

## Research Article

# Novel Interior Point Algorithms for Solving Nonlinear Convex Optimization Problems

Sakineh Tahmasebzadeh,<sup>1</sup> Hamidreza Navidi,<sup>1</sup> and Alaeddin Malek<sup>2</sup>

<sup>1</sup>Department of Applied Mathematics, Shahed University, P.O. Box 3319118651, Tehran, Iran

<sup>2</sup>Department of Mathematics, Tarbiat Modares University, P.O. Box 14115-175, Tehran, Iran

Correspondence should be addressed to Hamidreza Navidi; navidi@shahed.ac.ir

Received 13 April 2015; Revised 26 August 2015; Accepted 30 August 2015

Academic Editor: Ching-Jong Liao

Copyright © 2015 Sakineh Tahmasebzadeh et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes three numerical algorithms based on Karmarkar's interior point technique for solving nonlinear convex programming problems subject to linear constraints. The first algorithm uses the Karmarkar idea and linearization of the objective function. The second and third algorithms are modification of the first algorithm using the Schrijver and Malek-Naseri approaches, respectively. These three novel schemes are tested against the algorithm of Kebiche-Keraghel-Yassine (KKY). It is shown that these three novel algorithms are more efficient and converge to the correct optimal solution, while the KKY algorithm fails in some cases. Numerical results are given to illustrate the performance of the proposed algorithms.

## 1. Introduction

The simplex method has never had any serious competition until 1984 when Karmarkar proposed a new-polynomial-time algorithm in order to solve linear programming problems, especially for solving large scale problems [1, 2]. Polynomial complexity of Karmarkar's algorithm has an advantage in comparison with exponential complexity of the simplex algorithm [3–5]. The improvement of the Karmarkar's algorithm by Schrijver [6, 7] resulted in less number of iterations compared to Karmarkar's method. In 2004, Malek and Naseri [8] proposed a modified technique based on the interior point algorithm to solve linear programming problems more efficiently.

After the appearance of Karmarkar's algorithm for solving linear programming problems, the researchers developed the algorithm to solve the convex quadratic programming problem (e.g., see [9]). Considering the success of the interior methods for solving linear programming problems [10, 11], the researchers used the linearization methods to solve convex nonlinear programming problems. In 2007, Kebiche et al. [12] introduced a projective interior point method to solve more general problems with nonlinear convex objective function. The objective of this paper is to propose an optimal

step length in each iteration in combination with Karmarkar's algorithm in order to decrease the nonlinear objective function as fast as possible.

In Section 2, extension of Karmarkar's algorithm for nonlinear programming problem is considered. In Section 3, by considering the technique associated with KKY's algorithm, a modified algorithm is presented. In Section 4, the convergence of modified algorithm in Section 3 is proved. Then this algorithm is combined with Schrijver's and Malek-Naseri's algorithms successfully. In Section 5, numerical results are illustrated for KKY algorithm and the three suggested modified algorithms are compared to each other.

## 2. Extension of Karmarkar's Algorithm for Nonlinear Problems

Consider the following nonlinear optimization problem:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & Ax = b, \\ & x \geq 0, \end{aligned} \tag{1}$$

where  $f : R^n \rightarrow R$  is a nonlinear, convex, and differentiable function.  $A$  is  $m \times n$  matrix of rank  $m$  and  $b$  is  $m \times 1$ . The starting point  $x^0 > 0$  is chosen to be feasible.  $I$  is the identity matrix with order  $(n+1)$  and  $e_{n+1}^T$  is a row vector of  $n+1$  ones.

Assume  $y = (y_1, y_2, \dots, y_{n+1})^T = (y[n], y_{n+1})^T$ , the simplex  $S = \{y \in R^{n+1} \mid e_{n+1}^T y = 1, y \geq 0\}$ , and the projective transformation of Karmarkar  $\mathbb{T}_k$  defined by  $\mathbb{T}_k : R^n \rightarrow S$  such that  $\mathbb{T}_k(x) = y$ . Consider  $g(y) = y_{n+1}[f(\mathbb{T}_{k+1}^{-1}(y)) - f(\mathbb{T}_k^{-1}(y))]$ , where  $g : R^{n+1} \rightarrow R$  is a nonlinear, convex, and differentiable function and the optimal value of  $g$  is zero. Using the linearization process to the function  $g$  in neighborhood of the ball of center  $y^0$  as  $g(y) = g(y^0) + \nabla g(y^0)(y - y^0)$  for  $y \in \{y \in R^{n+1} \mid \|y - y^0\| \leq \beta\}$  along with Karmarkar projective transformation [12], we conclude that Problem (1) is equivalent to

$$\begin{aligned} \min \quad & \nabla g(y^0)^T y \\ \text{s.t.} \quad & A_k y = 0, \\ & e_{n+1}^T y = 1, \\ & \|y - y^0\| \leq \beta. \end{aligned} \quad (2)$$

As a result, the optimal solution of the preceding problem lies along the negative projection of the gradient  $C_p$  and is given as  $y^{k+1} = y^0 - \beta(C_p/\|C_p\|)$ , where  $y^0 = (1/(n+1), \dots, 1/(n+1))^T$  is the center of the simplex  $S$ .  $C_p$  is the projected gradient which can be shown to be  $C_p = [I - P^T(P P^T)^{-1}P]\nabla g(y^0)$ , where  $P = \begin{pmatrix} A_k \\ e_{n+1}^T \end{pmatrix}$ ,  $A_k = [AD_k, -b]$ ,  $D_k = \text{diag}\{x^k\}$ , and  $k$  is the number of iterations. The selection of  $\beta$  as a step length is crucial to enhance the efficiency of the algorithm.

The function  $g$  on  $Y = \{y \in R^{n+1} \mid A_k y = 0, y \in S\}$  is convex, since the function  $f$  on  $X = \{x \in R^n \mid Ax = b, x \geq 0\}$  is convex. The optimal solution of Problem (2) is given by  $y^{k+1} = y^0 - \beta(C_p/\|C_p\|)$  [12].

Consider the algorithm of Kebiche-Keraghel-Yassine (KKY) for solving Problem (2).

*KKY Algorithm.* Let  $\epsilon > 0$  be a given tolerance and let  $x^0 > 0$ .

*Step 1.* Compute  $y^0 = (1/(n+1), \dots, 1/(n+1))^T$  and  $\beta = (n+1)/3(n+2)$ . Put  $k = 0$ .

*Step 2.* Build  $D_k = \text{diag}\{x^k\}$ ,  $A_k = [AD_k, -b]$  and  $P = \begin{pmatrix} A_k \\ e_{n+1}^T \end{pmatrix}$ .

Compute  $C_p = [I - P^T(P P^T)^{-1}P]\nabla g(y^0)$ ,  $y^{k+1} = y^0 - \beta(C_p/\|C_p\|)$ , and  $x^{k+1} = D_k y^{k+1}[n]/y_{n+1}^k$ .

*Step 3.* While  $f(x^{k+1}) - f(x^k) > \epsilon$ , let  $k = k + 1$ , and go to Step 2.

However, as we can see in the following two examples, the KKY algorithm does not work for every nonlinear problem.

*Example 1.* Consider the quadratic convex problem:

$$\begin{aligned} \min \quad & f(x) = x_1^2 + x_2^2 - 2x_1 - 4x_2 \\ \text{s.t.} \quad & x_1 + 4x_2 + x_3 = 5, \\ & 2x_1 + 3x_2 + x_4 = 6, \\ & x_1, x_2, x_3, x_4 \geq 0. \end{aligned} \quad (3)$$

For tolerance  $\epsilon = 10^{-8}$ , after  $k = 161$  iterations for KKY algorithm we have

$$\begin{aligned} x^{161} &= [0.00000000, 2.00005934, -3.00008255, \\ &-0.00000000]^T \text{ and } f(x^{161}) = -3.9999999964. \text{ Since} \\ &\text{the condition } f(x^{162}) - f(x^{161}) = 4.47 \times 10^{-9} < 10^{-8} \\ &\text{holds, the algorithm stopped, whereas this solution is} \\ &\text{not feasible.} \end{aligned}$$

*Example 2.* Consider the nonlinear convex problem:

$$\begin{aligned} \min \quad & f(x) \\ &= \frac{3}{2}(x_1^2 + x_2^2) + 2(x_3^2 + x_4^2) - \ln(x_1 x_4) \\ &\quad - 3x_1 x_2 + 4x_3 x_4 - 2x_1 - 3x_4 \\ \text{s.t.} \quad & x_1 + \frac{1}{4}x_4 - 8x_5 - x_6 + 9x_7 = 0, \\ & x_2 + \frac{1}{2}x_4 - 12x_5 - \frac{1}{2}x_6 + 3x_7 = 0, \\ & x_3 + x_6 = 1, \\ & x_1, x_2, \dots, x_7 \geq 0. \end{aligned} \quad (4)$$

For  $\epsilon = 10^{-8}$ ,  $\beta = 8/27 \in [0.27, 0.36]$ , and  $k = 1$  iteration using KKY algorithm the solution  $x^1 = [-5.2199, -1.8871, 0.4463, 0.3791, -0.0087, 0.5536, 0.6232]^T$  is calculated. This solution is not feasible.

Here, two difficulties arise. First,  $x^k$  must stay feasible in each iteration. Second, in the moderate time the required tolerance must be satisfied. To overcome these difficulties we modify the KKY algorithm in the next section.

### 3. Modified Algorithm

*3.1. Modifications.* In the KKY [12] algorithm  $\beta \in [0.27, 0.36]$ . As it is observed in Examples 1 and 2, for these values of  $\beta$ , the KKY algorithm may produce infeasible solution. In the standard Karmarkar [1, 3, 4] instead of  $\beta$ ,  $\alpha r = ((n+1)/3(n+2))(1/\sqrt{(n+1)(n+2)})$  is used. Hence the solution in each iteration remains feasible. In this case, the optimal solution for Problem (2) is given by  $y^{k+1} = y^0 - \beta(C_p/\|C_p\|)$ , where  $\beta = \alpha r$ . Applying KKY algorithm with  $\beta = \alpha r$  to the problem in Example 2, it gives the feasible solution  $x^{80} = [0.7517262, 0.0000174, 0.0000056, 1.0209196, 0.0008742, 0.9999943, 0.0000035]^T$ ,  $f(x^{80}) = -1.3692774913$ , and  $f(x^{81}) = -1.3692882373$ . The algorithm stops after  $k = 80$

iterations since  $f(x^{81}) - f(x^{80}) = -1.0746 \times 10^{-5}$ , while the suitable accuracy  $\epsilon = 10^{-8}$  is not reached. In each iteration, linear search method is used to find  $0 < \lambda \leq 1$  such that  $f(x^{k+1}) \leq f(x^k)$ , where the suitable tolerance satisfies. Thus one may write Problem (2) in the following form:

$$\begin{aligned} \min \quad & \nabla g(y^0)^T y \\ \text{s.t.} \quad & A_k y = 0, \\ & e_{n+1}^T y = 1, \\ & \|y - y^0\| \leq \lambda \alpha r. \end{aligned} \quad (5)$$

**Lemma 3.** *The optimal solution for Problem (5) is given by  $y^{k+1} = y^0 - \lambda \alpha r(C_P / \|C_P\|)$ .*

*Proof.* We put  $x = y - y^0$ , and then we have  $Px = \begin{pmatrix} A_k \\ e_{n+1}^T \end{pmatrix} (y - y^0) = 0$  and Problem (5) is equivalent to

$$\begin{aligned} \min \quad & \nabla g(y^0)^T x \\ \text{s.t.} \quad & Px = 0, \\ & \|x\|^2 \leq \lambda^2 \alpha^2 r^2. \end{aligned} \quad (6)$$

$x^*$  is a solution of Problem (6) if and only if there exist  $\omega \in R^{m+1}$  and  $\mu \geq 0$  such that

$$\nabla g(y^0) + P^T \omega + \mu x^* = 0. \quad (7)$$

Multiplying both sides of (7) by  $P$  and since  $Px^* = 0$ , we get  $P\nabla g(y^0) + PP^T \omega = 0$ . Then  $\omega = -(PP^T)^{-1}(P\nabla g(y^0))$ ; by substituting  $\omega$  in (7) we find

$$x^* = -\frac{1}{\mu} [I - P^T (PP^T)^{-1} P] \nabla g(y^0). \quad (8)$$

By assuming  $C_P = [I - P^T (PP^T)^{-1} P] \nabla g(y^0)$ , we have

$$\begin{aligned} \|x^*\| &= \frac{1}{\mu} \|C_P\| = \lambda \alpha r \\ \implies \frac{1}{\mu} &= \frac{\lambda \alpha r}{\|C_P\|}, \\ \implies x^* &= -\lambda \alpha r \frac{C_P}{\|C_P\|}. \end{aligned} \quad (9)$$

And we have  $y^{k+1} = y^* = y^0 + x^* = y^0 - \lambda \alpha r(C_P / \|C_P\|)$ .  $\square$

Note that here we have proposed the algorithm similar to KKY, where  $\beta = \lambda \alpha r$ . This modified algorithm has the advantage that can find the feasible approximate solution in the suitable tolerance.

**3.2. Modified KKY Algorithm (MKKY).** Let  $\epsilon > 0$  be a given tolerance and  $x^0$  is a strictly feasible point.

*Step 1.* Compute  $y^0 = (1/(n+1), \dots, 1/(n+1))^T$ ,  $r = 1/\sqrt{(n+1)(n+2)}$ , and  $\alpha = (n+1)/3(n+2)$ . Put  $\lambda = 1$  and  $k = 0$ .

*Step 2.* Build  $D_k = \text{diag}\{x^k\}$ ,  $A_k = [AD_k, -b]$ , and  $P = \begin{pmatrix} A_k \\ e_{n+1}^T \end{pmatrix}$ . Compute  $C_P = [I - P^T (PP^T)^{-1} P] \nabla g(y^0)$ ,  $y^{k+1} = y^0 - \lambda \alpha r(C_P / \|C_P\|)$ , and  $x^{k+1} = D_k y^{k+1} [n] / y_{n+1}^k$ .

*Step 3.* While  $f(x^{k+1}) - f(x^k) > \epsilon$ , put  $k = k + 1$ .

Build  $D_k = \text{diag}\{x^k\}$ ,  $A_k = [AD_k, -b]$ , and  $P = \begin{pmatrix} A_k \\ e_{n+1}^T \end{pmatrix}$ . Compute  $C_P = [I - P^T (PP^T)^{-1} P] \nabla g(y^0)$  and  $y^{k+1} = y^0 - \lambda \alpha r(C_P / \|C_P\|)$ . Then  $x^{k+1} = T_k^{-1}(y^{k+1}) = D_k y^{k+1} [n] / y_{n+1}^k$ . Let  $k = k + 1$ , and go to Step 3.

*Step 4.* While  $f(x^{k+1}) > f(x^k)$ , compute  $\lambda = (1/2)\lambda$ ,  $y^{k+1} = y^0 - \lambda \alpha r(C_P / \|C_P\|)$ , and  $x^{k+1} = D_k y^{k+1} [n] / y_{n+1}^k$ .

*Step 5.* Let  $k = k + 1$ , and go to Step 3.

#### 4. Convergence for Modified Algorithm (MKKY)

In order to establish the convergence of the modified algorithm, we introduce a potential function associated with Problem (1) defined by

$$\phi(x) = (n+1) \ln(f(x) - z^*) - \sum_{i=1}^n \ln(x_i), \quad (10)$$

where  $z^*$  is the optimal value of the objective function.

**Lemma 4.** *If  $y^{k+1}$  is the optimal solution of Problem (5), then  $g(y^{k+1}) < g(y^0)$ .*

*Proof.* Since  $g(y^{k+1}) = g(y^0) + \nabla g(y^0)^T (y^{k+1} - y^0)$  and  $y^{k+1} = y^0 - \lambda \alpha r(C_P / \|C_P\|)$ , then

$$\begin{aligned} g(y^{k+1}) - g(y^0) &= \nabla g(y^0)^T \left( -\lambda \alpha r \frac{C_P}{\|C_P\|} \right) \\ &= -\frac{\lambda \alpha r}{\|C_P\|} (\nabla g(y^0)^T C_P) \\ &= -\frac{\lambda \alpha r}{\|C_P\|} \|C_P\|^2 = -\lambda \alpha r \|C_P\| < 0. \end{aligned} \quad (11)$$

Thus  $g(y^{k+1}) < g(y^0)$  and a reduction is obtained in each iteration.  $\square$

**Lemma 5.** *If  $y^{k+1}$  is the optimal solution of Problem (5), then*

$$g(y^{k+1}) \leq \left(1 - \frac{\lambda \alpha}{n+1}\right) g(y^0). \quad (12)$$

*Proof.* Let  $x^*$  be the optimal solution of Problem (1); we can write  $x^* = T_k^{-1}(y^*)$ .  $B(y^0, \lambda \alpha r)$  is a ball of center  $y^0$  with radius  $\lambda \alpha r$ . There are two cases:

(i) If  $y^* \in Y \cap B(y^0, \lambda\alpha r)$ , then

$$0 \leq g(y^{k+1}) \leq g(y^*) = 0 < \left(1 - \frac{\lambda\alpha}{n+1}\right) g(y^0), \quad (13)$$

and Lemma 5 holds.

(ii) If  $y^* \notin Y \cap B(y^0, \lambda\alpha r)$ , since  $Y$  is convex, intersection point of the boundary of  $Y \cap B(y^0, \lambda\alpha r)$  and the line segment between  $y^0$  and  $y^*$  should be feasible for Problem (5). Let  $\hat{y}^{k+1}$  be the intersection point; then  $\hat{y}^{k+1}$  satisfies  $\hat{y}^{k+1} = \theta y^* + (1-\theta)y^0$  for  $\theta \in (0, 1)$ , and  $\|\hat{y}^{k+1} - y^0\| = \lambda\alpha r$ . Thus,

$$\begin{aligned} \|\hat{y}^{k+1} - y^0\| &= \|\theta(y^* - y^0)\| = \lambda\alpha r \implies \\ \|(y^* - y^0)\| &= \frac{\lambda\alpha r}{\theta}. \end{aligned} \quad (14)$$

Hence,

$$\|y^* - y^0\|^2 = \|y^*\|^2 + \|y^0\|^2 - 2y^{0T}y^*, \quad (15)$$

and we have

$$\begin{aligned} \|y^*\| &= \sum_{i=1}^{n+1} (y_i^*)^2 \leq \left(\sum_{i=1}^{n+1} y_i^*\right)^2 = (e_{n+1}^T y^*)^2 = 1, \\ \|y^0\|^2 &= \left\|\frac{1}{n+1} e_{n+1}\right\|^2 = \frac{1}{n+1}, \\ y^{0T} y^* &= \frac{1}{n+1} e_{n+1}^T y^* = \frac{1}{n+1}. \end{aligned} \quad (16)$$

Thus,

$$\|y^* - y^0\|^2 \leq 1 + \frac{1}{n+1} - \frac{2}{n+1} = \frac{n}{n+1} \leq \frac{n+1}{n+2}. \quad (17)$$

From (14) we also have  $\|y^* - y^0\|^2 = (\lambda\alpha r/\theta)^2$ ; therefore

$$\begin{aligned} \left(\frac{\lambda\alpha r}{\theta}\right)^2 &\leq \frac{n+1}{n+2} \text{ or} \\ \frac{\lambda\alpha r}{\theta} &\leq \frac{\sqrt{n+1}}{\sqrt{n+2}}. \end{aligned} \quad (18)$$

Substituting  $r = 1/\sqrt{(n+1)(n+2)}$  in the above inequality, we have  $\theta \geq \lambda\alpha/(n+1)$ . Since  $g$  is convex and  $g(y^*) = 0$ , then

$$\begin{aligned} g(\hat{y}^{k+1}) &\leq \theta g(y^*) + (1-\theta)g(y^0) \\ &\leq \left(1 - \frac{\lambda\alpha}{n+1}\right) g(y^0). \end{aligned} \quad (19)$$

Furthermore,  $g(y^{k+1}) \leq g(y^*)$ ; then  $g(y^{k+1}) \leq (1 - \lambda\alpha/(n+1))g(y^0)$ .  $\square$

**Theorem 6.** In every iteration of the algorithm MKKY, potential function is reduced by a constant value  $\delta$  such that  $\phi(x^{k+1}) \leq \phi(x^k) - \delta$ .

*Proof.* Consider

$$\begin{aligned} \phi(x^{k+1}) - \phi(x^k) &= (n+1) \ln \left( \frac{f(x^{k+1}) - z^*}{f(x^k) - z^*} \right) \\ &\quad - \sum_{i=1}^n \ln \left( \frac{x_i^{k+1}}{x_i^k} \right) \\ &= (n+1) \ln \left( \frac{g(y^{k+1})}{g(y^0)} \right) \\ &\quad - \sum_{i=1}^{n+1} \ln(y_i^{k+1}) \\ &\leq (n+1) \ln \left( 1 - \frac{\lambda\alpha}{n+1} \right) \\ &\quad - \sum_{i=1}^{n+1} \ln(y_i^{k+1}) \\ &\leq -\lambda\alpha + \frac{\lambda^2\alpha^2}{2(1-\lambda\alpha)^2}. \end{aligned} \quad (20)$$

We used the result demonstrated by Karmarkar [1]:

$$-\sum_{i=1}^{n+1} \ln(y_i^{k+1}) \leq \frac{\lambda^2\alpha^2}{2(1-\lambda\alpha)^2}. \quad (21)$$

Then  $\phi(x^{k+1}) \leq \phi(x^k) - \delta$ , where  $\delta = -\lambda\alpha + \lambda^2\alpha^2/2(1-\lambda\alpha)^2$ .

If  $\alpha = (n+1)/3(n+2)$ , then  $\phi(x^{k+1}) \leq \phi(x^k) - 0.2$ .

Therefore  $\phi(x^{k+1}) \leq \phi(x^0) - 0.2(k+1)$ .  $\square$

**Theorem 7.** If  $x^0$  is the feasible solution of Problem (1) and  $x^*$  is the optimal solution with optimal value  $z^*$ , then one has following assumptions:

- (1)  $x^0 \geq 2^{-L} e_n$ .
- (2)  $x^* \leq 2^L e_n$ ; for any feasible solution  $x$  one has  $-2^{3L} \leq z^* \leq f(x) \leq 2^{3L}$ .

The algorithm MKKY finds an optimal solution after  $O(nL)$  iteration.  $L$  is the number of bytes.

*Proof.* Consider

$$\frac{f(x^k) - z^*}{f(x^0) - z^*} = \gamma(x^k) \exp \left( \frac{\phi(x^k) - \phi(x^0)}{n+1} \right), \quad (22)$$

where

$$\gamma(x^k) = \exp \left( \frac{\sum_{i=1}^n \ln(x_i^k) - \sum_{i=1}^n \ln(x_i^0)}{n+1} \right). \quad (23)$$

By assumptions (1) and (2) we have  $\gamma(x^k) \leq 2^{2L}$  in the feasible region; then

$$\begin{aligned} f(x^k) - z^* &\leq 2^{2L} (f(x^0) - z^*) \exp \left( \frac{\phi(x^k) - \phi(x^0)}{n+1} \right). \end{aligned} \quad (24)$$

TABLE 1: Computed solutions using four different algorithms for Examples 8, 9, 10, and 11.

Examples	$x$	KKY	MKKY	Sch-MKKY	MN-MKKY
Example 8	$x_1$	1.1176285677	0.7993104755	0.7992885091	0.7984677572
	$x_2$	0.8925922545	1.2006283419	1.2006478942	1.2013756245
	$x_3$	-0.0102208222	0.0000611826	0.0000635969	0.0001566183
	$x_4$	1.3324440588	0.3980537916	0.3979927208	0.395765081
Example 9	$x_1$	-5.2199797446	0.7514629509	0.7514297188	0.7514274997
	$x_2$	-1.8871925195	0.0000032699	0.0000065967	0.0000068198
	$x_3$	0.4463304312	0.0000010511	0.0000021234	0.0000022020
	$x_4$	0.3791651531	1.0175937272	1.0172339995	1.0172101531
	$x_5$	-0.0087312913	0.0007335551	0.0007190583	0.0007180983
	$x_6$	0.5536695688	0.9999989489	0.9999978766	0.9999977980
	$x_7$	0.6232230772	0.0000006675	0.0000013472	0.0000013940
Example 10	$x_1$	0.0000000025	0.0000000156	0.0000000055	0.0000000080
	$x_2$	1.9999999829	1.9999999858	1.9999999749	1.9999999586
	$x_3$	0.0000000025	0.0000000156	0.0000000055	0.0000000080
Example 11	$x_1$	0.3124560511	0.7142857707	0.7142857350	0.7142857453
	$x_2$	0.3438507936	0.1428572168	0.1428571684	0.1428571600
	$x_3$	0.0001576431	0.0000002101	0.0000000702	0.0000000498
	$x_4$	-1.4064826279	0.0000001087	0.0000000363	0.0000000258

TABLE 2: Comparison between four different algorithms for Examples 8, 9, 10, and 11.

Examples	Algorithm	$k$	$\ x^*\ $	$f(x^*)$	Error	CPU time (sec)
Example 8	KKY (infeasible)	—	—	—	—	—
	MKKY	549	1.4962795449	-7.1998265727	$1.7344 \times 10^{-4}$	182.8281
	Sch-MKKY	515	1.4962672544	-7.1998196616	$1.8034 \times 10^{-4}$	153.4063
	MN-MKKY	187	1.4958093895	-7.1995511207	$4.4888 \times 10^{-4}$	83.5313
Example 9	KKY (infeasible)	—	—	—	—	—
	MKKY	181	1.6125110840	-1.3693639744	$1.6216 \times 10^{-5}$	213.1250
	Sch-MKKY	87	1.6122679375	-1.3693475555	$3.2635 \times 10^{-5}$	122.4229
	MN-MKKY	67	1.6122518087	-1.3693463904	$3.3800 \times 10^{-5}$	79.1875
Example 10	KKY	17	1.9999999829	-3.9999999317	$6.8305 \times 10^{-8}$	11.4844
	MKKY	80	1.9999999858	-3.9999999433	$5.6661 \times 10^{-8}$	46.5469
	Sch-MKKY	37	1.9999999749	-3.9999998994	$1.0058 \times 10^{-7}$	22.5625
	MN-MKKY	15	1.9999999586	-3.9999998343	$1.6570 \times 10^{-7}$	10.7031
Example 11	KKY (infeasible)	—	—	—	—	—
	MKKY	61	0.7284314289	-0.2039626101	$3.2040 \times 10^{-8}$	35.7188
	Sch-MKKY	30	0.7284313844	-0.2039626256	$8.9867 \times 10^{-9}$	18.4531
	MN-MKKY	12	0.7284313929	-0.2039626212	$1.3372 \times 10^{-8}$	9.0781

According to Theorem 6, after  $k$  iterations, we have  $\phi(x^k) - \phi(x^0) \leq -k\delta$ . Thus,

$$\begin{aligned}
 f(x^k) - z^* &\leq 2^{2l} (f(x^0) - z^*) \exp\left(-\frac{k\delta}{n+1}\right) \\
 &\leq 2^{5L} \exp\left(-\frac{k\delta}{n+1}\right) \leq 2^{-4L}.
 \end{aligned}
 \tag{25}$$

Therefore,  $f(x^k) - z^* \leq 2^{-4L}$  for  $k \geq 45L(n+1)$ .  $\square$

In the next section MKKY algorithm is combined with the algorithm of Schrijver [6, 7] and Malek-Naseri's algorithm [8] to propose novel hybrid algorithms called Sch-MKKY

(Schrijver-Modified Kebiche-Keraghel-Yassine) and MN-MKKY (Malek-Naseri-Modified Kebiche-Keraghel-Yassine). These algorithms are different from MKKY in the use of optimal length in each iteration.

**4.1. Hybrid Algorithms.** Let us assume that  $r$  and  $y^0$  in the modified algorithm are expressed as follows:  $r = \sqrt{(n+2)/(n+1)}$  and  $y^0 = (1, 1, \dots, 1)^T$ . In the Sch-MKKY and MN-MKKY algorithms, choose  $\alpha = 1/(1+r)$  and  $\alpha = 1 - 1/(n+2)^4(1 + \sqrt{n(n+1)})$ , respectively. It is easy to prove that the theorems in Section 4 satisfy Sch-MKKY and MN-MKKY algorithms. Thus with the recent step length the convergence is guaranteed.

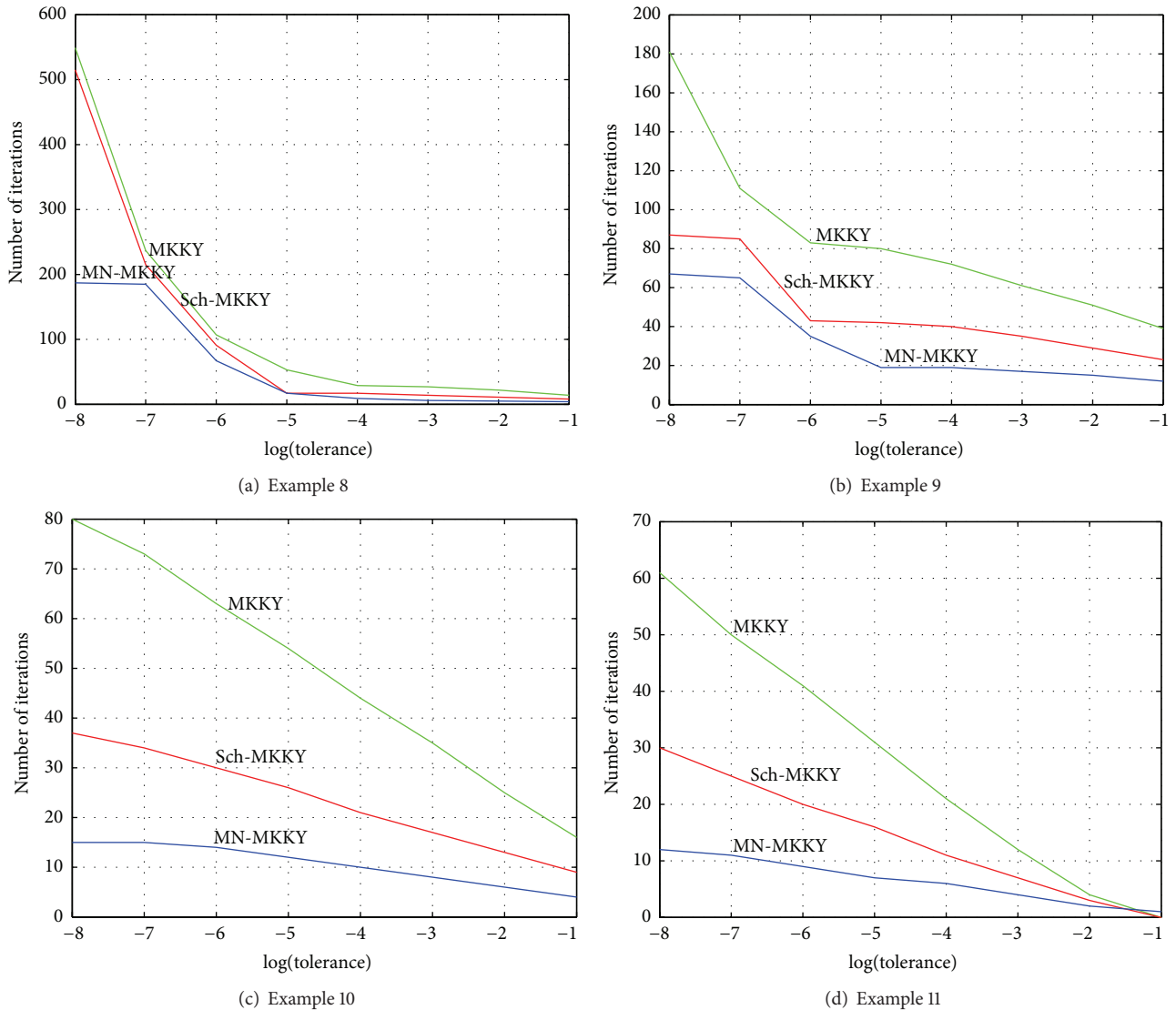


FIGURE 1: The tolerances against number of iterations are compared for MKKY, Sch-MKKY, and MN-MKKY algorithms for Examples 8, 9, 10, and 11.

### 5. Numerical Results

It is observed that the approximate solution from KKY algorithm is not feasible in some instances (see Examples 8, 9, and 11 in Table 1). All programs have been written with MATLAB 7.0.4 for  $\epsilon = 10^{-8}$  as tolerance. The computed solution for KKY, Sch-MKKY, and MN-MKKY is given in Table 1.

*Example 8.* Consider the quadratic convex problem:

$$\begin{aligned}
 \min \quad & f(x) = -2x_1 - 6x_2 + x_1^2 - 2x_1x_2 + 2x_2^2 \\
 \text{s.t.} \quad & x_1 + x_2 + x_3 = 2, \\
 & -x_1 + 2x_2 + x_4 = 2, \\
 & x_1, \dots, x_4 \geq 0.
 \end{aligned} \tag{26}$$

As it is shown, the KKY algorithm does not converge to the correct solution, while the computed solution of MKKY and two hybrid algorithms is feasible. Example 8 is solved by using Fmincon from MATLAB 7.0.4 and the absolute difference of the objective functions stated as Error. In Table 2 number of iterations, solution norms, optimal values of the objective function, Error, and the elapsed time for each algorithm have been given.

In Table 2 we show that MN-MKKY algorithm is more efficient than the other algorithms comparing the number of iterations and elapsed time.

In Figure 1, for various tolerances, the number of iterations is compared for different algorithms. It is observed that the MN-MKKY algorithm has better performance for each required tolerance.

*Example 9.* Consider Example 2. Table 2 shows the computed solution for four different algorithms. From Figure 1,

it is obvious that MN-MKKY algorithm is the best for this example.

*Example 10.* Consider the quadratic convex problem:

$$\begin{aligned} \min \quad & f(x) = x_1^2 + x_2^2 - 8x_2 + 8 \\ \text{s.t.} \quad & x_1 + 2x_2 + x_3 = 4, \\ & x_1, x_2, x_3 \geq 0. \end{aligned} \quad (27)$$

*Example 11.* Consider the nonlinear convex problem:

$$\begin{aligned} \min \quad & f(x) = x_1^2 + 2x_2^5 - x_1 \\ \text{s.t.} \quad & x_1 + 2x_2 - x_3 = 1, \\ & -3x_1 + x_2 + x_4 = -2, \\ & x_1, x_2, x_3, x_4 \geq 0. \end{aligned} \quad (28)$$

## 6. Conclusion

Having two ideas in our mind, (i) calculation of feasible solution in each iteration and (ii) the fact that the objective function value must decrease in each iteration with the fixed desired tolerance, this paper proposed three hybrid algorithms for solving nonlinear convex programming problem based on the interior point idea using various  $\beta$ 's of Karmarkar, Schrijver, and Malek-Naseri techniques. These methods have better performance than the standard Karmarkar algorithm, since in the latter algorithm one may not check the feasibility of solution in each iteration.

Our numerical simulation shows that the MN-MKKY algorithm has the best performance among the other algorithms. This algorithm uses less number of iterations to solve the general nonlinear optimization problems with linear constraints, since it uses the step length  $\beta$  of Malek-Naseri type.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## References

- [1] N. Karmarkar, "A new polynomial-time algorithm for linear programming," *Combinatorica*, vol. 4, no. 4, pp. 373–395, 1984.
- [2] H. Navidi, A. Malek, and P. Khosravi, "Efficient hybrid algorithm for solving large scale constrained linear programming problems," *Journal of Applied Sciences*, vol. 9, no. 18, pp. 3402–3406, 2009.
- [3] M. S. Bazzara, J. Jarvis, and H. D. Sherali, *Linear Programming and Network Flows*, John Wiley & Sons, New York, NY, USA, 1984.
- [4] A. T. Hamdy, *Operations Research*, Macmillan, New York, NY, USA, 1992.
- [5] R. M. R. Karp, "George Dantzig's impact on the theory of computation," *Discrete Optimization*, vol. 5, no. 2, pp. 174–185, 2008.
- [6] C. Roos, T. Terlaky, and J. Vial, *Theory and Algorithms for Linear Optimization*, Princeton University, 2001.
- [7] A. Schrijver, *Theory of Linear and Integer Programming*, John Wiley & Sons, New York, NY, USA, 1986.
- [8] A. Malek and R. Naseri, "A new fast algorithm based on Karmarkar's gradient projected method for solving linear programming problem," *Advanced Modeling and Optimization*, vol. 6, no. 2, pp. 43–51, 2004.
- [9] E. Tse and Y. Ye, "An extension of Karmarkar's projective algorithm for convex quadratic programming," *Mathematical Programming*, vol. 44, no. 2, pp. 157–179, 1989.
- [10] Z. Kebbiche and D. Benterki, "A weighted path-following method for linearly constrained convex programming," *Revue Roumaine de Mathématique Pures et Appliquées*, vol. 57, no. 3, pp. 245–256, 2012.
- [11] P. Fei and Y. Wang, "A primal infeasible interior point algorithm for linearly constrained convex programming," *Control and Cybernetics*, vol. 38, no. 3, pp. 687–704, 2009.
- [12] Z. Kebbiche, A. Keraghel, and A. Yassine, "Extension of a projective interior point method for linearly constrained convex programming," *Applied Mathematics and Computation*, vol. 193, no. 2, pp. 553–559, 2007.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

