Journal of Information & Optimization Sciences
ISSN 0252-2667 (Print), ISSN 2169-0103 (Online)
Vol. 45 (2024), No. 3, pp. 677–685
DOI : 10.47974/JIOS-1340

A penalty method for linear programming

Linda Menniche *
Department of Mathematics
LMPA Laboratory
Mohamed Seddik Ben Yahia University
BP 98, Ouled Aissa
Jijel 18000
Algeria

Djamel Benterki †
Bachir Merikhi §
Department of Mathematics
Laboratory of Fundamental and Numerical Mathematics (LMFN)
Setif1 University-Ferhat Abbas
19000 Setif
Algeria

Abstract

A logarithmic penalty method for linear optimization problem with a new approximate function is analyzed. The function addressed makes it possible to offer a displacement-step without major difficulties.We consider some numerical results which show the superiority of this approach versus line search methods.

Subject Classification: *(2010) 90C22, 90C51.*

Keywords: *Linear programming, Interior point methods, Logarithmic penalty methods.*

1. Introduction

The domain of linear optimization (LO) is one of the most widely used techniques in economics and industry. Many efficient approaches have been proposed to correctly resolve (LO). Among the most effective

* *E-mail:* **mennichelinda03@gmail.com (Corresponding Author)**
† *E-mail:* **djbenterki@univ-setif.dz**
§ *E-mail:* **bmerikhi@univ-setif.dz**

are those of the simplex and of interior point methods [1, 10, 11, 14, 15]. A logarithmic penalty methods using majorant or minorant functions are a class of interior point methods, introduced initially by Jean Pierre Crouzeix and Bachir Merikhi [4]. Based on the work of [4], Linda Menniche and Djamel Benterki [12], L. Menniche et al. [13] proposed some new approximate functions that offers displacements steps with a simple technique. Similarly, Larbi Cherif and Bachir Merikhi [2] proposed a penalty method using the technique of majorant function for solving nonlinear programming problem. The objective of our work is to improve the logarithmic barrier algorithm by proposing a new approximate function in order to avoid the technique of line search of Wolfe and improve the algorithm proposed by L. Menniche et al. [13]. This approximate function makes it possible to get closer more to the logarithmic barrier function and to significantly reduce the number and the computation time of an iteration.

The paper is composed of five sections: section 2 is reserved for a synthesis of the linear programming problem and its perturbed problem with some notations needed later. In section 3, we briefly present the barrier interior point method and propose a new approximate function for calculating the displacement-step. Concerning section 4, we present numerical experiments on some examples of linear programming problems to show the efficiency of our new proposed approach. We also give comparative results with those obtained by the approximate function proposed by L. Menniche and al. [13] and those obtained by the line search methods. Finally, a conclusion closes section 5.

2. Formulation of the problem

Consider the canonical linear problem

$$(D): \min b^t y, \text{ Subject to}: A^t y \geq c, y \in \mathbb{R}^m,$$

with A a $\mathbb{R}^{m\times n}$-matrix of full rank, $c \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$.
and its dual problem

$$(P): \max c^t x, \text{ Subject to}: Ax = b, x \in \mathbb{R}^n_+.$$

Preliminaries

The scalar product of x, y in $\mathbb{R}^n$ is

$$< x, y > = x^t y = \sum_{i=1}^{n} x_i y_i .$$

The Euclidean norm of $w \in \mathbb{R}^n$ is $\| w \| = < w, w >^{\frac{1}{2}}$.

In all the following, we suppose that the set of primal-dual strict feasibility solutions is not empty (condition **C1**).

Let (D_r) be the solutions of (P) and (D) perturbed problem without constraints associated to (D). (D_r) is formulated as

$$(D_r) : \min g_r(y), y \in \mathbb{R}^m .$$

Where for a strictly positive real *r*, we have

$$g_r(y) = \begin{cases} b^t y + nr \ln r - r \sum_{i=1}^{n} < A^t y - c, e_i > \text{if } A^t y - c > 0, \\ +\infty \qquad \text{otherwise.} \end{cases}$$

With $(e_i, i = 1, ..., n)$ denotes the canonical base in $\mathbb{R}^n$.

3. Resolution of the perturbed problem (D_r)

Since g_r is strictly convex, then if the problem (D_r) admits an optimal solution, it is unique.

To prove that (D_r) admits a solution, it is sufficient to demonstrate that the following set of cone of recession amounts to zero:

$$R_0((g_r)_\infty) = \{d \in \mathbb{R}^n, (g_r)_\infty(d) \le 0\} = \{0\}$$

where

$$(g_r)_\infty(d) = \lim_{t \to \infty} \frac{g_r(y + td) - g_r(y)}{t} = b^t d.$$

Lemma 3.1: [12] *Suppose that the condition **C1** holds and* $A^t d > 0$, *if* $b^t d \le 0$ *then* $d = 0$.

3.1 *Study of the convergence of* (D_r)

Lemma 3.2: [12] *Let* $r > 0$ *and* y_r *be a solution of* (D_r). *then the feasible point* y *becomes optimal of* (D) *where:* $\lim_{r \to 0} y_r = y$.

3.2 *Computation of Newton's descent direction*

To resolve the problem (D_r), we apply an interior point method. The optimality conditions are satisfied because (D_r) is strictly convex. Consequently, (y_r) will be a solution of (D_r) when it verifies:

$$\nabla g_r(y_r) = 0.$$

Thus, Newton's iteration is defined as $y_{k+1} = y_k + d_k$,
with d_k is obtened by

$$[\nabla^2 g_r(y)]d_k = -\nabla g_r(y).$$

Nothing ensures that the new iterated remains strictly feasible. To remedy this problem, we add at each iteration t_k, which satisfies $A^t(y_k + t_k d_k) - c > 0$.

3.3 *Effective computation of* t_k

The calculation of t_k generally passes by line search. Unfortunately, these laters are enormously expensive and impossible in certain difficult problems among which those of semidefinite programming. To avoid this difficulty, Jean-Piere Crouzeix and Bachir Merikhi [4] and Linda Menniche and Djamel Benterki [12] have introduced another simple approach based on majorant function ψ_0 to approaches the following function

$$\psi(t) = \frac{1}{r}(g_r(y+td) - g_r(y)), \tag{3.1}$$

such that y and $y+td$ are strictly feasible

This function ψ_0 offers by a simple manner, a displacement step at each iteration. The efficiency of this displacement step can be measured by numerical tests established at the end of this work.

Before presenting our new minorant function, we give in the following lemma another simple form of the function $\psi(t)$.

Lemma 3.3: [12] *We set* $\bar{t} = \sup\{t,\ \textit{Subject to}\ 1 + tz_i > 0\}$ *and* $z_i = \frac{<e_i, A^t d>}{<e_i, A^t y - c>}$, $\forall i = \overline{1,n}$.

$\psi(t)$ *can be reformulated as*

$$\psi(t) = t(\sum_{i=1}^{n} z_i - \| z \|^2) - \sum_{i=1}^{n} \ln(1 + tz_i), t \in 0, \bar{t}], \tag{3.2}$$

such that

$$\psi(0) = 0, \psi'(0) = -\psi''(0) = \| z \|^2 . \tag{3.3}$$

3.3.1 Approximate functions of ψ

a) Approximate function ψ_0

We looked for an approximate functions ψ_0 of ψ on $[0,\bar{t}]$ such that ψ_0 verifies (3.3). In [12], the authors applied the Theorem 5 given in [4] and proposed the approximate function

$$\psi_0(t) = t\gamma_0 - (n-1)\ln(1+t\alpha_0) - \ln(1+t\beta_0), \tag{3.4}$$

with

$$\begin{cases} \gamma_0 = n\bar{z} - \| z \|^2, \\ \alpha_0 = \bar{z} + \frac{\sigma_z}{\sqrt{n-1}}, \\ \beta_0 = \bar{z} - \sigma_z\sqrt{n-1}. \end{cases}$$

and

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i,\ \sigma_x^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 = \frac{1}{n}\sum_{i=1}^{n} x_i^2 - \bar{x}^2.$$

When we set $x_i = 1 + tz_i$, then $\bar{x} = 1 + t\bar{z}$ and $\sigma_x = t\sigma_z$,

It is clear that ψ_0 is defined and convex on $[0,\bar{t}_0[$, with $\bar{t}_0 = \sup\{t, \text{Subject to } 1 + t\beta_0 > 0\}$ and satisfied (3.3).

b) New approximate function ψ_N of $\psi(t)$

Here, we propose a simpler function than ψ_0 involving only one logarithm given by

$$\psi_N(t) = t\frac{\|z\|^2}{\alpha_0} - p\ln(1 + t\frac{\|z\|^2}{\alpha_0}),\ \forall t \ge 0,\ 0 < p < 1. \tag{3.5}$$

In this lemma, we give the relation between ψ and his approximated ψ_0 and ψ_N.

Lemma 3.4: *Let* ψ_0 *and* ψ_N *as defined in* (3.4) *and* (3.5). *Then,* ψ_0 *and* ψ_N *are strictly convex on the interval* $[0,\bar{t}_0[$, *and satisfies*

$$\psi_N(t) \le \psi(t) \le \psi_0(t) \le +\infty,\ \forall t \in 0,\bar{t}_0[.$$

Proof: We have $\psi(t) = n\bar{z}t - t\| z \|^2 - \sum_{i=1}^{n}\ln(1+tz_i)$, we set

$$\phi(t) = \psi(t) - \psi_N(t),$$
$$= (\gamma_0 - \frac{\|z\|^2}{\alpha_0})t - \sum_{i=1}^{n} \ln(1+tz_i) + p\ln(1+\frac{t\|z\|^2}{\alpha_0}).$$

Then

$$\phi(0) = \phi'(0) = 0,$$

and we have $\forall t \geq 0$

$$\phi'(t) = \gamma_0 - \frac{\|z\|^2}{\alpha_0} - \sum_{i=1}^{n} \frac{z_i}{1+tz_i} + \frac{p\frac{\|z\|^2}{\alpha_0}}{1+t\frac{\|z\|^2}{\alpha_0}},$$

and

$$\phi''(t) = \sum_{i=1}^{n}\left[\left(\frac{z_i}{1+tz_i}\right)^2 - \frac{z_i^2}{\left(1+t\frac{\|z\|^2}{\alpha_0}\right)^2}\right] \geq 0.$$

Which gives

$$\psi_N(t) \leq \psi(t),\ \forall t \geq 0.$$

So, since $\psi(t) \leq \psi_0(t) \leq +\infty$ holds from Theorem 5 given in [4], then

$$\psi_N(t) \leq \psi(t) \leq \psi_0(t) \leq +\infty,\ \forall t \in 0, \overline{t}_0[.$$

Unique minimum of the functions ψ_N and ψ_0 are given respectively by

$$t_N = \frac{\alpha_0^2}{\|z\|^2},\ t_0 = b_0 - \sqrt{b_0^2 - c_0},$$

where $b_0 = \frac{1}{2}\left(\frac{n}{\gamma_0} - \frac{1}{\alpha_0} - \frac{1}{\beta_0}\right)$ and $c_0 = \frac{-\|z\|^2}{\alpha_0\beta_0\gamma_0}$.

3.4 *Description of the algorithm*

- **Begin**
- **Initialization:** Let $y_0 \in \mathbb{R}^m$, such that $A^t y_0 > c$, $d_0 \in \mathbb{R}^m$, ε a fixed precision and $k = 0$.
- **While** $|b^t d_k| > \varepsilon$
  - Resolve $[\nabla^2 g_r(y_k)]d_k = -\nabla g_r(y_k)$,

- Compute $\overline{z}$ and σ_z,
- Obtain t_k using approximate function ψ_0 or ψ_N or line search methods,
- Take $y_{k+1} = y_k + t_k d_k$
- Set $k = k+1$.

- **End while**
- **End.**

4. Numerical tests

This section is reserved for comparative tests of the obtained algorithm on different examples with fixed-dimension and variable-dimension of linear programming problems [8, 9] using MATLAB R2008b language.

We set:

(1) (K) iteration number,

(2) (CPU) necessary computation time,

(3) (Ls) classical Wolfe's line search.

The taken precision $\varepsilon := 10^{-4}$.

We summarize in the table below the obtained results for different examples with fixed-dimension.

	t_N		t_0		Ls	
	K	CPU	K	CPU	K	CPU
Example 1	16	0.0116	16	0.0115	19	0.0194
Example 2	17	0.0130	27	0.0405	19	0.0220
Example 3	21	0.0252	28	0.0276	24	0.0340
Example 4	21	0.0173	21	0.0152	20	0.0220

4.1 *Variable-dimension example*

$$(DL)\begin{cases} \min[2(y_1 + y_2 +, \cdots, + y_m)] \\ y_i - 1 \geq 0,\ \forall i = \overline{1, m},\ \text{and } n = 2m. \end{cases}$$

We summarize in the table below the obtained results for different dimensions (m,n).

	t_N		t_0		Ls	
(m,n)	K	CPU	K	CPU	K	CPU
$(100,200)$	19	1.2383	21	3.9275	21	3.8405
$(200,400)$	18	4.6955	19	11.7435	22	5.1735
$(300,600)$	18	16.1931	21	19.31	22	20.0657
$(400,800)$	18	62.744	21	65.2224	23	69.7205
$(500,1000)$	18	137.6894	22	171.69	24	182.500
$(600,1200)$	17	296.8285	22	315.95	24	337.4537
$(1000,2000)$	18	1038.2	22	1370.8	24	1477.1

Comments on the obtained results

In established numerical tests, approximate functions require an acceptable number of iterations to yield a solution of (D) in minimal time. Besides, our new approximate function ψ_N is the best, compared with the approximate function ψ_0 and line search Wolfe's method

5. Conclusion

This study shows that the approximate functions offered displacement steps that introduced the improvement of the penalty methods. In fact, the computation time, and the number of iterations, are considerably reduced compared to Wolfe's line search methods.

Acknowledgements

The authors sincerely appreciate the anonymous referees and thank them for their comments that helped to make the overall article better.

References

[1] A. Asadi, C. Roos, Infeasible Interior point methods for linear optimization based on large neighbourhood. *Journal of Optimization Theory and Applications* 170, 562-590 (2016).

[2] L. B. Cherif, B. Merikhi, A penalty method for nonlinear programming, *RAIRO-Operations Research*, 53, 29-38 (2019).

[3] J.P. Crouzeix, A. Seeger, New bounds for the extreme values of a finite sample of real numbers, *Journal of Mathematical Analysis and Applications*, 197, 411-426 (2008).

[4] J.P. Crouzeix, B. Merikhi, A logarithm barrier method for semidefinite programming, *RAIRO-Operations Research*, 42, 123-139 (2008).

[5] A.V. Fiacco, G.P. McCormick, Nonlinear programming: Sequential unconstrained minimization techniques, Wiley Reprinted as volume 4 of SIAM Classics in Applied Mathematics Series (1990).

[6] R.A.K. Frish, The logarithmic potential method of convex programming, Technical report, University Institute of Economics, Olso, Noway (1955).

[7] N. Karmarkar, A new polynomial-time algorithm in linear programming, Combinatorica 4, 373-395 (1984).

[8] A. Keraghel, Etude adaptative et comparative des principales variantes de l'algorithme de Karmarkar, (Thèse de Doctorat) Universit é Joseph Fourier, Grenoble, France, (1989).

[9] A. Keraghel, D. Benterki, Sur les performances de l'algorithme de Karmarkar pour la programmation linéaire, Revue Roumaine des sciences techniques mécaniques appliquées, 46, 87-96 (2001).

[10] M. Kojima, N. Megiddo, S. Mizuno, A primal-dual infeasible interior point method for linear programming, *Mathematical Programming* 61, 263-280 (1993).

[11] I.J. Lustig, Feasibility issues in a primal-dual interior point method for linear programming. *Mathematical Programming* 49, 145-162 (1990/1991).

[12] L. Menniche, D. Benterki, A logarithmic barrier approach for linear programming, *Journal of Computational and Applied Mathematics* 312, 267-275 (2017).

[13] L. Menniche, D. Benterki, I. Benchetta, An efficient logarithmic barrier method for linear programming, *Journal of Information and Optimization Sciences* 42(8), 1799-1813 (2021).

[14] C. Roos, A ful-Newton step o(n) infeasible interior point algorithm for linear optimization. *SIAM Journal on Optimization* 6, 1110-1136 (2006).

[15] A. Zaarat, S. Radjef, Adaptative method for linear programming problems with hybrid variables, *Journal of Information and Optimization Sciences* 42(3), 513-531 (2021).

Received July, 2022

Revised November, 2022