Kybernetika

Maziar Salahi

An adaptive long step interior point algorithm for linear optimization

Kybernetika, Vol. 46 (2010), No. 4, 722--729

Persistent URL: http://dml.cz/dmlcz/140780

KYBERNETIKA — VOLUME *46* (2010), NUMBER 4, PAGES 722–729

AN ADAPTIVE LONG STEP INTERIOR POINT ALGORITHM FOR LINEAR OPTIMIZATION

Maziar Salahi

It is well known that a large neighborhood interior point algorithm for linear optimization performs much better in implementation than its small neighborhood counterparts. One of the key elements of interior point algorithms is how to update the barrier parameter. The main goal of this paper is to introduce an "adaptive" long step interior-point algorithm in a large neighborhood of central path using the classical logarithmic barrier function having $O(n \log \frac{(x^0)^T s^0}{\epsilon})$ iteration complexity analogous to the classical long step algorithms. Preliminary encouraging numerical results are reported.

Keywords: linear optimization, interior point methods, long step algorithms, large neighborhood, polynomial complexity

Classification: 90C05, 90C51

1. INTRODUCTION

In this paper we consider the following form of linear optimization problem:

$$\text{(P)} \qquad \min\{c^T x : Ax = b, x \geq 0\},$$

where $c \in R^n, b \in R^m, A \in R^{m \times n}$ and $rank(A) = m$ and its dual problem is given by

$$\text{(D)} \qquad \max\ \{b^T y : A^T y + s = c,\ s \geq 0\}.$$

Finding optimal solutions of (P) and (D) is equivalent to solve the following system:

$$\begin{aligned} Ax &= b, \quad x \geq 0, \\ A^T y + s &= c, \quad s \geq 0, \\ xs &= 0, \end{aligned} \tag{1}$$

where xs denotes the componentwise (Hadamard) product of the vectors x and s. In the primal-dual interior point methods (IPMs) the idea is to replace the third equation in (1) by the parametrized equation $xs = \mu e$, where e is the all one vector and μ is a positive scalar which is usually called the barrier parameter, namely we

have

$$\begin{aligned} Ax &= b, \quad x \geq 0, \\ A^T y + s &= c, \quad s \geq 0, \\ xs &= \mu e. \end{aligned} \tag{2}$$

Without loss of generality [4] we further assume that both (P) and (D) satisfy the interior point condition (IPC), i. e., there exists an (x^0, y^0, s^0) such that

$$Ax^0 = b,\ x^0 > 0, \qquad A^T y^0 + s^0 = c,\ s^0 > 0.$$

It is known that if the IPC holds, then system (2) has a unique solution for each $\mu > 0$. This solution, denoted by $(x(\mu), y(\mu), s(\mu))$, is called the μ-center of the primal-dual pair (P) and (D). The set of μ-centers gives *the central path* of (P) and (D) [6]. It has been shown that the limit of the central path (as μ goes to zero) exists and converges to primal and dual solutions when μ approaches zeros [4].

If we are given a feasible starting point for problems (P) and (D), the search directions within the primal-dual interior point algorithms which take us to the optimal solution are the solution of the following Newton system:

$$\begin{aligned} A\Delta x &= 0, \\ A^T \Delta y + \Delta s &= 0, \\ s\Delta x + x\Delta s &= \mu e - xs, \end{aligned} \tag{3}$$

where μ is a positive scalar, which is called the barrier parameter.

It is well known that long step interior point algorithms perform better than small step algorithms in practice [4, 7]. In the classical interior point algorithms, the barrier parameter μ is taken a fraction of $\mu_g := \frac{x^T s}{n}$ [7]. However from practical point of view, this might not be the best choice. For example, Mehrotra in his predictor-corrector algorithm has suggested a heuristic which determines μ based on predictor step's information [1]. In this paper, our main goal is to give an adaptive strategy of choosing μ based on the position of the current iterate in classical long step algorithm. This is achieved by solving a one dimensional equation depending on the position of the current iterate. Moreover, we keep all the iterates of the algorithm in the so called negative infinity norm neighborhood, the widely used large neighborhood, as follows:

$$\mathcal{N}_\infty^-(\gamma) := \left\{(x, y, s) \in \mathcal{F}^0 :\ x_i s_i \geq \gamma \mu_g\ \forall i = 1, \cdots, n\right\}, \tag{4}$$

where $\gamma \in (0, 1)$ is a constant independent of n, μ_g, and $\mathcal{F}^0$ denotes the interior of the primal and dual feasible regions. Obviously when γ is closer to zero then this neighborhood spreads almost all over the feasible region [3, 4, 7, 8].

The following technical lemma is useful in the derivation of a lower bound for the maximum step size in the computed search direction, which will be discussed later in this paper.

Lemma 1.1. Let $(\Delta x, \Delta y, \Delta s)$ be the solution of (3), then we have

$$||\Delta x \Delta s||_\infty^- \leq \frac{1}{4}\left(\frac{n\mu^2}{\gamma\mu_g} + n\mu_g - 2n\mu\right),$$

where $||z||_\infty^- := ||z^-||_\infty,\ (z^-)_j = \min\{z_j, 0\}$.

P r o o f. See [2]. □

2. ADAPTIVE LONG STEP ALGORITHM

To introduce our adaptive algorithm, we use the classical logarithmic barrier proximity measure as follows:

$$\Phi(x, s, \mu) := \frac{x^T s}{2\mu} - \frac{n}{2} + \frac{n}{2}\log\mu - \frac{1}{2}\sum_{i=1}^{n}\log(x_i s_i). \tag{5}$$

Lemma 2.1. As a function of μ, the global minimum of (5) attains at $\mu = \mu_g$.

P r o o f. We have

$$\Phi'_\mu = -\frac{x^T s}{2\mu^2} + \frac{n}{2\mu}.$$

It is obvious that $\Phi'(\mu_g) = 0$ and $\Phi'(\mu) > 0$ for $\mu > \mu_g$ and $\Phi'(\mu) < 0$ for $\mu < \mu_g$. Thus μ_g is its global minimizer. □

For simplicity the geometric mean of the vector xs is denoted by μ_h i. e.,

$$\mu_h = (x_1 s_1 \cdots x_n s_n)^{\frac{1}{n}},$$

which obviously one has $\mu_h \leq \mu_g$. Let us rewrite $\Phi(x, s, \mu)$ as

$$\Phi(x, s, \mu) = \frac{n}{2}\left(\frac{\mu_g}{\mu} - 1 + \log\frac{\mu}{\mu_h}\right).$$

Our goal is to have μ as an adaptive fraction of μ_g. To achieve this goal and for simplicity, we assume

$$\frac{\mu_g}{\mu} - 1 + \log\frac{\mu}{\mu_h} = \tau - 1$$

or

$$\frac{\mu_g}{\mu} + \log\frac{\mu}{\mu_h} - \tau = 0, \tag{6}$$

where $\tau > 1$ is a given constant. Thus the barrier parameter at each iteration is chosen as one of the solutions of (6). In the following lemma we give a condition under which equation (6) has two positive solutions. The smaller one is chosen as the μ value and is denoted by μ_t.

Lemma 2.2. For all $(x, s) \in R^n_{++} \times R^n_{++}$ for which $\mu_g \leq \tau\mu_h$, equation (6) has two positive solutions, one is smaller than μ_g and the other one is greater than μ_g.

P r o o f. Let us first assume that $\mu_g = \tau\mu_h$. Then from equation (6) one has $\mu_t = \mu_h$ which is obviously less than μ_g . Now since Φ is a strictly decreasing function of μ for $\mu < \mu_g$, thus $\Phi(x, s, \mu_g) < \frac{(\tau-1)n}{2}$. Furthermore, since Φ is a strictly increasing function of μ for $\mu > \mu_g$, then (6) has another solution that is greater than μ_g. Now let us assume that $\mu_g = \tau_1\mu_h$, where $1 \leq \tau_1 < \tau$. Obviously

$$\Phi(x, s, \mu_h) = \frac{(\tau_1 - 1)n}{2} < \frac{(\tau - 1)n}{2}.$$

Moreover we also know that the value of the proximity measure goes to infinity when μ approaches zero. Thus (6) has a solution which is strictly less than μ_h. Analogous to the previous case, it must have another solution which is greater than μ_g. □

The following technical lemma plays a crucial role in our future analysis.

Lemma 2.3. For any $(x, s) \in R^n_{++} \times R^n_{++}$ for which $\mu_g \leq \tau\mu_h$, one has $\tau \leq \frac{\mu_g}{\mu_t} \leq 2\tau$.

P r o o f. Since $\mu_g \leq \tau\mu_h$, then $\Phi(x, s, \mu_h) \leq \frac{(\tau-1)n}{2}$. This together with the fact that Φ for $\mu < \mu_g$ is strictly decreasing imply that $\mu_t \leq \mu_h$ and from (6) we have $\mu_g \geq \tau\mu_t$. Now to prove the other side, let $\mu_g = \tau_1\mu_h$, where $1 \leq \tau_1 \leq \tau$. Then (6) becomes $\frac{\mu_g}{\mu_t} + \log\tau_1 - \log\frac{\mu_g}{\mu_t} - \tau = 0$. Now this as a function of $\frac{\mu_g}{\mu_t}$ is convex and has two roots that are less than 2τ. □

Now we outline the algorithm using our new adaptive updating strategy:

Adaptive Algorithm.

A neighborhood parameter $\tau > 1$;
An accuracy parameter $\epsilon > 0$;
$(x^0, y^0, s^0) \in \mathcal{N}^-_\infty(\gamma)$ with $\gamma = \frac{1}{\tau}$.
while $x^T s \geq \epsilon$
Solve (3) with $\mu = \mu_t$, the smaller positive solution
of (6) and compute the maximum step size α_c
such that $(x(\alpha_c), y(\alpha_c), s(\alpha_c)) \in \mathcal{N}^-_\infty(\gamma)$;
Set $(x(\alpha_c), y(\alpha_c), s(\alpha_c)) = (x + \alpha_c\Delta x, y + \alpha_c\Delta y, s + \alpha_c\Delta s)$.
end

In the following lemma we show that for any iterate of the Adaptive Algorithm, equation (6) always has two positive solutions.

Lemma 2.4. Let (x, y, z), the current iterate of Adaptive Algorithm, be in $\mathcal{N}^-_\infty(\gamma)$. Then

$$\mu_g \leq \tau\mu_h,$$

where $\tau = \frac{1}{\gamma}$.

P r o o f. For any $(x, y, s) \in \mathcal{N}_\infty^-(\gamma)$ we have $x_i s_i \geq \gamma\mu_g \ \forall i = 1, \cdots, n$. This implies $\mu_h \geq \gamma\mu_g$. Now since $\tau = \frac{1}{\gamma}$, then we have $\mu_g \leq \tau\mu_h$. □

Corollary 2.5. For all (x, y, s) generated by Adaptive Algorithm, equation (6) has two positive solutions.

P r o o f. It follows from the previous lemma and Lemma 2.2. □

The following corollary follows from Lemmas 1.1 and 2.3 that is used in the next theorem.

Corollary 2.6. Let μ_t be the smaller positive solution of (6) for $(x, y, s) \in \mathcal{N}_\infty^-(\gamma)$. Then

$$||\Delta x \Delta s||_\infty^- \leq \frac{n\mu_g}{4}.$$

Theorem 2.7. Suppose that (x, y, s), the current iterate of Adaptive Algorithm, belong to $\mathcal{N}_\infty^-(\gamma)$ with $\gamma = \frac{1}{\tau}, \tau > 1$ and $(\Delta x, \Delta y, \Delta s)$ be the solution of (3) with $\mu = \mu_t$ as the smaller positive solution of (6). Then the maximum step size $\alpha_{\rm c}$, that keeps $(x(\alpha_{\rm c}), y(\alpha_{\rm c}), s(\alpha_{\rm c}))$ in $\mathcal{N}_\infty^-(\gamma)$, satisfies

$$\alpha_{\rm c} \geq \frac{2(\tau - 1)}{\tau^2 n}.$$

P r o o f. The goal is to find the maximum nonnegative α for which the relation $x_i(\alpha)s_i(\alpha) \geq \gamma\mu_g(\alpha), \ \ \forall i = 1, \cdots, n$. We have

$$\begin{aligned} x_i(\alpha)s_i(\alpha) &= x_i s_i + \alpha(\mu_t - x_i s_i) + \alpha^2 \Delta x_i \Delta s_i \\ &\geq (1 - \alpha)x_i s_i + \alpha\mu_t - \alpha^2 ||\Delta x \Delta s||_\infty^- \\ &\geq (1 - \alpha)\gamma\mu_g + \alpha\mu_t - \alpha^2 \frac{n\mu_g}{4}, \end{aligned}$$

where the second inequality follows from the fact that $(x, y, s) \in \mathcal{N}_\infty^-(\gamma)$ and the previous corollary. Moreover

$$\mu_g(\alpha) := \frac{(x(\alpha))^T s(\alpha)}{n} = \frac{(x + \alpha\Delta x)^T (s + \alpha\Delta s)}{n} = \left(1 - \alpha + \alpha\frac{\mu_t}{\mu_g}\right)\mu_g,$$

where the last equality follows from the orthogonality of Δx and Δs and the last equation of (3) with $\mu = \mu_t$.

In order to keep the next iterate in $\mathcal{N}_\infty^-(\gamma)$, one has to have

$$(1 - \alpha)\gamma\mu_g + \alpha\mu_t - \alpha^2 \frac{n\mu_g}{4} \geq \gamma\left(1 - \alpha + \alpha\frac{\mu_t}{\mu_g}\right)\mu_g,$$

which simplifies to

$$(1 - \gamma)\mu_t \geq \alpha\frac{n\mu_g}{4}.$$

Using Lemmas 2.3 and 2.4, it definitely holds for $\alpha = \frac{2(\tau-1)}{\tau^2 n}$, which completes the proof. □

Theorem 2.8. There exists

$$K = O\left(n \log \frac{(x^0)^T s^0}{\epsilon}\right)$$

such that

$$(x^k)^T s^k \leq \epsilon \quad \forall k \geq K.$$

P r o o f. After each iteration in the direction generated by system (3) with $\mu = \mu_t$ and $\alpha = \alpha_c$ one has

$$\mu_g(\alpha_c) = \left(1 - \alpha_c - \alpha_c \frac{\mu_t}{\mu_g}\right) \mu_g.$$

Now using Lemma 2.3 and the previous theorem it follows that

$$\mu_g(\alpha_c) \leq \left(1 - \alpha_c - \alpha_c \frac{1}{2\tau}\right) \mu_g = \left(1 - \alpha_c(\frac{2\tau + 1}{2\tau})\right) \mu_g.$$

Thus after k iterations we have

$$(x^k)^T s^k \leq \left(1 - \frac{\delta}{n}\right)^k (x^0)^T s^0,$$

where $\delta = \frac{(\tau-1)(2\tau+1)}{\tau^3}$. Now to have $(x^k)^T s^k \leq \epsilon$ it is sufficient to have

$$\left(1 - \frac{\delta}{n}\right)^k (x^0)^T s^0 \leq \epsilon,$$

or

$$k \log\left(1 - \frac{\delta}{n}\right) \leq \log \frac{\epsilon}{(x^0)^T s^0}.$$

Since $-\log(1 - t) \geq t$, we have

$$k \frac{\delta}{n} \geq \log \frac{(x^0)^T s^0}{\epsilon}$$

or

$$k \geq \frac{n}{\delta} \log \frac{(x^0)^T s^0}{\epsilon},$$

which completes the proof. □

3. NUMERICAL EXPERIMENTS

In this section, we present numerical results for several examples taken from the standard test library NETLIB for linear optimization problems. We have compared our Adaptive Algorithm with the classical long step algorithm which considers $\mu = 0.1\mu_g$ at each iteration. For all test problems Adaptive Algorithm uses $\tau = 5$ (Our computational experiments show this is the best choice, and other choices are sometimes better and sometimes worse than classical approach up to one or two iterations). For the rest of test problems in NETLIB both algorithms perform the same, so we omitted them in Table. As we see our Adaptive Algorithm is always better than

the classical approach. As we know Mehrotra's predictor-corrector algorithm is the widely used algorithm in implementation of interior point algorithms. It uses an adaptive heuristic to update the barrier parameter at each iteration. Our preliminary numerical experiments on this algorithm show the efficiency of our adaptive approach on Mehrotra's heuristic [1].

Table. Comparison of Adaptive and Classical Algorithms.

Problem	Adaptive Algorithm	Classical Algorithm
25fv47	43	44
agg	34	35
agg2	31	32
blend	19	20
bnl1	45	46
boeing1	37	38
boeing2	32	34
bore3d	29	32
capri	32	33
cycle	54	>100
perold	61	64
pilot4	62	70
pilotja	60	64
scfxm1	31	32
sc105	16	17
stocfor1	23	25

4. CONCLUDING REMARKS

In this paper a simple adaptive long step algorithm in a large neighborhood of the central path, using the classical logarithmic barrier function, is introduced. It is proved that it enjoys the same order of polynomial iteration complexity as the classical long step algorithms, while taking advantages of adaptive choice of the barrier parameter at each iteration. Finally, preliminary encouraging numerical results are reported.

(Received January 21, 2010)

REFERENCES

[1] S. Mehrotra: On the implementation of a (primal-dual) interior point method. SIAM J. Optim. *2* (1992), 575–601.

[2] S. Mizuno, M. J. Todd, and Y. Ye: On adaptive-step primal-dual interior-point algorithms for linear programming. Math. Oper. Res. *18* (1993), 4, 964–981.

[3] F. A. Potra: A superliner convergent predictor–corrector method for degenerate LCP in a wide neighborhood of the central path with $O(\sqrt{n}L)$-iteration complexity. Math. Program. Ser. A *100* (2004), 2, 317–337.

[4] C. Roos, T. Terlaky, and J. P. Vial: Interior Ooint Algorithms for Linear Optimization. Second edition. Springer Science, 2005.

[5] M. Salahi and T. Terlaky: A hybrid adaptive algorithm for linear optimization. Asia-Pacific J. Oper. Res. *26* (2009), 2, 235–256.

[6] G. Sonnevend: An "analytic center" for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming. In: Proc. 12th IFIP Conference System Modeling and Optimization (A. Prékopa, J. Szelezsán, and B. Strazicky, eds.), Budapest 1985. Lecture Notes in Control and Information Sciences, pp. 866–876. Springer Verlag, Berlin, 1986.

[7] S. J. Wright: Primal-dual Interior-point Methods. SIAM, Philadelphia 1997.

[8] G. Zhao: Interior–point algorithms for linear complementarity problems based on large neighborhoods of the central path. SIAM J. Optim. *8* (1998), 397-413.

Maziar Salahi, Department of Mathematics, Faculty of Sciences, University of Guilan, Namjoo Street, Rasht. Iran.

e-mail: salahim@guilan.ac.ir, salahi.maziar @gmail.com