Afr. Mat. (2016) 27:13–21
DOI 10.1007/s13370-014-0315-2

Finding a strict feasible dual solution of a convex optimization problem

El Amir Djeffal · Lakhdar Djeffal · Farouk Benoumelaz

Received: 22 February 2014 / Accepted: 24 December 2014 / Published online: 8 January 2015

Abstract In this paper, we use an interior points method of projective type for calculate the initial dual solution of the convex nonlinear programming using the Ye Lustig variant applied to linear programming, as a result we propose a modification in this algorithm to reduce the number of iterations and the computing time of this algorithm. The numerical tests confirm that the modified algorithm is robust.

Keywords Quadratic programming · Convex nonlinear programming · Interior point methods

Mathematics Subject Classification 90C30 · 90C51

1 Introduction

The nonlinear programming (*NLP*) is a model which traduces many real applications. It can be found in control theory, combinatory optimization. In term of research, it is one of subject treated with fervour, in particular the problem of initialization in problem of optimization [1–8].

Choice of starting primal and dual point is an important practical issue with a significant effect on the robustness of the algorithm. A poor choice (x^0, y^0, s^0) satisfying only the minimal conditions $x^0 > 0$, $s^0 > 0$. In interior point methods, the successive iterates should be strictly feasible. In consequence, a major concern is to find an initial primal feasible solution x^0 [7,9]. The object of this paper is the dual problem (y^0, s^0).

E. A. Djeffal (✉) · L. Djeffal · F. Benoumelaz
Mathematics Department, University Hadj Lakhdar, Batna, Algeria
e-mail: elamir.djeffal@gmail.com

L. Djeffal
e-mail: lakdar_djeffal@yahoo.fr

F. Benoumelaz
e-mail: fbenoumelaz@yahoo.fr

A nonlinear program in its standard form is of type:

$$(CNLP)\begin{cases} \min\ f(x) \\ s.t \\ Ax = b,\ x \geq 0 \end{cases}$$

where $f(x)$ is a reel nonlinear function, $A \in \mathbb{R}^{m\times n}$ with $rg(A) = m < n,\ c,\ x\ \in \mathbb{R}^n$ and $y,\ b \in \mathbb{R}^m$, and its dual problem

$$(CNLD)\begin{cases} \max L(x, y, s) \\ s.t \\ A^t y + s = \nabla f(x),\ s \geq 0,\ y \in \Re^m. \end{cases}$$

where $L(x, y, s)$ is a lagrangian function, $s \in \mathbb{R}^n,\ y \in \mathbb{R}^m$ are the vectors.

Some notations are used throughout the paper and they are as follows. $\Re^n$, $\Re^n_+$ and $\Re^n_{++}$ denote the set of vectors with n components, the set of nonnegative vectors and the set of positive vectors, respectively. $\Re^{n\times n}$ denotes the set of $n \times n$ real matrices.$\|.\|_2$ denote the Euclidean norm, $e = (1, \ldots, 1)^t$ is the vector of ones in $\Re^n$.

2 Presentation of the problem

The strict dual feasibility problem associated to the problem $(CNLD)$ is to find a vector (y, s) such that:

$$\left[(y, s) \in \Re^{m+n} : A^t y + s = \nabla f(x),\ s > 0\right]\ (DF)$$

We poses $y = y^+ - y^-$, where $y^+ > 0,\ y^- > 0$.

The problem (DF) is equivalent to the following problem

$$\left[(y, s) \in \Re^{m+n} : A^t y^+ - A^t y^- + s = \nabla f(x),\ y^+ > 0, y^- > 0,\ s > 0\right]\ (DF)$$

One way to solve a strictly dual feasible problem consists in introducing an additional variable ξ as follows:

$$(P_\xi)\begin{cases} \min \xi \\ s.t \\ Dd + \xi(\nabla f(x) - Da) = \nabla f(x) \\ d > 0,\ \xi > 0 \end{cases}$$

where D is the $(n) \times (n+2m)$ matrix defined by $D = (A^t,\ -A^t,\ I)$, $a \in \Re^{n+2m}_{++}$ is arbitrary and $d = (y^+,\ y^-,\ s) \in \Re^{n+2m}_{++}$

The problem (P_ξ) can be written as a following linear program:

$$(LP_\xi)\begin{cases} \min c''tl \\ s.t \\ D'l = \nabla f(x) \\ l > 0 \end{cases}$$

where $c'' \in \Re^{n+2m+1}$ is the vector defined by:

$$c''[i] = \begin{cases} 1,\ \text{if } i = n + 2m + 1 \\ 0,\ \text{otherwise} \end{cases}$$

D' is the $(n) \times (n+2m+1)$ matrix defined by $D' = (D,\ \nabla f(x) - Da)$ and $l \in \Re^{n+2m+1}$ is vector such that $l = (d,\ \xi)$.

For solving the problem (LP_ξ), we use the Ye Lustig variant of projective interior point method [5].

Lemma 1 [5] *x^* is a solution of problem (PF) if and only if $(d^*,\ \xi)$ is an optimal solution of problem $(LP_\xi)$with $d \in \Re^{n+2m}_{++}$ and ξ sufficiently small.*

To compute the optimal solution of problem (LP_ξ) , we use only the second phase of the projective interior point method. The corresponding algorithm is:

3 Algorithm for solving (LP_ξ)

Description of the algorithm

(a) Initialization: $\varepsilon > 0$ is fixed, $d^0 = e,\ \xi_0 = 1$ and $k = 0$

If $\|Dd - \nabla f(x)\|_2 < \varepsilon$, Stop: d^k is an ε-approximate solution of (DF).

If not go to **(b)**.

(b) If $\xi_k < \varepsilon$, Stop: d^k is an ε -approximate solution of (DF).

If not go to (**c**).

(**c**) **Step** k

Determinate:

$$G_k = diag(l^k), D = [D,\ \nabla f(x) - Da]\,, r = \frac{1}{\sqrt{(n+2m+1)(n+2m+2)}},$$
$$C_k = \left[D' G_k,\ \ -\nabla f(x)\right]$$
$$P_k = \left[I - C_k^t (C_k C_k^t)^{-1} C_k\right]\left[G_k c'' t,\ \ -c'' t l^k\right]^t$$

Compute:

$$z^{k+1} = \frac{e_{n+2m+2}}{n+2m+2} - \alpha_k r \frac{P_k}{\|P_k\|_2}$$
$$l^{k+1} = \frac{1}{z^{k+1}[n+2m+2]} G_k z^{k+1}\,[n+2m+1]$$

Take:

$\xi_k = l^{k+1}\,[n+2m]$, and go to **(b)**

4 Modified algorithm

At iteration k, the original algorithm gives a strictly feasible solution (d^k, ξ^k). To find the next iteration (d^{k+1}, ξ^{k+1}), we search a vector w^k such that the vector $d^k + w^k$ is a solution of problem (FD), i.e

$$D(d^k + w^k) = \nabla f(x) \tag{1}$$

and

$$d^k + w^k > 0 \tag{2}$$

(1) is equivalent to $Dd^k = \nabla f(x) - \xi_k q$, where

$$Dw^k = \xi_k q \tag{3}$$

(3) is equivalent to the following convex quadratic optimization problem

$$(COP)\quad \begin{cases} \min \left\| w^k \right\|^2 \\ s.t \\ Dw^k = \xi_k q \end{cases}$$

By optimality condition, we obtain the follwing system

$$\begin{cases} Dw^k = \xi_k q & (a) \\ w^k + D^t\theta = 0 & (b) \\ Dw^k + DD^t\theta = 0 & (c) \end{cases}$$

we obtain
$\theta = -(DD^t)^{-1}\xi_k q$
deduce
$w^k = -D^t\theta = D^t(DD^t)^{-1}\xi_k q.$
The calculation of w^k requires the calculation of the inverse of the matrix (DD^t), which is undesirable in practice, to correct, we proceed as follows, let

$$u^k = (DD^t)^{-1}\xi_k q$$

equivalent to the follwing linear symetric definite positive system

$$DD^t u^k = \xi_k q$$

Proposition 1 *For all* (d^k, ξ_k) *is a strict feasible solution of* (P_ξ), *if* $\max \left| v_i^k \right| < 1$ *then*:

$$d^k + w^k > 0 \text{ and } D(d^k + w^k) = \nabla f(x)$$

where

$$v^k = -diag\left(\frac{1}{d^k}\right) D^t u^k \text{ and } w^k \text{ is a solution of } (COP)$$

Proof **1.** $D(d^k + w^k) = \nabla f(x)$
let, w^k is a solution of (COP), we have
$w^k = D^t u^k$ and $u^k = \xi_k u^0$

$$\begin{aligned} D(d^k + w^k) &= D(d^k + D^t u^k) \\ &= Dd^k + DD^t u^k \\ &= \nabla f(x) - \xi_k\big(\nabla f(x) - Da\big) + \xi_k DD^k u^0 \\ &= \nabla f(x) - \xi_k\big(\nabla f(x) - Da\big) + \xi_k\xi_0\big(\nabla f(x) - Da\big) \\ &= \nabla f(x). \end{aligned}$$

2. $d^k + w^k > 0$
we have

$$D(d^k + w^k) = \nabla f(x)$$

so, $D(diag(d^k)e_{n+2m} + w^k) = \nabla f(x)$
then,

$$Ddiag(d^k)\Big(e_{n+2m} + diag\big[(d^k)\big]^{-1} w^k\Big) = \nabla f(x)$$

as $d^k > 0$ then,

$$e_{n+2m} + diag\left[(d^k)\right]^{-1} w^k > 0$$

for $d^k + w^k > 0$,
let:

$$v^k = -diag\left(\frac{1}{d^k}\right) w^k = -diag\left(\frac{1}{d^k}\right) D^t u^k$$

we have, $e_{n+2m} + diag\left[(d^k)\right]^{-1} w^k > 0$ then
$e_{n+2m} - v^k > 0$, giving:
$\max \left|v_i^k\right| < 1$
therefore, we have: $d^k + w^k > 0$. □

4.1 The modified algorithm

Description of the algorithm

(a′) Initialization: $\varepsilon > 0$ is fixed, $d^0 = e,\ \xi_0 = 1$ and $k = 0$
If $\left\| Dd^0 - \nabla f(x)\right\|_2 < \varepsilon$, Stop: d^k is an ε-approximate solution of (DF).
else
calculated u^0 solution of the linear system

$$DD^t u^0 = \xi_0(\nabla f(x) - Da).$$

(b′)
If $\xi_k < \varepsilon$
Stop: d^k is an ε-approximate solution of (DF).
else
Compute

$$u^k = \xi_k u^0$$
$$v^k = -diag\left(\frac{1}{d^k}\right) D^t u^k$$

If $\max \left|v_i^k\right| < 1$, stop: $d^k + D^t u^k$ is an ε-approximate solution of (DF).
else
go to **(c′)**
(c′) is identical to **(c)** of the algorithm original.
(d′) take $k = k + 1$ and go to **(b′)**.

5 Numerical tests

The algorithm has been tested on some benchmark problems issued from the paper [7,9].The implementation is manipulated in DEV C++, We have taken $\varepsilon = 10^{-8}$.

Example 1 [9]

$$\begin{cases} \min\ \frac{1}{2} x^t Qx + c^t x \\ s.t \\ Ax = b,\ x \geq 0 \end{cases}$$

Table 1 Numerical results

Algorithm	Number of iteration	Time (s)
Ye-Lustig variant algorithm	11	0.03
Modified algorithm	6	0.01

Table 2 Numerical results

Algorithm	Number of iteration	Time (s)
Ye-Lustig variant algorithm	9	0.02
Modified algorithm	1	0.01

where

$$A = \begin{pmatrix} 1 & 1.2 & 1 & 1.8 & 0 \\ 3 & -1 & 1.5 & -2 & 1 \\ -1 & 2 & -3 & 4 & 2 \end{pmatrix}, \quad c = \begin{pmatrix} 1 & -1.5 & 2 & 1.5 & 3 \end{pmatrix}^t, \quad b = \begin{pmatrix} 9.31 \\ 5.45 \\ 6.60 \end{pmatrix}$$

$$Q = \begin{pmatrix} 20 & 1.2 & 0.5 & 0.5 & -1 \\ 1.2 & 32 & 1 & 1 & 1 \\ 0.5 & 1 & 14 & 1 & 1 \\ 0.5 & 1 & 1 & 15 & 1 \\ -1 & 1 & 1 & 1 & 16 \end{pmatrix}$$

In this example, we find the dual initial point (s^0, y^0)
$s^0 = (29.824284 \;\; 10.732446 \;\; 0.544773 \;\; 1.013929 \;\; 0.488309)^t$
$y^0 = (20.822601 \;\; 5.717161 \;\; 3.343051)$
The experiment results see Table 1.

Example 2 [7]

$$\begin{cases} \min \; x_1^2 + 0.5x_2^2 + x_3^2 + 0.5x_4^2 - x_1x_3 + x_3x_4 - x_1 - 3x_2 + x_3 - x_4 \\ s.t \\ x_1 + 2x_2 + x_3 + x_4 + x_5 = 5 \\ 3x_1 + x_2 + 2x_3 - x_4 + x_6 = 4 \\ -x_1 - x_4 + x_7 = -1.5 \\ -x_1 + x_8 = 0 \\ -x_2 + x_9 = 0 \\ -x_3 + x_{10} = 0 \\ x_4 + x_{11} = 0 \end{cases}$$

In this example, we find the dual initial point (s^0, y^0)

$$s^0 = \begin{pmatrix} 1.364487 \;\; 0.047156 \;\; 0.332105 \;\; 0.068909 \;\; 0.888686 \;\; 0.021166 \\ 0.209226 \;\; 0.250756 \;\; 0.044618 \;\; 0.069522 \;\; 0.201388 \end{pmatrix}^t$$

$$y^0 = \begin{pmatrix} -0.888686 \;\; -0.021166 \;\; -0.209226 \;\; -0.250756 \;\; -0.044618 \\ -0.069522 \;\; -0.201388 \end{pmatrix}$$

The experiment results see Table 2.

Example 3 [9]

$$\begin{cases} \min \ \frac{1}{2}x^t Qx + c^t x \\ s.t \\ Ax = b, \ x \geq 0 \end{cases}$$

where

$$A = \begin{pmatrix} 1 & -1 & 1.9 & 1.25 & 1.2 & 0.4 & -0.7 & 1.06 & 1.5 & 1.05 \\ 1.3 & 1.2 & 0.15 & 2.15 & 1.25 & 1.5 & 0.4 & 1.52 & 1.3 & 1 \\ 1.5 & -1.1 & 3.5 & 1.25 & 1.8 & 2 & 1.95 & 1.2 & 1 & -1 \end{pmatrix},$$

$$b = \begin{pmatrix} 11.651 \\ 16.672 \\ 21.295 \end{pmatrix}, \quad c = \begin{pmatrix} -0.5 & -1 & 0 & 0 & -0.5 & 0 & 0 & -1 & -0.5 & -1 \end{pmatrix}^t$$

$$Q = \begin{pmatrix} 30 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 21 & 0 & 1 & -1 & 1 & 0 & 1 & 0.5 & 1 \\ 1 & 0 & 15 & -0.5 & -2 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & -0.5 & 30 & 3 & -1 & 1 & -1 & 0.5 & 1 \\ 1 & -1 & -2 & 3 & 27 & 1 & 0.5 & 1 & 1 & 1 \\ 1 & 1 & 1 & -1 & 1 & 16 & -0.5 & 0.5 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0.5 & -0.5 & 8 & 1 & 1 & 1 \\ 1 & 1 & 1 & -1 & 1 & 0.5 & 1 & 24 & 1 & 1 \\ 1 & 0.5 & 1 & 0.5 & 1 & 0 & 1 & 1 & 39 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 11 \end{pmatrix}$$

In this example, we find the dual initial point (s^0, y^0)

$$s^0 = \begin{pmatrix} 20.514528 & 7.457094 & 1.792312 & 10.121809 & 14.555866 \\ 0.601053 & 3.827566 & 2.716075 & 18.527227 & 0.800151 \end{pmatrix}^t$$

$$y^0 = (3.9721091 \ \ 8.084740 \ \ 4.420401)$$

The experiment results see Table 3.

Example 4 (Erikson's problem 1980)
We consider the following convex problem

$$\begin{cases} \min f(x) = \sum_{i=1}^{n} x_i \ln \frac{x_i}{a_i} \\ s.t \\ x_i + x_{i+m} = b_i, \ i = 1, \ldots, m, \quad n = 2m \\ x \geq 0 \end{cases}$$

where $a_i \in \Re_{++}$ and $b_i \in \Re$ are fixed. We have tested this example for different value of n, a_i and b_i.

Table 3 Numerical results

Algorithm	Number of iteration	Time (s)
Ye-Lustig variant algorithm	10	0.02
Modified algorithm	6	0.01

Algorithm	Number of iteration	Time (s)
$a_i = 1$, $b_i = 6$		
$n = 10$		
Ye-Lustig variant algorithm	32	0.03
Modified algorithm	23	0.02
$n = 100$		
Ye-Lustig variant algorithm	29	0.02
Modified algorithm	21	0.02
$n = 300$		
Ye-Lustig variant algorithm	27	0.02
Modified algorithm	20	0.01
$n = 500$		
Ye-Lustig variant algorithm	26	0.01
Modified algorithm	17	0.01
$a_i = 2$, $b_i = 4$		
$n = 10$		
Ye-Lustig variant algorithm	28	0.03
Modified algorithm	17	0.02
$n = 100$		
Ye-Lustig variant algorithm	25	0.03
Modified algorithm	15	0.01
$n = 300$		
Ye-Lustig variant algorithm	23	0.02
Modified algorithm	13	0.01
$n = 500$		
Ye-Lustig variant algorithm	21	0.02
Modified algorithm	11	0.01
$a_i = 10$, $b_i = 1$		
$n = 10$		
Ye-Lustig variant algorithm	32	0.04
Modified algorithm	25	0.02
$n = 100$		
Ye-Lustig variant algorithm	31	0.03
Modified algorithm	23	0.01
$n = 300$		
Ye-Lustig variant algorithm	31	0.03
Modified algorithm	23	0.01
$n = 500$		
Ye-Lustig variant algorithm	28	0.02
Modified algorithm	16	0.01

6 Conclusion

Our modification, we have improved the numerical behavior of the algorithm, reducing the number of iterations corresponding to the search of an initial strictly feasible dual solution of the problem (*CNLP*). It is also possible to applied this idea to the search of an optimal solution of the (*CNLP*) problem.

Acknowledgments The authors thank the referees for their careful reading and their precious comments. Their help is much appreciated.

References

1. Benterki, D.: Résolution des problèmes de programmation semi-définie par la méthode de réduction du potentiel. Université Ferhat Abbas, Sétif, Thèse de Doctorat (2004)
2. Djeffal, E.A., Djeffal, L.: Feasible short-step interior point algorithm for linear complementarity problem based on kernel function. Adv. Model. Optim. **15**(2), 157–172 (2013)
3. Djeffal, E.A.: Etude numérique d'une méthode de trajectoire centrale pour la programmation convexe sous contraintes lin éaires. Université Hadj Lakhdar, Batna, Thèse de magister (2009)
4. Benterki, D., Merikhi, B.: A modified algorithm for the strict feasibility. RAIRO Oper. Res. **35**, 395–399 (2001)
5. Lustig, I.J.: Feasibility issues in a primal-dual interior point method for linear programming. Math. Program. **49**, 145–162 (1991)
6. Benterki, D., Crouzeix, J.P., Merikhi, B.: A numerical implementation of an interior point method for semidefinite programming. Pesqui. Operacional **23**(1), 49–59 (2003)
7. Fan, X., Bo, Y.: A polynomial path following algorithm for convex programming. Appl. Math. Comput. **196**, 866–878 (2008)
8. Benterki, D., Keraghel, A.: Finding a strict feasible solution of a linear semidefinite program. Appl. Math. Comput. **217**, 6437–6440 (2011)
9. Roumili, H.: Méthodes de points intérieurs non r éalisables en optimisation. Théorie, algorithmes et applications, Th èse de Doctorat, Université Ferhat Abbas, Sétif (2007)