REVUE D'ANALYSE NUMÉRIQUE ET DE THÉORIE DE L'APPROXIMATION
Rev. Anal. Numér. Théor. Approx., vol. 43 (2014) no. 2, pp. 159–167
ictp.acad.ro/jnaat

IMPROVING COMPLEXITY OF KARMARKAR'S APPROACH FOR LINEAR PROGRAMMING

DJAMEL BENTERKI[*] and MOUSAAB BOUAFIA[†]

Abstract. In this paper, we are interested in the performance of Karmarkar's projective algorithm for linear programming. Based on the work of Schrijver, we offer a new displacement step better than Schrijver's one which led to a moderate improvement in the behavior of the algorithm shift. We show later that the algorithm converges after $\frac{n}{1-\log(2)+(\frac{nr^2}{10})}\log\left(\frac{c^te_n}{\varepsilon}\right)$ iterations.

This purpose is confirmed by numerical experiments showing the efficiency of the obtained algorithm, which are presented in the last of paper.

MSC 2000. 90C05, 65K05.

Keywords. Linear programming, interior point method, potential function.

1. INTRODUCTION

Karmarkar's algorithm for solving linear optimization problems [1] is the first truly efficient algorithm that solves these problems in polynomial time. The ellipsoid method also works in polynomial time, but it is ineffective in practice. Using a potential function shows that the Karmarkar's algorithm converges after a number $O(nq + n\log n)$ iterations for a displacement step $\alpha_K = \frac{1}{4}$. Since then, many researchers are interested in this algorithm in order to more improve its numerical behavior. The two covered basic elements covered are the direction of displacement that dominates the cost computation at each iteration and the displacement step which plays an important role in the speed of convergence. Indeed, by changing the analytical form of the potential function, Padberg [2] was able to reduce the number of iterations to $\mathcal{O}(nq)$ for a displacement step $\alpha_P = \frac{1}{2}$. For its part, Schrijver [7], keeping the same Karmarkar's potential function has been shown that the algorithm converges after $\frac{n}{1-\log(2)}\log\left(\frac{c^te_n}{\varepsilon}\right)$ iterations for a displacement step $\alpha_S = \frac{1}{1+nr}$.

[*] Laboratoire de Mathématiques Fondamentales et Numériques LMFN, Département de Mathématiques, Faculté des Sciences, Université Sétif-1, Algérie, e-mail: dj_benterki@yahoo.fr.

[†] LabCAV, Laboratoire de Contrôle Avancé, Département de Mathématiques, Faculté des Mathématiques et de l'Informatique et des Sciences de la Matière, Université de Guelma, Algérie, e-mail: mousaab84@yahoo.fr.

Our work falls within this framework, where drawing works of Karmarkar [1] and Shrijver [7], we propose a new displacement step to improve the numerical behavior of the algorithm of Karmarkar.

The paper is organized as follows. In Section 2, we present the linear program treated by Karmarkar. Section 3 is the main objective of our work. At first, we propose a new displacement step and then we show that it is better than that of Schrijver. On the other hand, we give an improved polynomial convergence result compared to the Schrijver's one. In Section 4, we present some computational results. Finally, a conclusion is provided in Section 5.

2. LINEAR PROGRAMMING PROBLEM TREATED BY KARMARKAR

In his article [1], Karmarkar treats the linear programming problem in the following reduced form:

$$(PK)\begin{cases} \min c^t x = z^* = 0 \\ Ax = 0 \\ x \in S_n = \left\{x \in \mathbb{R}^n_+ \ , \ e^t_n x = 1\right\}, \end{cases}$$

where $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$ is a matrix of full rank $(\operatorname{rank}(A) = m < n\ )$ and $e_n = (1, 1, ..., 1)^t \in \mathbb{R}^n$.

The vector $x^0 = \frac{e_n}{n} \in S_n$, is the center of the simplex S_n.

2.1. Karmarkar's Transformation on S_n.

The Karmarkar's transformation is defined at each iteration by:

$$\begin{aligned} T_k : S_n &\longrightarrow S_n \\ x \mapsto y = \quad & T_k(x) = \tfrac{D_k^{-1}x}{e_n^t D_k^{-1} x} \end{aligned}$$

such that $D_k = \operatorname{diag}\left(x^k\right)$ is the $n \times n$ diagonal matrix with the components of x^k as the diagonal entries.

Similarly, the transformation T_k is invertible and we have:

$$x = T_k^{-1}(y) = \frac{D_k y}{e_n^t D_k y}.$$

The transformed problem of (PK) by the transformation T_k is the linear programming problem:

$$(PKT)\begin{cases} \min (D_k c)^t y \\ A D_k y = 0 \\ y \in S_n = \left\{y \in \mathbb{R}^n_+ : e^t_n y = 1\right\}, \end{cases}$$

which can be written as:

$$(PKT)\begin{cases} \min\left(D_k c\right)^t y \\ \begin{bmatrix} AD_k \\ e_n^t \end{bmatrix} y = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ y \geq 0. \end{cases}$$

The transformed problem (PKT) is placed in the reduced form of Karmarkar, and the transformation T_k can bring the iterated point x^k in the center of the simplex S_n, i.e., $T_k\left(x^k\right) = \frac{e_n}{n}$.

Before applying the optimality conditions, Karmarkar [1] relaxes the problem (PKT) by replacing the condition $y \geq 0$ by the sphere $s\left(\frac{e_n}{n}, \alpha r\right)$, (it is easily shown in [2] that if $y \in s\left(\frac{e_n}{n}, \alpha r\right)$ then $y \geq 0$) with $r = \frac{1}{\sqrt{n(n-1)}}$ and $0 < \alpha < 1$.

The problem (PKT) becomes:

$$(PKT)_r\begin{cases} \min\left(D_k c\right)^t y \\ \begin{bmatrix} AD_k \\ e_n^t \end{bmatrix} y = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \left\|y - \frac{e_n}{n}\right\|^2 \leq (\alpha r\ )^2. \end{cases}$$

Using the optimality conditions, the optimal solution of the problem $(PKT)_r$ is given by:

$$y = \frac{e_n}{n} - \alpha r d_k,$$

where $d_k = \frac{P_k}{\|P_k\|}$, such that P_k is the projection of the vector $D_k c$ on the kernel of the constraint matrix AD_k.

Returning by the inverse transform T_k^{-1}, a feasible solution x^{k+1} is obtained of the original problem (PK) such that $x^{k+1} = T_k^{-1}\left(y^k\right) = \frac{D_k y^k}{e_n^t D_k y^k}$.

Remark 1. Initially, Karmarkar has taken the displacement step α between 0 and 1. This ensures that all iterates remain inside the simplex and show the polynomial convergence of the algorithm. In numerical terms, it appears that more the displacement step α is large more the algorithm converges faster. For this reason, several researchers suggest to use variable displacement step through the line search method. Unfortunately, the procedure is expensive and there are no convergence results. Our aim is to improve the theoretical results given by Karmarkar, by using short step that depend on the size of the considered problem. This idea is inspired by the work of R. Naseri & A. Valinejad [5] and Schrijver [7]. For the moment, we give the polynomial convergence theorem of Karmarkar's algorithm. □

2.2. Convergence of Karmarkar's algorithm. To study the polynomial convergence of the algorithm, Karmarkar used (in case $z^* = 0$) the following potential function:

$$f(x) = n \log c^t x - \sum_{i=1}^{n} \log x_i$$

defined on the set

$$D_x = \left\{ x \in \mathbb{R}^n : x > 0, Ax = 0, e_n^t x = 1 \right\}.$$

THEOREM 2 (Karmarkar's convergence theorem). [1]
If $0 < \alpha_K \leq \frac{1}{4}$, *then starting from* $x^0 = \frac{e_n}{n}$, *after* $\mathcal{O}(nq+\ n \log n)$ *iterations, the algorithm finds a feasible point* x^* *such that:*
(1) $c^t x^* = 0$
or
(2) $\frac{c^t x^*}{c^t x^0} \leq 2^{-q}$, *where q is a fixed precision.*

REMARK 3. In his work, Padberg [2] was able to improve the convergence of the classical algorithm of Karmarkar by replacing Karmarkar's potential function as follows:

$$h(x) = \frac{c^t x}{\left(\prod\limits_{i=1}^{n} x_i \right)^{\frac{1}{n}}}$$

defined on the set

$$D_x = \left\{ x \in \mathbb{R}^n : x > 0, Ax = 0, e_n^t x = 1 \right\}$$

and showed that for a displacement step $\alpha_P = \frac{1}{2}$, the algorithm converges after $\mathcal{O}\left(nq\right)$ iterations. In the same direction regardless of Padberg, Schrijver [7] has shown that for a displacement step $\alpha_S = \frac{1}{1+nr}$, the algorithm converges after $\frac{n}{1-\log 2} \log \frac{c^t e_n}{\varepsilon}$ iterations. □

In the next section, we propose a new value of displacement step α_B better than Schrijver's one, whose aim is to improve further the speed of convergence of Karmarkar's algorithm.

3. IMPROVING CONVERGENCE OF KARMARKAR'S ALGORITHM

In this section, we propose a new displacement step α_B better than the one which was introduced by Schrijver, in order to reduce the number of iterations and accelerate the convergence of Karmarkar's algorithm and then, we give the corresponding results of the algorithm's complexity. Before that, we present some lemmas below, in order to help the reader to understand the results that we have established.

LEMMA 4. [6] *Let* ψ_n *be a function defined by:*

$$\psi_n :]-1, +\infty[^n \ \to \mathbb{R}_+$$

$$x \quad \mapsto \quad \psi_n(x) = e_n^t x - \sum_{i=1}^{n} \log(1 + x_i)$$

then for $\|x\| < 1$ *we have:*

$$\psi_1(-\|x\|) \geq \psi_n(-x).$$

LEMMA 5. [6] *If* $x \in D_x$ *then:*

$$c^t x \leq \exp\left(\frac{f(x)}{n}\right)$$

where f *is the Karmarkar's potential function.*

More Generally, Keraghel [2] showed the following Lemma:

LEMMA 6. [2] *Let* x^k *be the* $k - th$ *iterate of Karmarkar's algorithm, then:*

$$\frac{c^t x^k}{c^t x^0} \leq \left(\exp\left[f(x^k) - f(x^0)\right]\right)^{\frac{1}{n}},$$

where

$$x^0 = \frac{e_n}{n}.$$

LEMMA 7. [6] *At each iteration* k, *the potential function decreases from the following quantity:*

$$f(x^{k+1}) - f(x^k) = -\Delta,$$

such that:

$$x^{k+1} = T_k^{-1}(y^k) \qquad and \qquad \Delta = n \log \frac{c^t D_k e_n}{c^t D_k y^k} + \sum_{i=1}^{n} \log y_i^k$$

LEMMA 8. [6] *If* $0 < \alpha < \frac{1}{nr}$ *then:*

$$\Delta \geq \alpha n^2 r^2 + n\psi_1\left(-\alpha \frac{r}{R}\right) - \psi_1(-\alpha nr),$$

where $R = \sqrt{\frac{n-1}{n}}$

Based on the previous lemmas, we present the following results.

LEMMA 9. *For* $\alpha_B = \frac{5+nr^2}{5+(5+nr^2)nr}$ *and* $\alpha_S = \frac{1}{1+nr}$, *we have:*

a) $\alpha_B > \alpha_S$,
b) $\psi_1\left(-\alpha_B nr^2\right) > \frac{1}{5}\alpha_B n^2 r^4$,
c) $\Delta \geq \psi_1\left(\frac{1}{5}nr\left(5 + nr^2\right)\right)$.

Proof. **a)** Since $5 + nr^2 > 5$, then

$$\left(5 + nr^2\right)(1 + nr) > 5 + \left(5 + nr^2\right)nr \Leftrightarrow \frac{\left(5+nr^2\right)}{5+(5+nr^2)nr} > \frac{1}{1+nr}$$

which gives:

$$\alpha_B > \alpha_S.$$

b) By definition, we have:

$$\psi_1\left(-\alpha_B nr^2\right) \quad = -\alpha_B nr^2 - \log\left(1 - \alpha_B nr^2\right),$$

Since $\log(1-x) < -x - \frac{x^2}{2}$ for $0 < x < 1$ then:

$$\begin{aligned}\psi_1\left(-\alpha_B nr^2\right) &> -\alpha_B nr^2 + \alpha_B nr^2 + \tfrac{\alpha_B^2 n^2 r^4}{2}\\ &> \tfrac{\alpha_B n^2 r^4}{2}\alpha_S > \tfrac{1}{5}\alpha_B n^2 r^4\end{aligned}$$

because:

$$\alpha_S = \tfrac{1}{1+nr} \geq \tfrac{1}{1+\sqrt{2}} > \tfrac{2}{5}$$

so:

$$\psi_1\left(-\alpha_B nr^2\right) > \tfrac{1}{5}\alpha_B n^2 r^4.$$

c) By Lemma 8 and since $\frac{1}{R} = nr$ we have:

$$\Delta \geq \alpha n^2 r^2 + n\psi_1\left(-\alpha nr^2\right) - \psi_1\left(-\alpha nr\right), \quad \text{with } 0 < \alpha < \tfrac{1}{nr},$$

for $\alpha = \alpha_B = \frac{5+nr^2}{5+(5+nr^2)nr}$ and by **b)** we get:

$$\begin{aligned}\Delta &> \alpha_B n^2 r^2 + n\left(\tfrac{1}{5}\alpha_B n^2 r^4\right) - \psi_1\left(-\alpha_B nr\right)\\ &> \alpha_B n^2 r^2 + \tfrac{1}{5}nr^2\left(\alpha_B n^2 r^2\right) + \alpha_B nr \ + \log\left(1-\alpha_B nr\right)\\ &> \left(n^2r^2 + \tfrac{1}{5}nr^2\left(n^2r^2\right) + nr\right)\alpha_B + \log\left(1-\alpha_B nr\right)\\ &> \tfrac{1}{5}\left(5n^2r^2 + nr^2\left(n^2r^2\right) + 5nr\right)\tfrac{5+nr^2}{5+(5+nr^2)nr} + \log\left(1 - \tfrac{5+nr^2}{5+(5+nr^2)nr}nr\right)\\ &> \tfrac{1}{5}nr\left(5nr + nr^2\left(nr\right) + 5\right)\tfrac{5+nr^2}{5+(5+nr^2)nr} + \log\left(1 - \tfrac{5+nr^2}{5+(5+nr^2)nr}nr\right)\\ &> \tfrac{1}{5}nr\left(\left(5+nr^2\right)\left(nr\right) + 5\right)\tfrac{5+nr^2}{5+(5+nr^2)nr} + \log\left(\tfrac{5}{5+(5+nr^2)nr}\right)\\ &> \tfrac{1}{5}nr\left(5+nr^2\right) + \log\left(\tfrac{5}{5+(5+nr^2)nr}\right)\\ &> \tfrac{1}{5}nr\left(5+nr^2\right) + \log\left(\tfrac{1}{1+(5+nr^2)\frac{nr}{5}}\right)\\ &> \tfrac{1}{5}nr\left(5+nr^2\right) - \log(1+\tfrac{1}{5}nr\left(5+nr^2\right)) = \psi_1\left(\tfrac{1}{5}nr\left(5+nr^2\right)\right)\end{aligned}$$

Hence:

$$\Delta > \psi_1\left(\tfrac{1}{5}nr\left(5+nr^2\right)\right).$$

□

REMARK 10. It is easily shown that ψ_1 is increasing on $[0,+\infty[$ and in particular:

$$\psi_1\left(\tfrac{1}{5}nr\left(5+nr^2\right)\right) \geq \psi_1\left(nr\right) \geq \psi_1\left(1\right) = 1 - \log\left(2\right).$$ □

3.1. Improved displacement step.

Recall that for $\alpha_S = \frac{1}{1+nr}$, Schrijver [7] showed that the algorithm converges after $\frac{n}{1-\log(2)}\log\left(\frac{c^t e_n}{\varepsilon}\right)$ iterations.

In this section, based on the previous Lemmas, we show in the following theorem that the new displacement step α_B reduces the number of iterations required for the convergence of Karmarkar's algorithm.

Theorem 11. *If*

$$\alpha_B = \tfrac{5+nr^2}{5+(5+nr^2)nr},$$

then Karmarkar's algorithm converges after

$$\tfrac{n}{1-\log(2)+(\frac{nr^2}{10})}\log\left(\tfrac{c^t e_n}{\varepsilon}\right)$$

iterations.

Proof. By Lemma 9 **c)** we have**:**

$$\Delta > \psi_1\left(\tfrac{1}{5}nr\left(5+nr^2\right)\right) = \psi_1\left(nr + \tfrac{nr}{5}nr^2\right) > \psi_1\left(1+\tfrac{1}{5}nr^2\right),$$

such as $nr > 1$ and ψ_1 is increasing, then

$$\Delta > \psi_1\left(1+\tfrac{1}{5}nr^2\right), \tag{3.1}$$

and since $\log(1+x) \leq x$, we have:

$$\psi_1\left(1+\tfrac{1}{5}nr^2\right) - \psi_1(1) > \tfrac{nr^2}{10}.$$

So from (3.1), we get:

$$\Delta > \psi_1(1) + \tfrac{nr^2}{10} \tag{3.2}$$

Also from Lemma 7, we have:

$$f(x^k) - f(x^{k-1}) = -\Delta, \quad \forall k \in \mathbb{N}^*,$$

using (3.2) we get:

$$f(x^i) - f(x^{i-1}) < -\left(\psi_1(1) + \tfrac{nr^2}{10}\right),$$

accordingly:

$$\sum_{i=1}^{k}(f(x^i) - f(x^{i-1})) < -\sum_{i=1}^{k}\left(\psi_1(1) + \tfrac{nr^2}{10}\right),$$

which gives:

$$f(x^k) - f(x^0) < -k\left(\psi_1(1) + \tfrac{nr^2}{10}\right), \quad \text{where } x^0 = \tfrac{e_n}{n},$$

then:

$$f(x^k) < -k\left(\psi_1(1) + \tfrac{nr^2}{10}\right) + f(x^0),$$

such that:

$$f(x^0) = f(\tfrac{e_n}{n}) = n\log c^t\tfrac{e_n}{n} \; - \sum_{i=1}^{n}\log\tfrac{1}{n} = n\log c^t e_n,$$

so:

$$\tfrac{f(x^k)}{n} < \tfrac{-k\left(\psi_1(1)+\frac{nr^2}{10}\right)+n\log c^t e_n}{n}, \tag{3.3}$$

by Lemma 5 and (3.3), we have:

$$c^t x^k < \exp\left(\tfrac{-k\left(\psi_1(1)+\frac{nr^2}{10}\right)+n\log c^t e_n}{n}\right).$$

Recall that the algorithm converges when $c^t x^k$ is less than ε ($\varepsilon > 0$ small enough). So, it returns to search k which satisfies:

$$\exp\Big(\frac{-k\big(\psi_1(1) + \frac{nr^2}{10}\big) + n\log c^t e_n}{n}\Big) < \varepsilon,$$

which can be done:

$$k > \tfrac{n}{1-\log 2+\frac{nr^2}{10}} \log \tfrac{c^t e_n}{\varepsilon}. \tag{3.4}$$

Finally $c^t x^k \le \varepsilon$ when k satisfies the inequality (3.4), hence the result. □

Remark 12. The previous Theorem 11 gives a displacement step α_B greater than the displacement step α_S of Schrijver which improve the practical behavior of the algorithm and the theoretical performance results, in the sense that the number of iterations given in this theorem is less than the number of iterations given by Schrijver. □

4. NUMERICAL EXPERIMENTS

In this part, we present a comparative tests on different numerical examples of linear programming problem taken from the literature [3]. We tested the Ye-Lustig's approach of Karmarkar's algorithm (case z^* is unknown) using differently two alternatives, once the displacement step α_S of (Schrijver) and again the displacement step α_B of (Bouafia & Benterki). The precision ε is taken between 10^{-4} and 10^{-6}.

In each case, k denotes the number of iterations required for the optimality.

Example	(m, n)	Number of iterations α_S Schrijver	Number of iterations α_B Bouafia & Benterki
1	$(3,4)$	$k = 18$	$k = 17$
2	$(3,5)$	$k = 26$	$k = 25$
3	$(4,7)$	$k = 36$	$k = 35$
4	$(5,9)$	$k = 47$	$k = 46$
5	$(5,11)$	$k = 46$	$k = 46$
6	$(6,12)$	$k = 48$	$k = 47$
7	$(16,26)$	$k = 71$	$k = 70$

Table 1. Comparative table

5. CONCLUSION

This study shows that the new displacement step α_B has introduced an improvement in the polynomial convergence results, and a small reduction in the number of iterations.

Note that when the dimension of the problem becomes important, the displacement step α_B and α_S converge to $\frac{1}{2}$. It will probably be interesting to focus much more on the analytical form of the potential function while preserving all of its good properties, in order to obtain a best complexity results.

Acknowledgements. The authors are grateful to the referee whose detailed comments greatly improve the presentation of the paper.

REFERENCES

[1] N. Karmarkar, *A new polynomial-time algorithm for linear programming*, Combinatorica, **4** (1984), pp. 373–395.

[2] A. Keraghel, *Etude adaptative et comparative des principales variantes dans l'algorithme de Karmarkar*, Dissertation thesis, Université Joseph Fourier, Grenoble, France, 1989.

[3] A. Keraghel and D. Benterki, *Sur les performances de l'algorithme de Karmarkar pour la programmation linéaire*, Revue Roumaine des Sciences Techniques-Mécanique Appliquées, **46** (2001) No. 1, pp. 87–96.

[4] I.J. Lustig, *A practical approach to Karmarkar's algorithm*, Technical report sol 85-5 Systems optimization laboratory; Dept. of operations research. Stanford. Univ.; Stanford California 94305, 1985.

[5] R. Naseri and A. Valinejad, *An extended variant of Karmarkar's interior point algorithm*, Applied Mathematics and Computation **184** (2007), pp. 737–742.

[6] C. Roos, T. Terlaki and J. Vial, *Optimization Theory and Algorithm for Linear Programming Optimization*, Princeton University, 2001.

[7] A. Schrijver, *Theory of linear and integer programming*, John Wiley & Sons, New York, 1986.

Received by the editors: November 2, 2014.

Published online: January 23, 2015.