
Notice that

$$\mathcal{K}_m = \{x \in \mathbf{R}^{n+1} : \sum_{i=1}^{n+1} x_i u_i(t_j) \geq 0, j = 1, 2, \cdots m\}.$$

It is clear that $\mathcal{K}_m \supset \mathcal{K}$ and by choosing a sufficiently refined grid of points $t_1, \cdots, t_m$, one can hope that $\mathcal{K}_m$ provides a reasonable polyhedral approximation for $\mathcal{K}$. In [5] the universal barriers have been computed for the cones of the type $\mathcal{K}_m$. It has been also shown that corresponding barrier functions approximate the barrier function F and by choosing sufficiently refined grid this approximation can be made with an arbitrary precision. Thus, the described situation seems to be ideal to conduct numerical experiments with approximations of universal barrier functions.

2. Affine-scaling algorithm. Consider the problem

(2.1) $$\langle c, x\rangle \to \min$$

(2.2) $$Ax = b,$$

(2.3) $$x \in \mathcal{K}.$$

Here $\mathcal{K}$ is a closed convex cone in $\mathbf{R}^n$ with nonempty interior which does not contain straight lines. Let F be a self-concordant barrier for $\mathcal{K}$ [12].

Denote by $\mathrm{int}(\mathcal{F})$ the set $\{y \in \mathbf{R}^n : Ay = b, y \in \mathrm{int}(\mathcal{K})\}$.

Let $x \in \mathrm{int}(\mathcal{F})$. Introduce the affine-scaling direction:

$$V(x) = H_F(x)^{-1}(c - A^T\lambda),$$

where $H_F(x)$ is the Hessian of F evaluated at a point x and $\lambda = \lambda(x)$ is chosen in such a way that $AV(x) = 0$, i.e.,$\lambda = \lambda(x)$ is a solution of the following system of linear equations:

(2.4) $$AH_F(x)^{-1}A^T\lambda = H_F(x)^{-1}c.$$

Given $x \in$int$(\mathcal{F}(x))$, we denote by

$$t(x) = \sup\{t > 0 : x - tV(x) \in \mathcal{K}\}.$$

We define an affine-scaling algorithm as an iterative scheme:

$$x_{k+1} = x_k - \alpha(x_k)V(x_k),$$

where $x \in$int$(\mathcal{F}), 0 < \alpha(x_k) < t(x_k)$ is a parameter. It is easy to see that $\langle c, x_k\rangle$ form a monotonically decreasing sequence.

Notice that the dual to the problem (2.1)-(2.3) has the form:

$$\langle \lambda, b \rangle \to \max,$$

$$c - A^T \lambda \in K^*,$$

where K^* is the dual of the cone K. It is expected (analogously to the classical linear programming case) that Lagrange multipliers

$$\lambda(x_k) = (AH_F^{-1}(x_k)A^T)^{-1}AH_F^{-1}(x_k)c$$

converge to the optimal solution of the dual problem when $k \to \infty$ (see numerical experiments below).

Notice that $t(x) \geq \frac{1}{\|V(x)\|_x}$, where $\|V(x)\|_x = \langle H_F(x)V(x), V(x)\rangle^{1/2}$. (see e.g. [12]). The convergence of the affine-scaling algorithm (under the various nondegenercy assumptions and various choices of the step size is understood very well for the case where $\mathcal{K}$ is a polyhedral cone and F is the classical logarithmic barrier (see e.g. [16])).

For other cones, however, the algorithm may not converge to the optimal solution (see e.g. [11]). Even for the linear programming problem the polynomiality of the affine-scaling algorithm is not established. However, we have chosen this algorithm for our numerical experiments because of the following reasons:
1. It is very easy to implement.
2. It does not require the line search.
3. It is possible to directly compare the performance of affine scaling algorithms based on the universal and logarithmic barrier.
4. In principle, it is possible to get an information about the optimal solution of the dual problem.

When we replace the cone $\mathcal{K}$ with $\mathcal{K}_m$, we obtain the following Linear programming problem which approximates the original (semi-infinite programming) problem:

$$\langle c, x \rangle \to \min,$$

$$Ax = b, \quad x \in \mathcal{K}_m.$$

In this paper, we apply (in most of the numerical examples) the affine-scaling algorithm based on the universal barrier function associated with $\mathcal{K}_m$ and compare it with the standard affine-scaling algorithm based on the classical log barrier function. When it is possible, we solve the original semi-infinite programming problem using semidefinite programming.

3. Universal barrier for the finite Chebyshev system. Recall that the finite Chebyshev system is defined by vectors $v_1, \cdots, v_m$ in $\mathbf{R}^{n+1}$ (for details see e.g. [5]).

We have

$$\mathcal{K}_m = \{x \in \mathbf{R}^{n+1} : \langle v_i, x \rangle \geq 0, i = 1, 2, \cdots, m\}.$$

One can show that $\mathrm{int}(\mathcal{K}_m) \neq \emptyset$ (see e.g. [10],p. 292).

The computable form of the universal barrier has been obtained in [5].

Let $x \in \text{int}(\mathcal{K}_m), n = 2\nu - 1$

$$F_m(x) = \frac{1}{2} \ln \det D_1(x)$$

$$D_1(x) = (d(\alpha, \beta)), \alpha, \beta = 0, 1, \cdots, 2\nu - 1,$$

$$d(\alpha, \beta) = \sum_{i=1}^{m-1} \frac{\det \begin{bmatrix} v_i(\alpha) & v_{i+1}(\alpha) \\ v_i(\beta) & v_{i+1}(\beta) \end{bmatrix}}{\langle v_i, x \rangle \langle v_{i+1}, x \rangle} \tag{3.1}$$

Here

$$v_i = \begin{bmatrix} v_i(0) \\ v_i(1) \\ \vdots \\ v_i(2\nu - 1) \end{bmatrix}$$

In the case of $n = 2\nu$, we need to assume that $v_m = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} = l_{n+1}$.

We used a Householder transform to fulfill this condition. Namely, let

$$H(u) = I - 2uu^T,$$

$$u = \frac{v_m - \|v_m\|_2 l_{n+1}}{\|v_m - \|v_m\|_2 l_{n+1}\|_2}.$$

Then we can easily see that

$$H(u)v_m = \|v_m\|_2 l_{n+1}.$$

We then transform the problem (2.1) - (2.3) using orthogonal transformation $H(u)$. Notice that $H(u)^{-1} = H(u)^T = H(u)$.

Then

$$F_m(x) = \frac{1}{2} \ln D_2(x) - \ln\langle x, v_m \rangle,$$

where

$$D_2(x) = (d_1(\alpha, \beta)), \alpha, \beta = 0, 1, \cdots, 2\nu - 1,$$

$$d_1(\alpha, \beta) = \sum_{i=1}^{m-2} \frac{\det \begin{bmatrix} v_i(\alpha) & v_{i+1}(\alpha) \\ v_i(\beta) & v_{i+1}(\beta) \end{bmatrix}}{\langle v_i, x \rangle \langle v_{i+1}, x \rangle}. \tag{3.2}$$

Notice that for $n = 1$ universal barrier essentially coincides with the classical logarithmic barrier.

Proposition 1. *Let $n = 1$. Then*

$$F_m(x) = \ln\det\begin{bmatrix} v_1(0) & v_m(0) \\ v_1(1) & v_m(1) \end{bmatrix} - \ln\langle v_1, x\rangle - \ln\langle v_m, x\rangle.$$

Proof. Notice that from the general formula (3.1) we have

$$\Delta = \sqrt{\det D_1(x)} = \sum_{i=1}^{m-1} \det\begin{bmatrix} \tilde{v}_i(0) & \tilde{v}_{i+1}(0) \\ \tilde{v}_i(1) & \tilde{v}_{i+1}(1), \end{bmatrix}, \tag{3.3}$$

where $\tilde{v}_i = \frac{v_i}{\langle v_i, x\rangle}$.

Take $x_0 \neq 0$, multiply both parts of (3.3) by it, then add the second row multiplied by x_1 to the first row in each determinant.

We obtain: $$x_0\Delta = \sum_{i=1}^{m-1} \det\begin{bmatrix} 1 & 1 \\ \frac{v_i(1)}{\langle v_i, x\rangle} & \frac{v_{i+1}(1)}{\langle v_{i+1}, x\rangle} \end{bmatrix}$$
$$= \sum_{i=1}^{m-1}\left[\frac{v_{i+1}(1)}{\langle v_{i+1}, x\rangle} - \frac{v_i(1)}{\langle v_i, x\rangle}\right]$$
$$= \frac{v_m(1)}{\langle v_m, x\rangle} - \frac{v_1(1)}{\langle v_1, x\rangle}$$
$$= \frac{v_m(1)[v_1(0)x_0 + v_1(1)x_1] - v_1(1)[v_m(0)x_0 + v_m(1)x_1]}{\langle v_m, x\rangle\langle v_1, x\rangle}$$
$$= \frac{[v_m(1)v_1(0) - v_1(1)v_m(0)]x_0}{\langle v_m, x\rangle\langle v_1, x\rangle}$$

The result follows. □

Remark The constraints $\langle v_i, x\rangle \geq 0, i = 2, \cdots, m-1$ are redundant for a Chebyshev system $v_1, \cdots, v_m$. Thus the final result is not surprising.

For $n = 2$ the universal barrier takes the form

$$F_m(x) = \ln(\sum_{i=1}^{m-2} \frac{\det[v_i, v_{i+1}, v_m]}{\langle v_i, x\rangle\langle v_{i+1}, x\rangle\langle v_m, x\rangle}).$$

This form is different from (3.1) but follows from [5].

Notice that $F_m(x)$ has the form:

$$F_m(x) = \ln(\sum_{i=1}^{m-2} \frac{\delta_i}{\langle v_i, x\rangle\langle v_{i+1}, x\rangle}) - \ln\langle x, v_m\rangle,$$

where $\delta_i = \det[v_i, v_{i+1}, v_m] > 0$.

An easy computation shows:

$$\nabla F_m(x) = \frac{\nabla\Gamma_m(x)}{\Gamma_m(x)} - \frac{v_m}{\langle x, v_m\rangle}$$

$$H_{F_m}(x) = \frac{H_{\Gamma_m}(x)}{\Gamma_m(x)} - \frac{\nabla\Gamma_m(x)\nabla\Gamma_m(x)^T}{\Gamma_m(x)^2} + \frac{v_m v_m^T}{\langle x, v_m\rangle^2}$$

$$\Gamma_m(x) = \sum_{i=1}^{m-2} \frac{\delta_i}{\langle v_i, x\rangle\langle v_{i+1}, x\rangle}$$
$$\nabla\Gamma_m(x) = -\sum_{i=1}^{m-2} \frac{\delta_i}{\langle v_i, x\rangle\langle v_{i+1}, x\rangle}\left(\frac{v_i}{\langle v_i, x\rangle} + \frac{v_{i+1}}{\langle v_{i+1}, x\rangle}\right)$$
$$H_{\Gamma_m}(x) = \sum_{i=1}^{m-2} \frac{\delta_i}{\langle v_i, x\rangle\langle v_{i+1}, x\rangle}\Big(\Big[\frac{v_i}{\langle v_i, x\rangle} + \frac{v_{i+1}}{\langle v_{i+1}, x\rangle}\Big]\Big[\frac{v_i}{\langle v_i, x\rangle} + \frac{v_{i+1}}{\langle v_{i+1}, x\rangle}\Big]^T + \frac{v_i v_i^T}{\langle v_i, x\rangle^2} + \frac{v_{i+1}v_{i+1}^T}{\langle v_{i+1}, x\rangle^2}\Big).$$

Similar formulas can be easily written down for an arbitrary n.

4. Setting up stage for numerical experiments. We have solved a large number of problems using affine-scaling algorithm based on universal barrier functions. Here we describe, however, only one class. Let $u_0, \cdots, u_n$ be a Chebyshev system of continuously differentiable functions and $\mathcal{K}$ is defined as in (1.2). Let $c \in \mathrm{int}(\mathcal{K}^*)$ (interior of the dual cone), $\eta \in [a, b]$.

Consider the following problem:

$$\langle c, x\rangle \to \min,$$

$$p(\eta) = 1, \quad p = \sum_{i=0}^{n} x_i u_i \in \mathcal{K} \tag{4.1}$$

One possible interpretation of this problem: its optimal value is the "maximal mass" of c at the point η (see e.g. [10], Chapter 3, sec. 2). This problem always has an optimal solution. It is very easy to construct $c \in \mathrm{int}(\mathcal{K}^*)$. Choose $\eta_1, \cdots, \eta_{n+1} \in [a, b], \rho_i > 0, i = 1, 2, \cdots, n+1$, consider the vectors

$$W(\xi) = \begin{bmatrix} u_0(\xi) \\ \vdots \\ u_n(\xi) \end{bmatrix}, \xi \in [a, b].$$

Then

$$\sum_{i=1}^{n+1} \rho_i W(\eta_i) \in \mathrm{int}(\mathcal{K}^*). \tag{4.2}$$

Since this construction works for an arbitrary Chebyshev system, it is easy to compare the performance of our algorithms on different Chebyshev systems.

Consider a Chebyshev system

$$1, \cos t, \cos 2t, \cdots, \cos nt$$

on the interval $[0, \pi]$. For this system the problem (4.1) has an analytic solution.

Consider the following somewhat more general situation. Let $S(m, \mathbf{R})$ be the vector space of m by m symmetric matrices and $\Lambda : S(m, \mathbf{R}) \to \mathbf{R}^l$ be a linear map. Let $S_+(m, \mathbf{R})$ be the cone of nonnegative definite symmetric matrices and $\mathcal{K} = \Lambda(S_+(m, \mathbf{R}))$.

We assume that $\mathcal{K}$ is a closed convex cone in $\mathbf{R}^l$ with a nonempty interior.

Let

$$\langle X, Y\rangle_S = \mathrm{Tr}(XY), X, Y \in S(m, \mathbf{R}).$$

We define the dual map $M : \mathbf{R}^l \to S(m, \mathbf{R})$ as follows:

$$\langle M(a), Y\rangle_S = \langle a, \Lambda(Y)\rangle, a \in \mathbf{R}^l, Y \in S(m, \mathbf{R}).$$

Here $\langle , \rangle$ is the standard scalar product in $\mathbf{R}^l$. It is immediate that

$$\mathcal{K}^* = \{c \in \mathbf{R}^l : M(c) \in S_+(m, \mathbf{R})\}.$$

Under some additional assumptions (e.g. $\mathrm{Ker}(\Lambda) \cap S_+(m, \mathbf{R})) = 0$), we will have:

$$\mathrm{int}(\mathcal{K}^*) = \{c \in \mathbf{R}^l : M(c) > 0\},$$

i.e. $M(c)$ is positive definite if and only if $c \in \mathrm{int}(\mathcal{K}^*)$.

The problem

$$\langle c, x\rangle \to \min, \tag{4.3}$$

$$\langle a_i, x\rangle = b_i,\ i = 1, 2, \cdots, k, \tag{4.4}$$

$$x \in \mathcal{K} = \Lambda(S_+(m, \mathbf{R})) \tag{4.5}$$

is equivalent to the semidefinite programming problem:

$$\langle M(c), X\rangle_S \to \min \tag{4.6}$$

$$\langle M(a_i), X\rangle_S = b_i,\ i = 1, 2, \cdots, k, \tag{4.7}$$

$$X \in S_+(m, \mathbf{R}). \tag{4.8}$$

Remark The construction described above is usually behind the semidefinite reductions for optimization problems involving "cones of square" (see e.g. [2]).

Proposition 2. *Let in (4.3) -(4.5), $k = 1$ and there exists $\xi_1, \cdots \xi_s \in \mathbf{R}^m$ such that*

$$M(a_1) = \xi_1\xi_1^T + \cdots + \xi_s\xi_s^T$$

and $M(c) > 0$.

Then the problem (4.3) -(4.5) admits a solution in an analytic form described as follows. Consider s by s matrix

$$\Gamma(c) = (\xi_i^T M(c)^{-1}\xi_j), \quad , i, j = 1, 2, \cdots s.$$

Let $\lambda_{\max}$ be maximal eigenvalue of $\Gamma(c)$. We assume that $\lambda_{\max} > 0$ (i.e. $\Gamma(c) \neq 0$) and $(\eta_1, \cdots, \eta_s)^T$ be corresponding eigenvector normalized by the condition:

$$\eta_1^2 + \eta_2^2 + \cdots + \eta_s^2 = 1.$$

We further suppose that

$$\{x \in \mathbf{R}^l : \langle a_1, x\rangle = b_1, x \in int(\mathcal{K})\} \neq \emptyset \textit{ and } b_1 = 1.$$

Take

$$q = \sum_{j=1}^{s} \frac{\eta_j M(c)^{-1}\xi_j}{\lambda_{\max}}$$

and let $x = \Lambda(qq^T)$*. Then* x *is an optimal solution to (4.3) -(4.5) with optimal value* $\frac{1}{\lambda_{\max}}$.

Proof. We are going to show that qq^T is an optimal solution to the semidefinite programming problem (4.6) -(4.8). Notice that the dual to (4.6) -(4.8) has to form

$$\lambda \to \max \tag{4.9}$$

$$M(c) \geq \lambda M(a_1) = \lambda \sum_{i=1}^{s} \xi_i \xi_i^T. \tag{4.10}$$

Due to our assumptions, both (4.6)-(4.8) and (4.9)-(4.10) have optimal solutions and the optimal values coincide. Let us check that qq^T is feasible for (4.6)-(4.8) and that $\langle M(c), qq^T\rangle_S = \frac{1}{\lambda_{\max}}$.

We have

$$\begin{aligned}
\langle M(c), qq^T\rangle_S &= q^T M(c) q = \frac{1}{\lambda_{\max}^2}\Big(\sum_{j=1}^{s} \eta_j M(c)^{-1}\xi_j\Big)^T\Big(\sum_{i=1}^{s}\eta_i\xi_i\Big) \\
&= \sum_{i,j=1}^{s} \frac{\eta_i\eta_j\xi_j^T M(c)^{-1}\xi_i}{\lambda_{\max}^2} = \frac{\eta^T\Gamma(c)\eta}{\lambda_{\max}^2} = \frac{1}{\lambda_{\max}}.
\end{aligned}$$

$$\begin{aligned}
\langle M(a_1), qq^T\rangle_S &= q^T M(a_1) q = \sum_{i=1}^{s} q^T\xi_i\xi_i^T q = \sum_{i=1}^{s}\langle q, \xi_i\rangle^2 \\
&= \sum_{i=1}^{s} \frac{\langle \sum_{j=1}^{s} \eta_j M(c)^{-1}\xi_j, \xi_i\rangle^2}{\lambda_{\max}^2} \\
&= \sum_{i=1}^{s} \frac{(\sum_{j=1}^{s} \Gamma(c)_{ij}\eta_j)^2}{\lambda_{\max}^2} \\
&= \sum_{i=1}^{s} \frac{\lambda_{\max}^2\eta_i^2}{\lambda_{\max}^2} = 1.
\end{aligned}$$

It is suffices to check that $\lambda = \frac{1}{\lambda_{\max}}$ is feasible to (4.9)-(4.10), i.e. that (4.10) holds for $\lambda = \frac{1}{\lambda_{\max}}$.

But (4.10) is equivalent to

$$\Gamma(c)^{-1} \geq \lambda I$$

(see [9], Proposition 4.3) or $\Gamma(c) \leq \dfrac{1}{\lambda} I$

This inequality clearly holds for $\lambda = \frac{1}{\lambda_{\max}}$ where $\lambda_{\max}$ is the maximal eigenvalue for Γ.

□

Return to the problem 4.1 with $u_i = \cos(it)$. Let us show how to solve (4.1) using Proposition 2.

2. Let

$$\xi_1 = \begin{bmatrix} 1, \\ \cos\eta \\ \vdots \\ \cos n\eta \end{bmatrix}, \quad \xi_2 = \begin{bmatrix} 0, \\ \sin\eta \\ \vdots \\ \sin n\eta \end{bmatrix}$$

Notice that $\xi_1\xi_1^T + \xi_2\xi_2^T = (\cos(i-j)\eta), \quad i,j = 0,1,\cdots,n.$
Consider the map $\Lambda : S(n+1,\mathbf{R}) \to \mathbf{R}^{n+1}$,

$$\Lambda(X) = \begin{bmatrix} x_0, \\ x_1 \\ \vdots \\ x_n \end{bmatrix},$$

$$x_i = 2\sum_{|\alpha-\beta|=i} X_{\alpha\beta}, \ i = 1,2,\cdots,n, \quad x_0 = \sum_{\alpha=0}^{n} X_{\alpha\alpha}.$$

One can show by a direct computation that

$$M(a) = (a_{|i-j|})_{i,j=0,1,\cdots,n} = T(a). \ \text{(This is the definition of } T(a)).$$

Take $a = \xi_1$. Then $\langle a, x\rangle = 1$ is equivalent to $p(\eta) = 1, \ p(\eta) = \sum_{i=0}^{n} x_i \cos(i\eta)$. (with cone K consisting of trigonometric polynomials $p(\eta)$ nonnegative on the interval $[0,\pi]$.)

Our problem (4.1) can be rewritten in the form:

$$\langle T(c), X\rangle_S \to \min, \tag{4.11}$$

$$\langle T(a), X)\rangle_S = 1, \tag{4.12}$$

$$X \in S_+(n+1,\mathbf{R}). \tag{4.13}$$

Notice that $T(a) = \xi_1\xi_1^T + \xi_2\xi_2^T$.
Hence the optimal solution can be obtained using Proposition 2

<u>Example</u> Consider the case $T(c) = I$, i.e. $c = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$.

$$\Gamma(c) = \begin{bmatrix} \langle \xi_1, \xi_1\rangle & \langle \xi_1, \xi_2\rangle \\ \langle \xi_1, \xi_2\rangle & \langle \xi_2, \xi_2\rangle \end{bmatrix}.$$

In our case

$$\langle \xi_1, \xi_1 \rangle = \sum_{k=0}^{n} \cos(k\eta)^2 = \frac{n}{2} + 1 + \frac{\cos[(n+1)\eta]\sin(n\eta)}{2\sin\eta},$$

$$\langle \xi_2, \xi_2 \rangle = \sum_{k=0}^{n} \sin(k\eta)^2 = \frac{n}{2} - \frac{\cos[(n+1)\eta]\sin(n\eta)}{2\sin\eta},$$

$$\langle \xi_1, \xi_2 \rangle = \sum_{k=0}^{n} \sin(k\eta)\cos(k\eta) = \frac{1}{2}\sum_{k=0}^{n} \sin(2k\eta) = \frac{\sin[(n+1)\eta]\sin(n\eta)}{2\sin\eta},$$

$$\lambda_{\max} = \frac{\|\xi_1\|^2 + \|\xi_2\|^2 + \sqrt{(\|\xi_1\|^2 - \|\xi_2\|^2)^2 + 4\langle \xi_1, \xi_2\rangle^2}}{2}$$

$$= \frac{n+1+\sqrt{1+\frac{2\cos(n+1)\eta\sin(n\eta)}{\sin\eta}+\frac{\sin^2(n\eta)}{\sin^2\eta}}}{2}$$

One can show (using elementary manipulations with trigonometric functions) that the expression under the radical is equal to

$$\frac{\sin^2((n+1)\eta)}{\sin^2\eta}$$

Hence, $\dfrac{1}{\lambda_{\max}} = \dfrac{2}{n+1+\left|\frac{\sin(n+1)\eta}{\sin\eta}\right|}.$

Remark This result is obtained in [10] p.75 by completely different methods.

5. Numerical Experiments. Consider the following optimization problem:

$$\langle c, x\rangle \to \min, \tag{5.1}$$

$$\langle d, x\rangle = 1, \tag{5.2}$$

$$\langle v_i, x\rangle \geq 0 \quad i = 1, \cdot, m, \tag{5.3}$$

where

$$c_i = \sum_{j=0}^{n-1} \rho_j u_i(\xi_j), \quad i = 0, 1, \cdots, n-1, \quad \xi_j \in (a, b), \tag{5.4}$$

$$\rho_j \in \mathbf{R}, j = 0, 1, \cdots, n-1.$$

$$d = [u_0(\xi'), u_1(\xi'), \cdots, u_{n-1}(\xi')]^T, \quad \xi' \in (a, b),$$

$$v_i = [u_0(\zeta_i), u_1(\zeta_i), \cdots, u_{n-1}(\zeta_i)]^T, \ \zeta_i \in (a, b), \quad i = 1, 2, \cdots, m.$$

$u_i, \ i = 0, 1, \cdots, n-1$ is a Chebyshev system on the interval $[a, b]$.

Note that $\langle d, x\rangle = 1 \Leftrightarrow p(\xi') = 1$.

In all these examples, we take $\rho_j = 1, \forall j$, $\zeta_i, i = 1, 2, \cdots, m$, $\xi_i, i = 0, 1, \cdots n-1$ form uniform grids on the interval $[a, b]$.

The results of numerical experiments with trigonometric Chebyshev systems are presented in the Tables 1-3 . We solve the problem for the Chebyshev system $1, \cos\vartheta, \cdots, \cos(n-1)\vartheta$. The cost function is chosen according to (5.4) with $\rho_i = 1$. The point ξ_i are chosen as a uniform grid on the interval $[0, \pi], i = 0, 1, \cdots, n-1$. We solve this problem with the polyhedral approximation $\mathcal{K}_m$, (see (5.3)), where $m = 20n$ for the first table, and $m = 30n$ for the second and third table. (In all our experiments we use uniform grid for ζ_i.) We compute Hessian approximately by taking $m' = 15n$ in the first table, and $m' = 20n$ in the second table. Here m' is the upper summation limit in the Hessian (see e.g.(3.1),(3.2)). We compute Hessian without approximation in the third table. In other words, we take $m' = 30n$. Column 2 represents the optimal value for the problem solved by the Affine-scaling algorithm based on the universal barrier function (AU) (step size is $\mu = 0.99$ of the maximal step size). Column 3 represents the optimal value obtained by solving our problem using primal-dual LP algorithm from SeDuMi. Column 4 represents the optimal value of the primal problem solved by classical affine-scaling algorithm (ACl). Column 5 gives optimal value using analytic solution described in Proposition 2 (for the cone $\mathcal{K}$). The next column gives the optimal solution of the dual problem (which in this case coincide with the optimal value of the dual problem) obtained from AU. The next column contains the similar data obtained by processing the optimal dual solution from SeDuMi. The next column-the same data for classical affine-scaling algorithm. The last three columns describe number of iterations for AU, SeDuMi, and ACl.

We see that affine-scaling algorithm based on AU solves problem in 5-8 iterations and significantly outperform both SeDuMi and ACl in this respect. The accuracy of the solution starts deteriorating for both AU and ACl around $n = 25$ but AU still better than ACl (despite the fact that Hessian for AU is computed approximately). SeDuMi provides very stable results in all dimensions.

The picture remains essentially the same in the second table. Here the number of inequality constraints describing the cone $\mathcal{K}_m$ is $30n$ (i.e. greater than in table 1). AU visibly outperforms both SeDuMi and ACl in terms of number of iterations. SeDuMi again provides the most accurate results. In table 3 we finally compute the Hessian of AU without any approximation. We need to adjust the step size in ACl to get accuracy at least comparable with AU. It leads to an increase in a number of iterations for ACl quite significantly. AU yields pretty accurate results and still outperforms SeDuMi in terms of number of iterations.

The following tables present optimal solutions obtained using various barriers and methods.

Table 1 cos-system, $m = 20n$, and $m' = 15n$ for computing Hessian. Equality constraint-$p(\pi/3) = 1$.

n	(P) Opt.Val.	Se(P)Opt.val.	(P)Opt.V.Cl	Analy Sol.	(D)sol.	Se(D) sol.	(D)-Sol-cl	# of it.	# of it.Sed	# of it Cl.
5	2.3348	2.3308	2.3308	2.3333	2.3281	2.3208	2.3308	5	10	5
10	2.1535	2.1530	2.1809	2.1538	2.1531	2.1530	2.1810	7	12	12
15	2.2282	2.2035	2.2302	2.2046	2.2111	2.2035	2.2302	7	15	16
20	2.0787	2.0699	2.1230	2.0706	2.0703	2.0699	2.1230	7	14	21
25	2.1028	2.0648	2.1233	2.0656	2.0773	2.0648	2.1233	7	18	26
30	2.2011	2.1324	2.1534	2.1333	2.1454	2.1324	2.1532	7	18	29
35	2.1241	2.0546	2.0849	2.0556	2.0612	2.0546	2.0849	7	19	26
40	2.0725	2.0460	2.0936	2.0465	2.0530	2.0460	2.0936	8	21	26

Table 2 cos-system, $m = 30n$, and $m' = 20n$ for computing Hessian. Equality constraint-$p(\pi/3) = 1$.

n	(P) Opt.Val.	Se(P)Opt.val.	(P)Opt.V.Cl	Analy Sol.	(D)sol.	Se(D) sol.	(D)-Sol-cl	# of it.	# of it.Sed	# of it Cl.
5	2.3354	2.3322	2.3322	2.3333	2.3322	2.3322	2.3322	5	9	8
10	2.1553	2.1533	2.1767	2.1538	2.1535	2.1533	2.1707	7	14	18
15	2.2396	2.2043	2.2313	2.2046	2.2147	2.2043	2.2313	6	15	19
20	2.0793	2.0704	2.1704	2.0706	2.0714	2.0704	2.1724	7	17	22
25	2.0793	2.0652	2.1725	2.0656	2.0734	2.0652	2.1725	7	19	31
30	2.1486	2.1329	2.1733	2.1333	2.1410	2.1329	2.1773	8	23	25
35	2.0846	2.0553	2.1166	2.0556	2.0622	2.0553	2.1166	7	23	28
40	2.0657	2.0464	2.1241	2.0465	2.0561	2.0464	2.1242	8	25	28

Table 3 cos-system, $m = 30n$, and $m' = 30n$ for computing Hessian. Equality constraint-$p(\pi/3) = 1$.

n	(P) Opt.Val.	Se(P)Opt.val.	(P)Opt.V.Cl	Analy Sol.	(D)sol.	Se(D) sol.	(D)-Sol-cl	# of it.	# of it.Sed	# of it Cl.
5	2.3322	2.3322	2.3322	2.3333	2.3322	2.3322	2.3322	7	9	14
10	2.1533	2.1533	2.1542	2.1538	2.1533	2.1533	2.1542	9	14	20
15	2.2061	2.2043	2.2044	2.2046	2.2062	2.2043	2.2044	12	15	23
20	2.0705	2.0704	2.0706	2.0706	2.0705	2.0704	2.0705	10	17	28
25	2.0671	2.0652	2.0663	2.0656	2.0671	2.0652	2.0661	12	19	36
30	2.1359	2.1329	2.1347	2.1347	2.1359	2.1329	2.1343	14	23	33
35	2.0569	2.0553	2.0557	2.0556	2.0568	2.0553	2.0548	14	23	36
40	2.0478	2.0464	2.0479	2.0465	2.0478	2.0464	2.0476	14	23	42

Results for some other Chebyshev systems

In the next table we consider numerical experiments with the Chebyshev system $1, x, x^2, \cdots, x^{n-1}$ on the interval $[-1, 1]$. We solve this problem with the polyhedral approximation $K_m, m = 20n$. This system turned out to be quite unstable from the point of view of affine-scaling algorithm based on the universal barrier function. In particular, the Hessian $H_F(x)$ is ill-conditioned starting from the first iterations. The primal-dual LP algorithm clearly outperforms our approach in terms of accuracy. We were able to solve the problems for small n.

Table 4 Polynomial -system over $(-1,1)$, $m = 20n$, and $m' = 20n$ for computing Hessian. Equality constraint-$p(0.3) = 1$.

n	(P) Opt.V.(0.67)	Se(P)Opt.V.	(P)Opt.V.Cl (0.67)	(D)sol.	Se(D) sol.	(D)-Sol-cl	# of it.	# of it.Sed	# of it Cl.
5	3.1461	3.1461	3.1462	3.1461	3.1461	3.1462	12	10	13
6	2.8771	2.8770	2.8771	2.8770	2.8970	2.8770	12	9	14
7	2.4724	2.4524	2.4525	2.4724	2.4524	2.4524	15	11	15
8	2.7940	2.7940	2.7940	2.7940	2.7940	2.7940	14	13	16
9	2.8612	2.8611	2.8612	2.8611	2.8611	2.8611	13	10	16
10	3.0687	3.0686	3.0687	3.0686	3.0686	3.0686	13	13	17
11	2.6737	2.6735	2.6736	2.6736	2.6735	2.6736	16	14	18
12	2.6732	2.6731	2.6732	2.6754	2.6731	2.6731	22	19	22
13	2.8014	2.8013	2.8014	2.8013	2.8013	2.8013	20	18	19
14	2.7827	2.7809	2.7810	2.7954	2.7809	2.7809	13	15	19

Table 5 shows numerical experiments when we try to interpolate between the universal barrier F_{ch} and the classical barrier F_{cl} for the polyhedral approximation of the cone of nonnegative trigonometric polynomials. The "pure" universal barrier clearly outperforms any interpolation. Notice, however that always

$$H_{F_{cl}}(x) - H_{F_{ch}}(x) > 0.$$

In particular, $H_{F_{cl}}$ is much better conditioned than $H_{F_{ch}}$

Table 5 cos -system $(0,\pi)$ $n = 20$, $m = 30n$, and $m' = 20n$ for computing Hessian. Equality constraint-$p(\pi/3) = 1$ using barrier $F = \epsilon F_{ch} + (1-\epsilon)F_{cl}$.

ϵ	(P) Opt. Val.	(P) Opt. val. (Sed)	# of it.	# of it. (Sed)	(D) Opt. Val.	(D) Opt. val. (Sed)
1	2.0793	2.0704	7	19	2.0714	2.0704
0.8	2.1423	2.0704	15	19	2.1423	2.0704
0.5	2.1617	2.0704	19	19	2.1617	2.0704
0.3	2.1841	2.0704	17	19	2.1841	2.0704
0	2.1724	2.0704	22	19	2.1724	2.0704

Tables 6-11 show numerical experiments with some other Chebyshev systems. Table 6 shows the result where we do not use a polyhedral approximation of the cone K. We use numerical integration package DE–Quadrature by T. Ooura (http://www.kurims.kyoto-u.ac.jp/ ooura/index.html and [15]) for precise computation of the integrals of the type (1.1). In this case, we of course cannot check feasibility easily. Instead, we are moving along the affine-scaling direction till we hit the boundary of the Dikin's ball. To compare with LP results we need to take a very refined polyhedral approximation (m = 1000n). The accuracy is good for small n. Some results for polyhedral approximations for the cones of nonnegative exponential systems are given in the Table 7.

Table 6 (Continuous)Exponential system $1, e^t, e^{2t}, \cdots, e^{(n-1)t}$ over $(-1,1)$, Equality constraint $p(0.3) = 1$.

n	(P) Opt. Val (# of It.).	(D) Opt. Val	(P) Opt. Val Se m = 1000n. (# of It.)	(D) Opt. Val Se.
6	2.2867(13)	2.2860	2.2860(20)	2.2860

Table 7 Exponential system $1, e^t, e^{2t}, \cdots, e^{(n-1)t}$ over $(-1,1)$, $p(0.3) = 1$, $m = 50n$, and $m' = 50n$ for computing Hessian.

n	(P) Opt.Val. (0.33)	Se(P)Opt.val.	(P)Opt.V.Cl (0.33)	(D)sol.	Se(D) sol.	(D)-Sol-cl	# of it.	# of it.Sed	# of it Cl.
6	2.2844	2.2841	2.2844	2.2839	2.2841	2.2842	40	17	45
7	2.5840	2.5839	2.5841	2.5839	2.5839	2.5837	20	12	33
8	2.9568	2.9190	2.9193	2.8650	2.9190	2.9190	21	17	36
9	2.9323	2.7694	2.2694	3.1664	2.7702	2.2702	31	12	79

Table 8 gives some computational results for cones of nonnegative polynomial splines which form a weak Chebyshev system. We fix the degree of polynomial to 5, and we add the spline terms; n is the number of functions in the system. We were able to solve the problems for small n. With a rather small step size $\mu = 0.33$, the number of iterations is comparable to SeDuMi. We start to lose accuracy when n is larger then 9.

Table 8 Spline system $1, t, t^2, t^3, t^4, t^5, (t+0.2)^5_+, (t+0.4)^5_+, \cdots$,over $(-1,1)$, $p(0.3) = 1$. $m = 50n$, and $m' = 50n$ for computing Hessian.

n	(P) Opt.Val. (0.33)	Se(P)Opt.val.	(P)Opt.V.Cl (0.33)	(D)sol.	Se(D) sol.	(D)-Sol-cl	# of it.	# of it.Sed	# of it Cl.
6	2.8883	2.8883	2.8883	2.8883	2.8883	2.8883	12	10	14
7	2.4677	2.4676	2.4678	2.4676	2.4676	2.4678	13	12	15
8	2.7009	2.7009	2.7009	2.7041	2.7009	2.7007	13	13	18
9	2.9436	2.9428	2.9431	2.9484	2.9428	2.9428	13	13	16

The next table, Table 9, displays computational results for hyperbolic sinh and cosh system. We were able to solve the problems for small n with good accuracy. For these experiments, we take step size $\mu = 0.67$. As before, we were able to solve the problems for small n.

Table 9 System- $1, \cosh t, \sinh t, \cosh 2t, \sinh 2t, \cdots, \cosh \frac{(n-1)}{2}t, \sinh \frac{(n-1)}{2}t$, over$(-1,1)$, $p(0.3) = 1$. $m = 50n$, and $m' = 50n$ for computing Hessian.

n	(P) Opt.Val.(0.67)	Se(P)Opt.val.	(P)Opt.V.Cl (0.67)	(D)sol.	Se(D) sol.	(D)-Sol-cl	# of it.	# of it.Sed	# of it Cl.
5	3.2027	3.2027	3.2027	3.2027	3.2027	3.2027	12	12	13
7	2.4859	2.4858	2.4859	2.4857	2.4859	2.4859	20	13	19

Table 10 displays computational results for another interesting Chebyshev system. We were able to solve this problem for small n. Also with large number of grids used to compute Hessian and feasibility check, we get accurate results.

Table 10 System $1, \cos t, \cdots, \cos 5t, \frac{1}{2+\cos t}$, over $(0, \pi)$, $p(\pi/3) = 1$. $m = 50n$, and $m' = 50n$ for computing Hessian.

(P) Opt.Val.(0.67)	Se(P)Opt.val.	(P)Opt.V.Cl (0.67)	(D)sol.	Se(D) sol.	(D)-Sol-cl	# of it.	# of it.Sed	# of it Cl.
2.3330	2.3327	2.3329	2.3334	2.3327	2.3327	12	13	17

Table 11 shows numerical experiments when we try to use three different types of barriers. One is universal barrier generated by the Chebyshev system F_{ch}, another barrier is the classical barrier F_{cl}. The other barrier F_{adj} is the combination between the two barriers. We use F_{ch} for the very first iterations. Once the Hessian is ill conditioned and the minimum eigenvalue becomes negative, we switch to F_{cl}. For this, we take $m = 50n$, and $m' = 50n$ for computing Hessian. The step size $\mu = 0.99$ was used. We compare the optimal value to the optimal value obtained by using SeDuMi.

Table 11 Polynomial system. $n = 12$ using three types of barrier functions

Barrier	P Opt. Val.	D-Opt Val.	# of iterations
F_{ch}	2.7851	2.9430	6
F_{cl}	2.7199	2.7199	13
F_{adj}	2.7225	2.7225	16
SeDuMi	2.6755	2.6755	17

Finally, we consider the following optimization problem:

$$\langle c, x\rangle \to \min, \tag{5.5}$$

$$\langle d, x\rangle = 1, \quad \sum_{i=0}^{n-1} x_i = 1, \quad x_0 + 2x_1 + 3x_2 = 1, \tag{5.6}$$

$$\langle v_i, x\rangle \geq 0, \; i = 1, \cdots, m \tag{5.7}$$

where

$$c_i = \sum_{j=0}^{n-1} \rho_j u_i(\xi_j), \quad i = 0, 1, \cdots, n-1, \quad \xi_j \in (0, \pi), \quad \rho_j \in \mathbf{R}, j = 0, 1, \cdots, n-1.$$

$$d = [u_0(\xi'), u_1(\xi'), \cdots, u_{n-1}(\xi')]^T, \quad \xi' \in (0, \pi),$$

$$v_i = [u_0(\zeta_i), u_1(\zeta_i), \cdots, u_{n-1}(\zeta_i)]^T, \zeta_i \in (0, \pi), \quad i = 1, 2, \cdots, m.$$

$u_i = \cos(\zeta)$, $i = 0, 1, \cdots, n-1$ is a cos–system on the interval $[0, \pi]$.

Table 12 shows numerical results for solving the problem (5.5)–(5.7) using the Affine-scaling algorithm based on the universal barrier function and classical affine-scaling algorithm both with step size $\mu = 0.67$. We also compare our results with primal–dual algorithm solved by SeDuMi. The results were quite satisfactory. Also the number of iterations using universal barrier function is clearly fewer than the number of iterations using classical affine-scaling algorithm

Table 12 Results for the problem (5.5)–(5.5) with $\xi' = \pi/3$, $m = 50n$, and $m' = 50n$ for computing Hessian.

n	(P) Opt.V.	Se(P)Opt.V.	(P)Opt.V.Cl	# of it.	# of it.Sed	# of it Cl.
5	3.8570	3.8568	3.6571	10	11	11
10	2.6564	2.6559	2.6561	12	17	22
15	3.0591	3.0572	3.0573	12	14	21
20	3.5712	3.5702	3.5703	13	17	29
25	4.0016	3.9981	3.9983	13	19	29
30	4.4052	4.4036	4.4037	13	18	27
35	4.9217	4.9175	4.9180	14	20	32
40	5.4025	5.3999	5.4000	14	24	36

6. Concluding remarks. Our experiments confirm the theoretical assumptions that algorithms based on the universal barrier function significantly reduce the number of iterations without loss in accuracy. We also computed the Hessian using various approximation e.g. by reducing the number of grid points. Of course our approximations are of very special nature but they lead to satisfactory numerical results. However, for certain Chebyshev systems (e.g. polynomial) the Hessian of the universal barrier function is ill-conditioned starting from the first iterations. It leads to the break down of the algorithm for $n \approx 14$. This is to some extent similar to the situation arising for the same class of problems when they solved by the reduction to semi-definite programming problem [8]. In the former case the explanation is based on ill-conditioning of positive definite Hankel matrices. Notice, however that due to the invariance of universal barrier under linear isomorphisms of cones, the mechanism of ill-conditioning is probably related to the sensitivity of our method to the errors arising in the description of polyhedral approximations of the cone of nonnegative polynomials. In our opinion, it is worthwhile to continue to study numerical properties of algorithms based on the approximations of the universal and other self-concordant barrier functions.

REFERENCES

[1] L. Faybusovich, *Euclidean Jordan algebras and interior-point algorithms*, Positivity, vol. 1 (1997), no 4, pp. 331-357.

[2] L. Faybusovich, *On Nesterov's approach to semi-infinite programming*, Acta Appl. Math. vol. 74 (2002), no 2, pp. 195-215.

[3] L. Faybusovich, *Self-concordant barriers for cones generated by Chebyshev systems*, SIAM J. Optim., vol. 12 (2002), pp. 770-781.

[4] L. Faybusovich and R. Arana, *A long-step primal-dual algorithm for symmetric programming problem*, Syst. Contr. Letters, vol. 43 (2001), no 1, pp 3-7.

[5] L. Faybusovich and M. Gekhtman, *Calculation of universal barrier functions for cones generated by Chebyshev systems over finite sets*, SIAM J. Optim. vol. 14 (2004), no 4, pp. 965-979.

[6] L. Faybusovich, T. Mouktonglang and T. Tsuchiya, *Implementation of infinite-dimensional interior-point method for solving multi-criteria linear-quadratic control problem*, Optim. Methods Softw. vol. 21 (2006), no 2. pp. 315-341.

[7] L. Faybusovich and T. Tsuchiya, *Primal-dual algorithms and infinite-dimensional Jordan algebras of finite rank*, Mathematical Programming, vol.97 (2003), no. 3, pp. 471-493.

[8] Y. Genin, Y. Hachez, Y. Nesterov and P. Van Dooren, *Optimization problems over positive pseudopolynomial matrices*, SIAM J. Matrix Anal. Appl. vol 25 (2003), no 1, pp. 57-79.

[9] Y. Hachez and Y. Nesterov, *Optimization problems over nonnegative polynomials with interpolation constraints* In: Positive polynomials in control and Inf. sci. vol. 312, Springer, Berlin, 2005.

[10] M. Krein and A. Nudelman, *The Markov Moment Problem and Extremal Problems*, AMS, Provideuce, RI, 1977.

[11] M. Muramatsu , *Affine-scaling algorithm fails for semidefinite programming*, Math. Progr., vol. 83 (1998) , pp. 393-406.

[12] Y. Nesterov and A. Nemirovskii, *Interior-point Polynomial algorithms in Conver Programming*, SIAM Stud. Appl. Math. 13, SIAM, Philadelphia, 1994.

[13] Y. Nesterov and M. Todd, *Primal-dual interior-point methods for self-scaled cones*, SIAM.J.OPT., vol. 8 (1998), no 2, pp. 324-364.

[14] J. Sturm, *Implementation of interior point methods for mixed semidefinite and second order cone optimization problems*, Optim . Methods Softw. vol. 17 (2002), no 6. 1105-1154.

[15] H. Takahasi and M. Mori, *Double exponential formulas for numerical integration*, Pub. RIMS Kyoto Univ. 9, 1974, 721-741.

[16] T. Tsuchiya and M. Muramatsu, *Global convergence of a long-step affine scaling algorithm for degenerate linear programming problem*, SIAM J. Optim. 5 (1995), n0. 3 pp. 525-551.

[17] M. Wright, *The interior-point revolution in optimization: history, recent development and lasting consequences*, Bull. Amer. Math. Soc., vol. 42 (2005), no 7, pp. 39-56.