A Higher Order Derivative Model of the Barrier Function for Linear Programming

Lu Zhihong* and Wei Shouping*

Abstract

In this paper, an interior point approach is presented for linear programming problems by using the logarithmic barrier function method, which makes use of information on higher derivatives of the barrier function to explore search directions. The corresponding algorithm is derived, and can produce feasible successive iterations that have global convergence. The computational results indicate that the suggested method seems better than the standard Newton methods.

1. Introduction

The modern era of interior point method dates to 1984, when Karmarkar first proposed his algorithm for linear programming problems[1]. Since then, various interior point methods have been considered and developed one way or another as an alternative to the simplex methods. Logarithmic barrier function method is a particular choice among many interior point methods due to its good performances and "self concordance" property[2,3,4]. Performance of such algorithms tends to be closely related to the careful iterative reduction of a barrier parameter and the computation of the search direction based upon the quadratic programming subproblem. However, making use of information on higher order derivatives of logarithmic barrier function to construct interior point method, which generate better search directions than the traditional technique, such as the standard Newton's method, maybe an attractive alternative[6,7,8]. Therefore, it is very important to use the higher order interior point method to study linear programming problems.

The outline of the paper is as follows. In Section 2, a brief review of interior point method that uses a second order approximation is described for linear programming problems. A barrier function method that takes advantage of the high order derivatives is proposed and its global convergence is discussed in Section 3. Section 4 gives numerical experiments and indicates that the method proposed seems promising and reliable.

2. Interior point methods for linear programming problems

Consider the following linear programming problem in the standard form:

$$(\boldsymbol{LP}) \quad \begin{cases} \min\ f(x) = c^T x \\ s.t.\ \ Ax = b \\ \qquad x \geq 0 \end{cases} \tag{1}$$

where $c, x \in R^n$, $A \in R^{m\times n}$ and $b \in R^m$.

We make the basic assumption that the matrix A has full row rank. A point x is said to be an interior point if $x \in \Omega_{int} \stackrel{def}{=} \left\{ x \in R^n \middle| Ax = b, x > 0 \right\}$.
Assuming that Ω_{int} is nonempty.

For $\forall x \in \Omega_{int}$, by using the logarithmic barrier function method, we obtain the barrier- programming problem associated with (LP):

$$\text{(BP)} \quad \begin{cases} \min\ B(x,\mu_k) = c^T x - \mu_k \sum_{i=1}^{n} \log(x_i) \\ s.t.\ Ax = b \end{cases} \tag{2}$$

where $\log$ denotes the natural logarithm and $\mu_k > 0$ denotes the barrier parameter.

Because the logarithmic barrier function $B(x,\mu_k)$ requires its arguments to be positive, the solution x of (BP) must belong to interior point region Ω_{int}. It is well known[5] that for any sequence $\{\mu_k\}$ with $\mu_k \downarrow 0$, all limit points of $\{x(\mu_k)\}$ are solutions of (LP).

*Hydroelectric & Digital Engineering Inst., HuaZhong Univ.of Sci.&Tech., Wuhan 430074, China

To achieve an approximate solution of (BP) for a certain value μ of μ_k, we solve the following quadratic subproblem at the kth iteration

$$\begin{cases} \min \nabla B^T d + \frac{1}{2} d^T (\nabla^2 B) d \\ s.t.\ Ad = 0 \end{cases} \quad (3)$$

where $\nabla B = \nabla B(x,\mu) = c - \mu X^{-1} e$, $\nabla^2 B = \nabla^2 B(x,\mu) = \mu X^{-2}$, $X = diag\left(x^1, x^2, \cdots, x^n\right)$, $x^i \ (i = 1,2,\cdots,n)$ is the composition of the vector x, $e = (1,\cdots,1)^T \in R^n$.

The first order Karush Kuhn Tucker (KKT) conditions for (BP) are then (or the optimization conditions for problem (3)

$$\begin{pmatrix} \nabla^2 B & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} d \\ -\lambda \end{pmatrix} = -\begin{pmatrix} \nabla B \\ 0 \end{pmatrix} \quad (4)$$

where $\lambda \in R^m$ is the multiplier vector for the equation constraint of the problem (3).

The prototype barrier algorithm for (LP) can be specified as follows:

Algorithm 1 —The second order derivative model

Step1: Given $x_0 \in \Omega_{int}$ and $\mu_0 > 0$, set $k := 0$.

Step2: If some termination test is satisfied, then stop; else go to Step 3.

Step3: Calculate search directions d_k from (4) by performing one or more Newton steps, starting at $x = x_k$ and fixing $\mu = \mu_k$.

Step4: Perform the line search to determine a step size α_k such that

$$f(x_k + \alpha_k d_k) \le f(x_k),\ x_k + \alpha_k d_k \in \Omega_{int}.$$

Step5: Choose $\mu_{k+1} \in (0, \mu_k)$, set $k := k+1$ and go to Step 2.

The various algorithms that use above framework differ in the way that they choose the starting point, the barrier parameter μ_k, and the step size α_k. For instance, short -step algorithms take a single Newton step at each of the iterations, while long-step algorithms use more than one Newton step for each μ_k. A comprehensive review of interior point methods about step length and diversified search strategies is given and discussed by Wright [5]. In addition, making use of higher order interior point method to obtain solutions of the optimization problems is another efficient approach, which seems to offer the computational advantage claimed and obtains fast convergence [See 6,7,8]. In the next section, a logarithmic function interior method that makes use of information on higher order derivatives is considered and some properties of algorithm given are described.

3. Analysis and algorithm

Several variants of the higher order interior point methods have been developed during the past few years [6,7,8]. Usually, they differ in how to choose schemes for incorporating higher order information about the system (4). In this section, we utilize the higher order derivations of the logarithmic barrier function B (x, μ) in (BP) to construct interior point method, that is, by using a truncated Taylor series expansion with the third order accuracy for barrier function $B(x,\mu)$, search directions d of (BP) from a point $x \in \Omega_{int}$ is obtained by solving the following optimization problem

$$\begin{cases} \min \phi(x,\mu) = \nabla B^T d + \frac{1}{2} d^T (\nabla^2 B) d - \frac{\mu}{3} \sum_{i=1}^{n} \frac{(d^{(i)})^3}{(x^{(i)})^3} \\ s.t.\ Ad = 0 \end{cases}$$

…..(5)

where $d^{(i)}$ and $x^{(i)}$ are the ith component of the vector d and x, respectively.

The Lagrangian function of problem (5) can be computed as

$$L = \nabla B^T d + \frac{1}{2} d^T (\nabla^2 B) d - \frac{\mu}{3} \sum_{i=1}^{n} \frac{(d^{(i)})^3}{(x^{(i)})^3} - \lambda^T A d$$

Hence

$$\frac{\partial L}{\partial d} = \nabla B + \nabla^2 Bd - \mu X^{-3} Dd - A^T \lambda = 0 \quad (6)$$

$$\frac{\partial L}{\partial \lambda} = A^T d = 0 \quad (7)$$

where $D = diag\left(d^{(1)}, d^{(2)}, \cdots, d^{(n)}\right)$.

Multiplying both sides of equation (6) by d^T, and taking account equation (7), we get

$$\begin{aligned} d^T \nabla B &= -d^T \nabla^2 Bd + \mu d^T X^{-3} Dd \\ &= -d^T \nabla^2 B(I - X^{-1} D)d \end{aligned} \quad (8)$$

It is easy to know from (8), we have

Theorem 1. If $d_k{}^T \nabla^2 B_k (I - X_k{}^{-1} D_k) d_k > 0$, then d_k is a decent direction of (BP) at x_k.

If the appropriate search direction d_k has been found, we would also to know how far (=α_k) we should travel along the direction d_k. A simple approach is as follows

$$\begin{aligned} \alpha_k &= 0.995\left[arg \max\left\{\alpha \in (0,1) \middle| x_k + \alpha d_k > 0\right\}\right] \\ &= 0.995 \min\left\{\left(1, -\frac{x_k^{(i)}}{d_k^{(i)}}\right) \middle| d_k^{(i)} < 0\right\}. \end{aligned} \quad (9)$$

It should be noted here that if x_0 is such that $Ax_0 = b$ and $x_0 > 0$, then $Ax_k = b$ and $x_k > 0$ for every k.

We now state our algorithm which uses the higher derivative information for (BP).

<u>Algorithm 2</u> —The higher order derivative model for (BP)

Step 1: Given $x_0 \in \Omega_{int}$ and $\mu_0 > 0$, set $k := 0$.

Step 2: If some termination test is satisfied, then stop; else go to Step 3.

Step 3: Calculate problem (5) with $x = x_k$ and fixed $\mu = \mu_k$ to obtain d_k.

Step 4: Compute the step length α_k by (9) and choose some fixed σ, if α_k satisfying

$$B(x_k, \mu_k) - B(x_k + \alpha_k d_k, \mu_k) \geq -\sigma \alpha_k d_k^T \nabla B(x_k, \mu_k), \quad (10)$$

then go to Step 5; else set $\alpha_k := \alpha_k / 2$ and go to Step 4.

Step 5: Set $x_{k+1} = x_k + \alpha_k d_k,\ \mu_{k+1} = \beta \mu_k$, $k := k+1$ and go to Step 2.

Next, we give some properties about the above algorithm 2 as follows.

<u>Lemma 1</u>. Let $x_k \to x^* > 0$ and $\nabla^2 B_k (I - X_k{}^{-1} D_k) \to H^* > 0$, if x^* is not a KKT point of (BP), then $\bar{\alpha} = inf\{\alpha_k\} > 0$, where α_k is the step size that satisfies inequation (10).

<u>Proof</u>. The proof is straightforward. □

Theorem 2. When the algorithm 2 is used to solve (BP) from any initial point $x_0 \in \Omega_{int}$, it either stops at a KKT point, say x_k, in finite iterations, or generates an infinite sequence {x_k} such that $x_k \to x^* > 0$, $\nabla^2 B_k (I - X_k{}^{-1} D_k) \to H^* > 0$, then each cluster point of {x_k} is a KKT point.

<u>Proof</u>. If the algorithm terminates at x_k in finite steps, then the first part of the theorem is obvious. Therefore, without loss of generality, assuming that an infinite sequence {x_k} is generated. Now, suppose that x^* is not a KKT point of (BP), then from (8) and the algorithm 2, we have

$$\begin{aligned} B(x_k, \mu_k) - B(x_k + \alpha_k d_k, \mu_k) &\geq -\sigma \alpha_k d_k^T \nabla B \\ &= \sigma \alpha_k d_k^T \nabla^2 B(I - X_k^{-1} D_k) d_k \geq 0 \end{aligned} \quad (11)$$

Hence,

$$B(x_k, \mu_k) \geq B(x_k + \alpha_k d_k, \mu_k). \quad (12)$$

Since $B(x_k, \mu_k)$ is bounded, for all sufficiently large k we have

$$\{B(x_k, \mu_k) - B(x_k + \alpha_k d_k, \mu_k)\} \to 0 \quad (13)$$

From (11) we obtain

$$\sigma\alpha_k d_k^T \nabla^2 B(I - X_k^{-1} D_k) d_k \to 0. \quad (14)$$

By $\nabla^2 B_k (I - X_k{}^{-1} D_k) \to H^* > 0$ and from the lemma 1(σ is a positive parameter), We can get $d_k \to d^* = 0$. (15)

Thus, from (15) and taking limit each sides of (6) and (7) respectively, we obtain

$$\begin{cases} \nabla B(x^*, \mu) - A^T \lambda = 0, \\ A^T d^* = 0. \end{cases} \quad (16)$$

It would follows that x^* is a KKT point for (BP). This is a contradiction and hence we complete the proof. □

Now, we describe the higher order interior point algorithm for (LP).

Algorithm 3

Step 1: Given $x_0 \in \Omega_{int}$ and $\mu_0 > 0$, set $k := 0$.

Step 2: If some termination test is satisfied, then stop; else go to Step 3.

Step 3: Calculate problem (5) with $x = x_k$ and fixed $\mu = \mu_k$ to obtain d_k.

If $d^T \nabla B < 0$, then go to Step 5; Otherwise, go to Step 4.

Step 4: Calculate problem (3) with $x = x_k$ and fixed $\mu = \mu_k$ to obtain d_k.

Step 5: Compute the step length α_k by (9) and choose some fixed σ, if α_k satisfying

$$B(x_k, \mu_k) - B(x_k + \alpha_k d_k, \mu_k) \geq -\sigma\alpha_k d_k^T \nabla B(x_k, \mu_k),$$

then go to Step 6; Otherwise, set $\alpha_k := \alpha_k / 2$ and go to Step 5.

Step 6:

Set $x_{k+1} = x_k + \alpha_k d_k$, $\mu_{k+1} = \beta\mu_k$ (β is a positive parameter).

Step 7: Set $k := k+1$ and go to Step 2.

4. Numerical experiments

In this section, we present two simple example and some medium-scale examples to illustrate the application of the suggested higher order interior point algorithm 3 in Section 3. For purposes of comparison, Algorithm 1 is also applied to these examples. We chose the algorithmic parameters as follows. The tolerance error $\varepsilon = 10^{-8}$, $\mu_0 = 0.9$, $\beta = 0.15$, $\sigma = 0.35$. Let the stopping criterion of the Algorithm 1 and the Algorithm 3 be $\|d_k\| \leq \varepsilon$. The method 1 and the method 2 refer to as the second derivative interior point method (Algorithm 1) and the higher order interior point method (Algorithm 3), respectively.

Example 1:

$$\min \; f(X) = 3x_1 + 2x_2 + x_3 + 4x_4$$

$$s.t. \; \begin{cases} 2x_1 + 4x_2 + 5x_3 - x_5 = 230 \\ 3x_1 - x_2 + 7x_3 - 2x_4 - x_6 = 46 \\ 5x_1 + 2x_2 + x_3 + 6x_4 - x_7 = 345 \\ x_i \geq 0, i = 1,2,\cdots,7 \end{cases}$$

Table 1: Computational results for Example 1(with staring point $X = (50,2,100,10,378,782,69)$)

Variables	Exact solution	Method 1	Method 2
x_1	65	64.9999941	65.0000016
x_2	0	0.0000001	0
x_3	20	20.0000013	20.0000008
x_4	0	0	0
x_5	0	0	0
x_6	289	289.0000373	289.0000065
x_7	0	0	0
f (X)	215	214.9999838	215.0000056

Example 2:

$$\min \; f(X) = -x_1 + 3x_2 + 3x_3 + 2x_4 + 4x_5 + 2x_6 + 2x_7 + 5x_8 + x_9 - 4x_{10}$$

$$s.t. \; \begin{cases} 3x_1 + 2x_2 - 5x_3 + 3x_4 + 8x_5 - 7x_6 + 3x_7 + 6x_8 - 4x_9 - 9x_{10} = 0 \\ 2x_1 + 3x_2 - 9x_4 + 4x_5 + 3x_6 - x_7 + 9x_8 - 5x_9 - 6x_{10} = 0 \\ -3x_1 + 10x_2 - 2x_3 + x_4 - x_5 - 4x_6 + 3x_7 - 2x_8 + 6x_9 - 8x_{10} = 0 \\ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} = 1 \\ x_i \geq 0, i = 1,\cdots,10 \end{cases}$$

Example 3:

The medium-scale examples are used to evaluate the numerical performance in Table 3, where the m, n columns give the number of equality constraints, the dimension (number of variables), respectively. All the components of coefficient matrix A and the vector c of (LP) are uniformly distributed random numbers from [0,1]. Let $b = Ae$, that is, $X^0 = e$ can be used as starting point.

Table 2: Computational results for Example 2(with starting point $X = (0.1, 0.1, \cdots, 0.1)$

Variables	Exact solution	Method 1	Method 2
x_1	316/614	0.5146574	0.5146576
x_2	119/614	0.1938100	0.1938111
x_3	0	0	0
x_4	0	0	0
x_5	0	0	0
x_6	0	0	0
x_7	0	0	0
x_8	0	0	0
x_9	85/614	0.1384367	0.1384364
x_{10}	94/614	0.1530958	0.1530946
f (X)	-0.4071661	-0.4071739	-0.4071663

Table 3: Computational results for medium-scale examples

Problem size		Method 1		Method 2	
M	n	Iterations	time(s)	Iterations	time(s)
20	30	21	0.085	18	0.035
50	80	23	0.209	19	0.086
100	120	28	0.614	19	0.325

From Table 1 and Table 2, we may note that the new approach found better solutions than those using the second derivative interior point method. From Table 3, the iteration counts is relatively small, and the computational time is fast, then it is advisable to use the higher order logarithmic barrier method for the larger linear programming problems.

5. Conclusions

Logarithm barrier function method is an appropriate choice among several interior point methods and is becoming more and more popular due to its better performance and "self concordance" property. In this paper, a method based on higher order interior point algorithm for linear programming problems is presented and a global convergence for this algorithm has been proved. The computational results indicate that our method is very promising and efficient.

References

1. N.K.Karmarkar. A new polynomial time algorithm for linear programming. *Combinatorica*, 1984 (4), 373-395.
2. Gill P.E., Murray W., Saunders M.A., Tomlin J.A. and Wright M.H. On projected Newton barrier methods for linear programming and equivalence to Karmarkar's projective method. *Mathematical programming*, 1986(36), 183-209.
3. R.Cominetti, J.San Martin. Asymptotic analysis of the exponential penalty trajectory in linear programming. *Mathematical programming*, 1994(67), 169-187.
4. O.Guller, L.Tuncel. Characterization of the barrier parameter of homogeneous convex cones. *Mathematical programming*, 1998(81), 55_76.
5. M.H.Wright. *Interior methods for constrained optimization*, in: Acta numer.1992, Cambridge University press, Cambridge, 1992, 341-407.
6. J.Lustig, R.E.Marsten, D.F.Shanno. On implementing Mehrotra's predictor-corrector interior point method for linear programming. *SIAM Journal on Optimization*, 1992(2), 435-449.
7. Y.Ye. Interior point algorithms: Theory and Analysis. Wiley-Interscience Series in *Discrete Mathematics and Optimization*, Wiley, New York, 1997.
8. J.Gondzio. Multiple centrality

corrections in a primal-dual method for linear programming. *Computational Optimization and Applications*, 1996(6), 137-156.