Available online at www.pelagiaresearchlibrary.com

Pelagia Research Library

Advances in Applied Science Research, 2011, 2 (6):24-31

Pelagia Research Library

ISSN: 0976-8610
CODEN (USA): AASRFC

On Optimal Termination Rule for Primal-Dual Algorithm for Semi-Definite Programming

B.O. Adejo[1] and E. Ogala[2]

Department of Mathematical Sciences, Kogi State University, Anyigba

---

ABSTRACT

In this article, we compare three previous termination rules for primal-dual short step path-following algorithm for semi-definite programming proposed earlier by Monteiro, Adejo and Adejo and Singh independently, which were based on analyses carried out independently by Franklin and Singh et al for Karmarkar's algorithm for linear programming, Here, we develop a more efficient termination rule which on implementation saves at least 95% iterations over that of Monteiro and at least 8% iterations over that of Adejo and Singh.

Keywords: Semi-definite, upper ceiling function $\lceil \quad \rceil$, primal-dual methods, path-following methods, NP-hard problems.

INTRODUCTION

Not long after Karmarkar [5] in 1984 developed the first ever projective interior point algorithm that guaranteed iterates that lie in the interior of the feasible set, it was recognized that interior point methods (IPMS) for LPs can be used the same way for a matrix version of linear programming problems (LPs) as well. this matrix version is Known as semi-definite programs (SDPs). SDPs arise quite often in engineering disciplines, statistics, systems and control, signal processing etc. Roughly speaking, an SDP is the same as LP, except that the constraints are matrix inequalities instead of a set of scalar inequalities. Thus, SDP is an extension of linear programming (LP), where the component wise inequalities between vectors are replaced by matrix inequalities, or, equivalently the first orthant is replaced by the cone of positive semi-definite matrices. Semi-definite programming unifies several standard problems (e.g linear and quadratic programming) and finds many applications in engineering and combinatorial optimization. Although semi-definite programs are much more general than linear programs, they are not much harder to solve. Most interior- point-methods for linear programming have been generalized to semi-definite programs. As in linear programming, these methods have polynomial worst-case complexity and perform very well in practice. In control theory, they are very popular these days and for many NP-hard problems, SDPs can be used to obtain meaningful lower or upper bounds.

In a semi-definite program (SDP), we minimize a linear function of a variable $x \in R^m$ subject to a matrix inequality:

$$\text{minimize } c^T x$$

$$\text{subject to } F(x) \geq 0 \qquad \ldots (1.1)$$

where $$F(x) \cong F_0 + \sum_{i=1}^{m} x_i f_i$$

The problem data are the vector $c \in R^m$ and m + 1 symmetric matrices $F_0, \ldots, F_m \in R^{nxn}$. The inequality sign in $F(x) \geq 0$ means that $F(x)$ is positive semi-definite. i. e, $z^T F(x) z \geq 0$ for all $z \in R^n$. We call the inequality $F(x) \geq 0$ a linear matrix inequality (LMI).

Now, we let R^n denote n-dimensional Euclidean space, while $R^{m\,x\,n}$ denote the set of all m x n matrices with real entries, If S^n denote the set of all n x n real symmetric matrices, then $c.x \in s^n$, with the Frobenius matrices inner product of c and x defined as

$$c.x = Trace\,(c^T x)$$
$$= \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} \; x_{ij}$$

If we define a function $\|.\|_f : S^n \to R$ as $\|Q\|_f = (Q.Q)^{1/2}$, then $(S^n, \|.\|_f)$ is a normed linear space and the norm $\|.\|_f$ that defines the space is called the Frobenius norm. If for any $Q \in s^n$, $Q \geq 0$ implies that Q is positive semi-definite, while $Q > 0$ implies that Q is positive definite, then $S_+^n = \{Q \in s^n \mid Q \geq 0\}$, while $S_{++}^n = \{Q \in s^n \mid Q > 0\}$. Now, for $c,\ A \in S\ and\ b \in R^n,\ a\ SDP$ can be stated as

$$(P)\ \min imize \quad c\,.x$$
$$subject\ to\ A_1\, x = b_i, \quad i = 1, 2, ---, m \qquad \ldots (1.2)$$
$$S > 0$$

The dual (D) of the SDP (1.2) can be stated as

(D) maximize $b^T y$

$$\text{subject to } \sum_{i=1}^{m} A_i\, y_i \; + S = C \qquad \ldots (1.3)$$

where $y \in R^m$ and $S \in S_+^n$

Alizadeh (3) extended potential reduction method developed by *Ye* (8) for LP to SDP. This has led to the extension of many interior point (IP) methods to semi-definite programming.

[2] Primal-Dual Algorithm for SDP

Let $F^0(P) = \{x \in S^0 / A_i.x = b, x \geq 0\}$ be the interior of the feasible set for the primal problem (1.4) and $F^0(D)) = \left\{(s, y) \in S^0 x R^0 / \sum_{i=1}^{n} A_i\ y_i + S = C, S \geq 0\right\}$ be the interior of the feasible set for the dual problem (1.4), then, we assume the following:

(i) $F^0(P)xF^0(D) \neq 0$ and

(ii) the matrices $A_1, A_2, \ldots, A_n$ that defines the SDP (1.2) are linearly independent.
The set of primal-dual optimal solutions consist of the solutions $(x,\ S,\ y) \in S_+^n\ x\ S_{++}^n X\ R^n$ to the following system:

(I) (i) $A_i.x - b_i = 0,\ i = 1,\ 2, \ldots, m$

(ii) $\sum_{i=1}^{m} A_i y_i + S - C = 0$

(iii) $x.\ S = 0$

where the last equation I (iii) is the complementarity equation.

To symmetrize (I), Zhang (7) introduced a general symmetrization mapping:

(II) $H_p : R^{mxn} \to S^n$ defined as

$$H_p(m) = \frac{1}{2}\left[PMP^{-1} + (PMP^{-1})^T\right]$$

$\forall\ m \in R^{n\,x\,n}$, where $P \in R^{nxn}$ is some non-singular matrix .

Based on symmetric mapping (II), the system of equations (I) can be written as

(III) (i) $A_i.x - b_i = 0,\ i = 1,\ 2, \ldots, n$

(ii) $\sum_{i=1}^{n} A_i y_i + S - C = 0$

(iii) $H_p(x\,.\,S) = 0$

Consequently, the search direction $(\Delta\, x,\ \Delta S,\ \Delta y) \in S_+^n\ \ x\ S_{++}^n\ x\ R^n$ at a point $(x,\ S,\ y)$ is the solution of the following system:

(IV) (i) $A_i.\ \Delta x = b_i - A_i.x, \qquad i = 1,\ 2, \ldots, m$

(ii) $\sum_{i=1}^{m} A_i \Delta y_i + \Delta S = C - S - \sum_{i=1}^{m} A_i y_i$

(iii) $H_p(\Delta x S + \Delta) = \delta\mu I = H_p(x.\ s)$

where $\delta = [0, 1]$ is the centering parameter and $\mu = \frac{1}{n}(x.\, s)$

$$= \frac{1}{n}\left[\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij}\, S_{ij}\right]$$

and I is the identity matrix. If $d(x.\, s)$ denote the measure of centrality at a point $(x.\, s) \in S_{+}^{n}\ x\ S_{++}^{n}$, then

$$d\ (x.s) = \left\| x^{1/2} S x^{-1/2} - \mu I \right\|_f$$

$$= \left(\sum_{i=1}^{n} (\lambda_i [x.\, s] - \mu)^2\right)^{1/2}$$

where $\lambda_i \in [x.\, s]$ denote the eigen value of the matrix $(x\ .s)$. Based on this centrality condition, we define a Frobenius neighbourhood $N_p(\gamma)$ of the central path as follows:

$$N_f(\gamma) = \{(x,\ s,\ y) \in F^0(P)\ x\ F^{\ 0}(D) / d(x.\, s) \le \gamma\mu\}$$

where γ is a real constant in the interval (0, 1).

The algorithm that follows is a short-step path-following algorithm based on Monteiro – Zhang unified search direction obtained using equations IV (i), IV (ii) and IV (iii)

Algorithm:

Choose real constants γ and δ in (0, 1) such that

$$\frac{7\left(\gamma^2 + \delta^2\right)}{1-\gamma} \le \left[1 - \frac{\delta}{\sqrt{n}}\right]\gamma \qquad \text{. . . (1.4)}$$

$$\frac{\gamma\sqrt{8}}{1-\gamma} \le 1$$

and let $\sigma = \left[1 - \frac{\delta}{\sqrt{n}}\right]$

Let q be any positive integer and $\left(x^0,\ s^0,\ y^0\right) \in F^0(p)\ x\ F^{\ 0}(D)$ be an initial starting point satisfying the condition $(x^0, s^0) \le \gamma\mu_0$,

where $\mu_0 = \frac{x^0.s}{n}$

Repeat until convergence $\mu_k \le 2^{-q}\mu_0$ is reached.

(i) Choose a non-singular matrix $P^k \in R^{mxn}$

(ii) Compute search direction $\left(\Delta x^k, \Delta s^k, \Delta y^k\right)$ as the solution of system IV with $p = p_i\, .\mu_i$ and $(x,\ s,\ y) = \left(x^k, s^k, y^k\right)$

(iii) Obtain the next point $\left(x^{k+1}s^{k+1}, y^{k+1}\right)$ in the solution sequence as

(iv) $\left(x^{k+1}, s^{k+1}, y^{k+1}\right) = \left(x^k s^k, y^k\right) + \left(\Delta x^k, \Delta s^k, \Delta y^k\right)$

(v) Set $\mu_{k+1} = \left(\dfrac{x^{k+1}s^{k+1}}{n}\right)$ and increase k by 1

Theorem 1

If γ and δ are some fixed real constants such that $0 < \delta < 1$ and $0 < \gamma < 1/4$ satisfy the following inequalities:

$$\frac{7(\gamma^2 + \delta^2)}{1-\gamma} \cdot \frac{1}{\gamma} \le \left(1 - \frac{\delta}{\sqrt{n}}\right) \qquad \ldots (1.5)$$

then, the sequence of iterates $\{(x^k, s^k, y^k)\}$ generated by the algorithm in the neighbourhood

$$N_f(\gamma) = \{(x, s, y) \in F^6(p) \times F^0(D) : d(x.s) \le \gamma\mu\}$$

satisfies $\left(\dfrac{x^k s^k}{x^0 s^0}\right) \le (exp(-\delta))\dfrac{k}{\sqrt{n}}$

<u>Proof</u>

From Monteiro[6]

$$\frac{2\sqrt{2}}{1-\gamma} \le 1$$

$$\gamma\left[1 + 2\sqrt{2}\right] \le 1$$

$$\gamma \le \frac{1}{1+2\sqrt{2}} \approx 0.261$$

Now (i) any value of γ in the interval $(0, 1/4)$ will satisfy the condition of the theorem in Monteiro [6], since the interval $(0, 1/4)$ lies in the interval $\left(0, \dfrac{1}{1+2\sqrt{2}}\right)$.

(ii) the inequality (1.5) is equivalent to the inequality (1.6) in Monteiro [6]. Hence, all the results that are valid for the theorem in Monteiro [6] will also be valid for our own analysis.

Hence $\left(\dfrac{x^k.s^k}{x^0.s^0}\right) = \left(1 - \dfrac{\delta}{\sqrt{n}}\right)$

$$\ln \frac{x^k.s^k}{x^0.s^0} = k\ln\left(\frac{\delta}{\sqrt{n}}\right)$$

$$= k\left[\frac{-\delta}{\sqrt{n}} - \frac{1}{2}\left(\frac{\delta}{\sqrt{n}}\right)^2 + \frac{1}{3}\left(\frac{-\delta}{\sqrt{n}}\right)^3 - \ldots\right]$$

$$< \frac{k(-\delta)}{\sqrt{n}}$$

$$\left(\frac{x^k.s^k}{x^0.s^0}\right) < (\exp(-\delta))\frac{k}{\sqrt{n}}$$

Theorem 2

In at most $k_{\max} = \left\lceil \frac{q \ln 2\sqrt{n}}{\delta} \right\rceil$ iterations, the algorithm finds a solution to the SDP (1.2) with

$$\left(\frac{x^k . s^k}{x^0 . s^0} \right) < 2^{-q}$$

From Karmarkar [5],

$$\left(\frac{x^k . s^k}{x^0 , s^0} \right) < 2^{-q}, \ q \text{ is a positive integer}$$

$$\ln \left(\frac{x^k . s^k}{x^0 . s^0} \right) < -q \ln 2$$

$$\ln \left(x^k . s^k \right) - \ln \left(x^0 . s^0 \right) < -q \ln 2 \qquad ...(1.6)$$

From the inequality in theorem 1

$$\left(\frac{x^k . s^k}{x^0 . s^0} \right) < (\exp(-\delta)) \frac{k}{\sqrt{n}}$$

$$\ln \left(\frac{x^k . s^k}{x^0 . s^0} \right) < \ln \left[\exp(-\delta) \right]^{k/\sqrt{n}} \ [\ 7\ ]$$

$$\ln \left(x^k . s^k \right) - \ln \left(x^0 . s^0 \right) < \frac{k}{\sqrt{n}} \ln e^{-\delta}$$

$$\ln \left(x^k . s^k \right) - \ln \left(x^0 . s^0 \right) < \frac{-\delta k}{\sqrt{n}} \qquad ...(1.7)$$

Now, inequality (1.6) is true if k satisfies

$$\left(\frac{-k\delta}{\sqrt{n}} \right) \leq -q \ln 2$$

$$\frac{k\delta}{\sqrt{n}} \geq q \ln 2$$

$$k \geq \left\lceil \frac{q \ln 2\sqrt{n}}{\delta} \right\rceil$$

Now, as in Franklin [4] and Singh et al [7], we may define the number of iterations to find optimal solution as

$$k_{\max} \left\lceil \frac{q \ln 2\sqrt{n}}{\delta} \right\rceil$$

CONCLUSION

The algorithm will stop if the number of iterations k reaches $k_{\max}$. If the k reaches $k_{\max}$ before the convergence check $\left(\frac{x^k.s^k}{x^0.s^0}\right) < 2^{-q}$ is reached, we stop and conclude that the SDP (1.2) has no solution. The larger the value of δ in (0, 1) satisfying (1.5), the faster the convergence of the algorithm. Monteiro [6] proposed $\gamma = \delta = 1/20$ and with this choice, he obtained $k_{\max} = \left| 20q \ln 2\sqrt{n} \right|$. Adejo [1] proposed $\gamma = \delta = 1/12$ and obtained $k_{\max} = \left|12q \ln 2\sqrt{n}\right|$ with a termination rule which reduced at least 40% iterations in comparison to Monteiro's choice.
Adejo and Singh [2] chose $\delta = \frac{9}{10}$ and $\gamma = \frac{1}{12}$ as obtained $k_{\max} = \left| 1.1q \ln 2\sqrt{n} \right|$ which further reduces by 94.5% the number of iterations in comparison to that of Monteiro [6]. Adejo and Singh [2] termination rule was better than that achieved in Adejo [1] by $90.8\% \approx 91\%$. Here, our current termination rule on implementation sares 94.95% iterations over Monteiro [6] and with 8.18%. Slight improvement on the iterations over Adejo and Singh [3].

Justification

The larger the value of δ in (0,1) that satisfies (1.4), the faster the rate of convergence. Now, Monteiro [6] chose $\delta = 1/20,$ Adejo [1] chose $\delta = 1/12,$ Adejo and Singh chose $\delta = 9/10$ while here we choose $\delta = 99/100$.

Since $\delta = 99/100 > \delta = 9/10 > = 1/12 > \delta = 1/20$, it implies that our choice of $\delta = 99/100$ will ensure faster convergence than those of $\delta = 9/10$, $\delta\, 1/12$, $\delta = 1/20$.

Hence, for Monteiro [6], $\delta = 1/20$ with $K_{\max} = \left| 20\, q \ln 2\sqrt{n} \right|$

For Adejo [1], $\delta = 1/12$ with $K_{\max} = \left| 12\, q \ln 2\sqrt{n} \right|$

while, Adejo and Singh [2], $\delta = 9/10$ with $K_{\max} = \left| 1.1\, q \ln 2\sqrt{n} \right|$

Now, percentage (%) improvement of Adejo [1] over Monteiro [6]

$$= \left(\frac{20-12}{20}\right) x\, 100$$
$$= 40\%$$

% improvement of Adejo and singh [2] over Adejo [1]

$$= \left(\frac{20-1.1}{20}\right) x\, 100$$
$$= 94.5\%$$

% improvement of Adejo and Singh [2] over Adejo [1]

$$= \left(\frac{12 - 1.1}{12}\right) x\, 100$$

$$= 90.83\%$$

% improvement of Adejo and Ogala over Monteiro [6]

$$= \left(\frac{20 - 1.01}{20}\right) x\, 100$$

$$= 94.95\%$$

% improvement of Adejo and Ogala over Adejo and Singh [2]

$$= \left(\frac{1.1 - 1.01}{1.1}\right) x\, 100$$

$$= 8.18\%$$

REFERENCES

[1] Adejo, B. O. (**2004**): 'Modifications of Some Polynomial-time Algorithms for Linear Programming', Unpublished Ph.D thesis, Department of Mathematics, Ahmadu Bello University, Zaria

[2] Adejo, B. O. and Singh, D. (**2007**): *Nigerian Journal of Scientific Research* (NJSR), Vol 6, 21 – 23

[3] Alizadeh, F. (**1995**): *SIAM Journal of Optimization*, 5 : 13 – 51

[4] Franklin, J. (**1987**): ' Convergence in Karmarkar's Algorithm', *SIAM Journal of Numerical Analysis*, 24(4): 929 – 945

[5] Karmarkar, N. K. (**1984**): *Combinatorica* 4 : 373 – 395

[6] Monteiro, R. D. C. (**1997**): *SIAM Journal of Optimization*, 7 : 663 – 678

[7] Singh, J. N, Singh, D and Ranjan, Shashi (**1995**): *Journal of Information and Optimization Sciences*, 16 (3), 585 – 591

[8] Ye, Y. (**1990**): *SIAM Journal of Computing*, 19. 457 – 466

[9] Zhang, Y. (**1998**): *SIAM Journal of Optimization* 8: 365 - 386