数理解析研究所講究録
1198巻 2001年 186-194

半正定値計画問題に現れる密行列のための一般化共役残差法

張紹良　中田和秀　新田義寛

東京大学大学院工学系研究科物理工学専攻

Shao-Liang Zhang　Kazuhide Nakata　Yoshihiro Nitta

Department of Applied Physics, The University of Tokyo

Abstract

It is a great concern in the field of Semidefinite Programming to solve the large and dense linear systems which arise from the interior-point method. Often direct methods are too expensive in terms of computer memory and CPU-time requirements, then the only alternative is to use iterative methods. Here, we apply the generalized conjugate residual method to this type of SDP's linear systems. The theoretical properties and practical performance of the method will be discussed.

1 Introduction

Recently Semidefinite Programming (SDP hereafter) has been considered as an extension of Linear Programming. It has a number of interesting applications to physical problems, control problems and other mathematical programming [16]. It is well-known that the interior-point methods are well suited for solving SDP with the primal problem and its dual problem (primal-dual SDP hereafter), and theoretical results and computational experiments have shown this fact [1,5].

As an iterative approach, it is necessary to determine a search direction for the next approximate solution in the interior-point methods. There is an elegant and powerful class of ways for determining the search direction. Throughout the present paper, we only focus our attention on the so-called HRVW/KSH/M search direction [6,9,11,12]. When one uses the interior-point methods for solving primal-dual SDP, one usually has to efficiently solve a large and dense linear system (SDP's linear system hereafter) as an immediate problem at each iterative step where its coefficient matrix is an $m \times m$ symmetric positive

definite and dense matrix in general. If the size of the matrix is not large, direct methods such as Cholesky factorization are often used and its attractive efficiency has been confirmed by many numerical experiments using some successful libraries [4,15,2,14]. However, if the size of the matrix is getting larger, all direct approaches are prohibitive in terms of expensive computational costs of $O(m^3)$ and much storage requirements to keep all elements of m^2.

The only alternative is to use iterative methods for which it is not necessary to keep simultaneously all elements of the matrix [18,10,13]. The purpose of the present paper is to apply the generalized conjugate residual method (GCR hereafter) for solving SDP's linear systems.

The plan of the present paper is as follows. First, we will introduce a simple description of SDP and the interior-point methods for solving primal-dual SDP, and consider SDP's linear systems in section 2. After making a clear understanding of the structure of the matrices of SDP's linear systems by using rewriting the matrices, we will consider to apply GCR for solving SDP's linear systems, and discuss the theoretical properties of GCR in section 3. Finally, we will give several numerical experiments to confirm the practical performance of GCR in section 4.

2 SDP and Primal-Dual Interior-Point method

Let $R^{n\times n}$ denote the set of all $n \times n$ real matrices, $S^{n\times n}$ denote the set of all $n \times n$ symmetric real matrices. We define an inner product of X and Y in $R^{n\times n}$ by

$$X \bullet Y \equiv Tr(X^T Y)$$

where $Tr(A)$ denotes the *trace* of A, the sum of its diagonal elements. For $X \in S^{n\times n}$, we write $X \succ O$ to denote positive definiteness of X and $X \succeq O$ to denote positive semidefiniteness of X.

Let $A_i \in S^{n\times n}$ $(0 \leq i \leq m)$, $b_i \in R$ $(1 \leq i \leq m)$. Here we assume that A_i $(1 \leq i \leq m)$ are linearly independent which implies that $m \leq n(n+1)/2$. Using above notation, let us recall the standard form of primal-dual SDP often described as follows:

$$
\begin{array}{lll}
\textbf{Primal problem:} & \text{minimize} & A_0 \bullet X \\
& \text{subject to} & A_i \bullet X = b_i \quad (1 \le i \le m), \quad X \succeq O. \\
\textbf{Dual problem:} & \text{maximize} & \sum_{i=1}^{m} b_i y_i \\
& \text{subject to} & \sum_{i=1}^{m} A_i y_i + Z = A_0, \qquad Z \succeq O.
\end{array} \tag{1}
$$

$(X, \boldsymbol{y}, Z)$ is called an optimal solution of (1) if X is an optimal solution of **Primal problem** and $(\boldsymbol{y}, Z)$ is an optimal solution of **Dual problem**.

Interior-point approaches are well suited for solving primal-dual SDP [1,16], and produce iterative solutions along the so-called central path which is parameterized by parameter $\mu > 0$ as follows.

$$
A_i \bullet X = b_i (1 \le i \le m), \ \sum_{i=1}^{m} A_i y_i + Z = A_0, \ XZ = \mu I, \ X \succ O, \ Z \succ O.
$$

Note that Newton's method can not be directly applied to the above nonlinear equations, because of the different dimensions.

To solve numerically the above nonlinear equations, a common technique is to choose a search direction $(dX, d\boldsymbol{y}, dZ)$, then iterate the next approximate solution as follows.

$$
(X, \boldsymbol{y}, Z) = (X, \boldsymbol{y}, Z) + \alpha (dX, d\boldsymbol{y}, dZ).
$$

Several successful approaches for search directions have already been proposed. Here we only discuss the HRVW/KSH/M search direction which can be reduced as follows [6,9,11,12].

$$
\begin{array}{ll}
B d\boldsymbol{y} = \boldsymbol{f}, & (2) \\
B_{ij} = A_i \bullet X A_j Z^{-1} \ (1 \le i \le m, 1 \le j \le m), & (3) \\
f_i = p_i - A_i \bullet X(X^{-1}K - D)Z^{-1} \ (1 \le i \le m), & \\
dZ = D - \sum_{i=1}^{m} A_i dy_i, \ d\hat{X} = X(X^{-1}K - dZ)Z^{-1}, \ dX = (d\hat{X} + d\hat{X}^T)/2, &
\end{array}
$$

where $K = X \bullet Z\beta/nI - XZ$ for a parameter $\beta \in (0, 1)$. In general, the $m \times m$ matrix B is usually dense even if A_i are sparse.

3 CG and GCR for the SDP's linear system

In this section, we first discuss the property of B. Note that for the symmetric positive definite X and Z in (3), we have $X = X^{1/2}X^{1/2}$ and $Z^{-1} = Z^{-1/2}Z^{-1/2}$ where $X^{1/2}$ and $Z^{-1/2}$ are the square roots of X and Z^{-1} respectively.

Using $Tr(A) = Tr(A^T)$ and $Tr(AB) = Tr(BA)$, then we have

$$\begin{aligned} B_{ij} &= A_i \bullet XA_jZ^{-1} = Tr(A_i^T \ XA_jZ^{-1}) \\ &= Tr((X^{1/2}A_iZ^{-1/2})^T \ X^{1/2}A_jZ^{-1/2}) \\ &= (X^{1/2}A_iZ^{-1/2}) \bullet (X^{1/2}A_jZ^{-1/2}) \\ &= (X^{1/2}A_jZ^{-1/2}) \bullet (X^{1/2}A_iZ^{-1/2}) = B_{ji} \end{aligned}$$

which means that B is symmetric.

In addition, let $\boldsymbol{a}_{i1}, \cdots, \boldsymbol{a}_{in} \in R^n$ denote the columns of the matrix $X^{1/2}A_iZ^{-1/2}$, and $\boldsymbol{c}_i \in R^{n^2}$ denote the vector $(\boldsymbol{a}_{i1}^T, \cdots, \boldsymbol{a}_{in}^T)^T$, then we can rewrite B_{ij} as follows.

$$B_{ij} = Tr((\boldsymbol{a}_{i1}, \cdots, \boldsymbol{a}_{in})^T \ (\boldsymbol{a}_{j1}, \cdots, \boldsymbol{a}_{jn})) = (\boldsymbol{c}_i, \boldsymbol{c}_j).$$

Let C be the $n^2 \times m$ matrix with columns $\boldsymbol{c}_1, \cdots, \boldsymbol{c}_m$, then B can be expressed as

$$B = C^TC \qquad \in S^{m \times m}$$

which implies that B is positive definite.

Since B is symmetric and positive definite, it is natural to apply the conjugate gradient method (CG hereafter) to SDP's linear system (2) [18,10,13]. The algorithm of CG can be described as follows [7].

ALGORITHM 1 *CG*

$\boldsymbol{x}_0$ *is an initial guess,* $\boldsymbol{r}_0 = \boldsymbol{f} - B\boldsymbol{x}_0$, *set* $\boldsymbol{p}_0 = \boldsymbol{r}_0$,
For $i = 0, 1, \cdots$ *until* $\| \boldsymbol{r}_i \| \ / \ \| \boldsymbol{r}_0 \| \leq \varepsilon$ Do :

$$\alpha_i = \frac{(\boldsymbol{r}_i, \boldsymbol{r}_i)}{(\boldsymbol{p}_i, B\boldsymbol{p}_i)},$$

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \alpha_i\boldsymbol{p}_i, \quad \boldsymbol{r}_{i+1} = \boldsymbol{r}_i - \alpha_iB\boldsymbol{p}_i,$$

$$\beta_i = \frac{(\boldsymbol{r}_{i+1}, \boldsymbol{r}_{i+1})}{(\boldsymbol{r}_i, \boldsymbol{r}_i)},$$

$\boldsymbol{p}_{i+1} = \boldsymbol{r}_{i+1} + \beta_i \boldsymbol{p}_i,$
EndDo

However, when the approximate solution $(X, \boldsymbol{y}, Z)$ is close to the optimal solution of (1), one has to encounter an ill-posed matrix B because X and Z become nearly singular. In general, CG for solving singular linear systems usually shows a bad performance of convergence, furthermore diverges [8]. Note that the residual $\boldsymbol{r}_i$ in CG satisfies

$$||\boldsymbol{r}_i||_{B^{-1}} = \min_{\boldsymbol{x} \in \boldsymbol{x}_0 + K_i(B, \boldsymbol{r}_0)} ||\boldsymbol{f} - B\boldsymbol{x}||_{B^{-1}}$$

where $K_i(B, \boldsymbol{r}_0) = Span\{\boldsymbol{r}_0, B\boldsymbol{r}_0, \cdots, B^{i-1}\boldsymbol{r}_0\}$.

In the present paper, we prefer to apply GCR to SDP's linear systems whether B become nearly singular. The algorithm of GCR can be described as follows [3].

ALGORITHM 2 *GCR*

$\boldsymbol{x}_0$ *is an initial guess,* $\boldsymbol{r}_0 = \boldsymbol{f} - B\boldsymbol{x}_0$, *set* $\boldsymbol{p}_0 = \boldsymbol{r}_0$,
For $i = 0, 1, \cdots$ *until* $\| \boldsymbol{r}_i \| / \| \boldsymbol{r}_0 \| \leq \varepsilon$ Do :

$$\alpha_i = \frac{(\boldsymbol{r}_i, B\boldsymbol{p}_i)}{(B\boldsymbol{p}_i, B\boldsymbol{p}_i)},$$

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \alpha_i \boldsymbol{p}_i, \quad \boldsymbol{r}_{i+1} = \boldsymbol{r}_i - \alpha_i B\boldsymbol{p}_i,$$

$$\beta_{i,j} = -\frac{(B\boldsymbol{r}_{i+1}, B\boldsymbol{p}_j)}{(B\boldsymbol{p}_j, B\boldsymbol{p}_j)}, \quad j = 0, 1, \cdots, i \tag{4}$$

$$\boldsymbol{p}_{i+1} = \boldsymbol{r}_{i+1} + \sum_{j=0}^{i} \beta_{i,j} \boldsymbol{p}_j, \tag{5}$$

$$B\boldsymbol{p}_{i+1} = B\boldsymbol{r}_{i+1} + \sum_{j=0}^{i} \beta_{i,j} B\boldsymbol{p}_j, \tag{6}$$

EndDo

There are three definite reasons why we chose GCR for solving the SDP's linear systems. The first reason is that GCR always converges when the coefficient matrix is symmetric and positive definite even if singular [17]. The second reason is due to the following property of GCR

$$||\boldsymbol{r}_i||_2 = \min_{\boldsymbol{x} \in \boldsymbol{x}_0 + K_i(B, \boldsymbol{r}_0)} ||\boldsymbol{f} - B\boldsymbol{x}||_2.$$

It implies that GCR will give a better performance of convergence than CG. The third one is due to the fact that $\beta_{i,j} = 0$ for $j = 0, 1, \cdots, i-1$ in (4) when B

is symmetric and positive definite. Therefore, (4), (5) and (6) in ALGORITHM 2 can be reduced as the follows.

$$\beta_i = -\frac{(B\boldsymbol{r}_{i+1}, B\boldsymbol{p}_i)}{(B\boldsymbol{p}_i, B\boldsymbol{p}_i)}, \ \boldsymbol{p}_{i+1} = \boldsymbol{r}_{i+1} + \beta_i \boldsymbol{p}_i, \ B\boldsymbol{p}_{i+1} = B\boldsymbol{r}_{i+1} + \beta_i B\boldsymbol{p}_i.$$

It implies that GCR can be reduced as the conjugate residual method (CR hereafter).

Table 1 shows a comparison between CG and CR with respect to computational costs at each iteration and memory.

Table 1. Comparison w.r.t. cost and memory.

	CG	CR
matrix-vector multiplication	1	1
vector updates	6m	8m
inner products	2m	3m
memory vectors	4m	4m

Although CR is expensive than CG with respect to vector updates and inner products at each iteration, the total computational costs during each iteration in CG and CR are almost equal because the dense matrix-vector multiplication is the major part. Fewer iterations to achieve the stopping criterion means better performance of convergence.

4 Numerical experiments

In this section, we report several numerical experiments to compare the computational performance between CG and CR. All experiments have been carried out with double precision floating point arithmetic on an Alpha computer (750MHz). We always choose the initial guess $\boldsymbol{x}_0 = \boldsymbol{0}$ in ALGORITHM 1 and ALGORITHM 2. All figures throughout this section display the logarithmic relative residual 2-norm $\log_{10} ||\boldsymbol{r}_i||/||\boldsymbol{r}_0||$ (on the vertical axis) versus the iteration number i (on the horizontal axis).

Here we consider a Maximum Clique problem. Here all data in (1) are given as follows: A_0 is the matrix whose all elements are equal to 1, A_1 is the identity matrix, A_l $(l = 2, \cdots, m)$ is a matrix in which the ij-th and ji-th elements are equal to -1 for a given integer pair (i, j) $(1 \leq i < j \leq n)$ and else elements are all equal to 0, and $b_1 = 1$ and $b_l = 0$ $(l = 2, \cdots, m)$.

We applied the interior-point method to a Maximum Clique problem and its dual problem simultaneously with $m = 9957$ and $n = 200$. The following two figures show the numerical results of CG versus CR for solving SDP's linear systems at 5th step and 10th step in the interior-point method respectively. The stopping criterion is set as 10^{-3}. Table 2 shows the performance for CG and CR in terms of the total numbers of iterations and CPU times to achieve the stopping criterion. Throughout the Maximum Clique problem, we suppose deliberately that the all elements of the matrix B in (2) can be not kept simultaneously in the memory of the computer. To compute $B\boldsymbol{p}_i$ in ALGORITHM 1 and ALGORITHM 2, the element B_{ij} is computed extemporaneously via (3) at each iterative step. Note that the computation of $B\boldsymbol{p}_i$ costs very much, then takes most of CPU time.

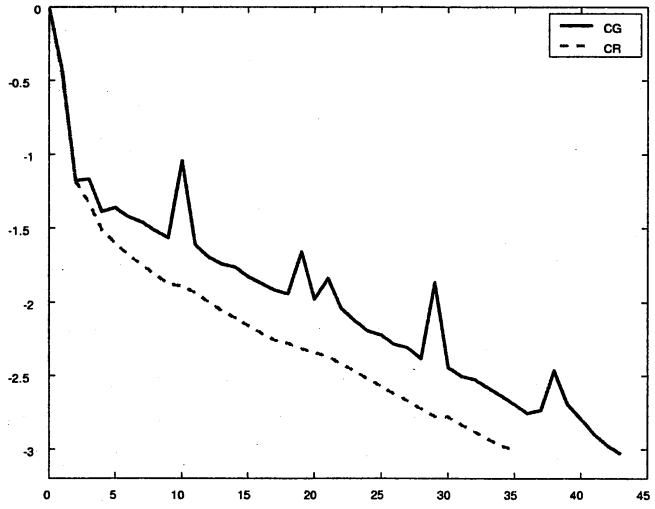

Fig. 1. Residual history (at 5th step).

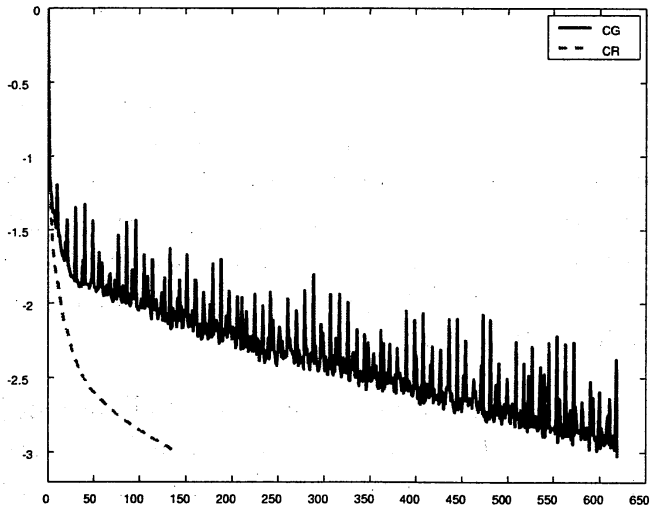

Fig. 2. Residual history (at 10th step).

We observe that CR began to converge faster than CG at 5th step from Fig. 1 and Table 2. At 10th step where the approximate solution $(X, \boldsymbol{y}, Z)$ in the interior-point method was very close to the optimal solution, we had to solve an ill-posed matrix B because X and Z became numerically nearly singular. From Fig. 2 and Table 2, we observe that CR became more efficient and more useful for this difficult one. Therefore, it can be said that CR is suitable for the Maximum Clique problem.

Table 2. The performance for the Maximum Clique problem.

	CG		CR	
	Iterations	CPU time(s)	Iterations	CPU time(s)
5th step	43	882.81	35	743.49
10th step	619	12180.48	142	2833.92

5 Concluding remarks

In the present paper, we considered the application of GCR to the large and dense linear systems which arise from Semidefinite Programming. Comparing the theoretical aspect of CG and GCR, we emphasize that GCR has some attractive and competitive features for solving SDP's linear systems. The numerical experiments showed that GCR is suitable and powerful for solving SDP's linear systems.

References

[1] F. Alizadeh, J.-P. A. Haeberly and M. L. Overton, *Primal-dual interior-point methods for semidefinite programming: convergence rates, stability and numerical results*, SIAM J. Optim., **8** (1998) 746–768.

[2] B. Borchers, *CSDP 2.3 user's guide*, Optimization Methods and Software, **11-12** (1999) 597–611.

[3] S. C. Eisenstat, H. C. Elman and and M. H.Schultz, *Variational Iterative Methods for Nonsymmetric Systems of Linear Equations*, SIAM J. Numer. Anal., **20** (1983) 345–357.

[4] K. Fujisawa, M. Kojima and K. Nakata, *SDPA (Semidefinite Programming Algorithm) - user's manual -*, (Technical Report B-308, Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, 1995).

[5] K. Fujisawa, M. Fukuda, M. Kojima and K. Nakata, *Numerical evaluation of SDPA (SemiDefinite Programming Algorithm)*, in: H. Frenk el at.,eds, High Performance Optimization, (Kluwer Academic Publishers, 1999) 267–301.

[6] C. Helmberg, F. Rendl, R. J. Vanderbei and H. Wolkowicz, *An interior-point method for semidefinite programming*, SIAM J. Optim., **6** (1996) 342–361.

[7] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Standards, **49** (1952) 409–435.

[8] E. F. Kaasschieter, *Preconditioned conjugate gradients for solving singular systems*, J. Comput. Appl. Math., **24** (1988) 265-275.

[9] M. Kojima, S. Shindoh and S. Hara, *Interior-point methods for the monotone semidefinite linear complementarity problems*, SIAM J. Optim., **7** (1997) 86–125.

[10] C. Lin and R. Saigal, *On solving large-scale semidefinite programming problems– a case study of quadratic assignment problem*, (Technical report, Department of Industrial and Operations Engineering, University of Michigan, 1998).

[11] R. D. C. Monteiro, *Primal-dual path-following algorithms for semidefinite programming*, SIAM J. Optim., **7** (1997) 663–678.

[12] R. D. C. Monteiro and Y. Zhang, *A unified analysis for a class of path-following primal-dual interior-point algorithms for semidefinite programming*, Math. Programming, **81** (1998) 281–299.

[13] K. Nakata, K. Fujisawa and M. Kojima, *Using the conjugate gradient method in interior-point methods for semidefinite programs (in Japanese)*, Proceedings of the Institute of Statistical Mathematics, **46** (1998) 297–316.

[14] F. A. Potra, R. Sheng and N. Brixius, *SDPHA - a MATLAB implementation of homogeneous interior-point algorithms for semidefinite programming*, (Department of Mathematics, University of Iowa, 1997).

[15] K. C. Toh, M. J. Todd and R. H. Tütüncü, *SDPT3 - a MATLAB software package for semidefinite programming*, (Dept. of Mathematics, National University of Singapore, 1996).

[16] L. Vandenberghe and S. Boyd, *Semidefinite programming*, SIAM Rev., **38** (1996) 49–95.

[17] S.-L. Zhang, Y. Oyanagi and M. Sugihara, *Necessary and Sufficient Conditions for Convergence of Orthomin(k) on Singular and Inconsistent Linear Systems*, Numer. Math., **87-2** (2000) 391-405.

[18] Q. Zhao, S. E. Karisch, F. Rendl and H. Wolkowicz, *Semidefinite programming relaxation for the quadratic assignment problem*, (CORR Report 95-27, Department of Combinatorics and Optimization, University of Waterloo, 1996).