Neural, Parallel, and Scientific Computations 18 (2010) 471-478

PARAMETER α VERSUS NUMBER OF ITERATIONS IN KARMARKAR'S ALGORITHM FOR LP

SYAMAL K. SEN AND JUAN F. AVILA

Department of Mathematical Sciences, Florida Institute of Technology
150 West University Boulevard, Melbourne, FL 32901–6975
sksen@fit.edu delagarita@gmail.com

ABSTRACT. Karmarkar's projective transformation algorithm for solving a linear program (LP) is the first polynomial time algorithm which was published in 1984. This algorithm implemented in Mathematica in this article has used a real parameter $0 < \alpha < 1$ whose value affects its convergence. That is, for different values of α, the number of iterations required to obtain a desired accuracy will be different. We study here if, for a given LP, there is an optimal α for which the number of iterations will be minimum subject to the precision of the computer. Also we investigate how α behaves from one LP to another.

1. INTRODUCTION

The exterior-point methods such as simplex methods [1,2,3] which are exponential-time in a worst case scenario [2] were the only methods which ruled the computational linear optimization scene for over three decades (up to mid-eighties). The Karmarkar projective transformation algorithm (KA) [4] is the first polynomial time interior-point method, published in 1984, that computes a solution to an LP in $O(n^{3.5})$ operations. This publication brought resurgence in the research activities in linear optimization. Since then there are several highly efficient polynomial-time algorithms that are reported in literature [5,6] and are currently being used quite extensively. However there still exists a scope to explore certain computational aspects in Karmarkar algorithm and observe its computational potential at least for academic interest. In this context we have studied the effect of the parameter α on the number of iterations in KA.

2. CONVERSION FROM LP TO KLP

A standard LP (constraints in an equality form) or any LP whose constraints are in an inequality form can be converted to the Karmarkar form of linear program

Second author is on leave from Universidad Nacional, Costa Rica.

Received April 15, 2010

(KLP) [8,9]. We provide the KLP for the following LP without derivation. For derivation refer [8].

$$\text{Max } z = c^t x \text{ s.t. } Ax \le b, \quad x \ge 0 \tag{1}$$

where $A = (a_{ij})$ is an $m \times n$ matrix, $b = (b_i)$ is an m-vector and $c = (c_j)$ and $x = (x_j)$ are n-vectors.

Let e_n^t be an n-vector $(1, 1, \ldots, 1)$, $\mathcal{Z}_{i\times j}$ the null matrix of order $i \times j$, and I_j the identity matrix of order j. Then the symbolic KLP is

$$\text{Min. } w = y_{2m+2n+3}, \text{ s.t. } \mathcal{A}y = 0, \quad y \ge 0, \quad y = e_n/n \text{ is feasible} \tag{2}$$

where $y = (y_1, y_2, \ldots, y_{2m+2n+3})$ and $\mathcal{A}$ is an $(m+n+3) \times (2m+2n+3)$ block matrix given by

$$\mathcal{A} = \left[\begin{array}{l|l|l|l|l|l|l}
c^t & -b^t & \mathcal{Z}_{1\times m} & \mathcal{Z}_{1\times n} & 0 & 0 & b^t e_m - c^t e_n \\ \hline
A & \mathcal{Z}_{m\times m} & I_m & \mathcal{Z}_{m\times n} & \mathcal{Z}_{m\times 1} & -b & b - Ae_n - e_m \\ \hline
\mathcal{Z}_{n\times n} & A^t & \mathcal{Z}_{n\times m} & -I_n & \mathcal{Z}_{n\times 1} & -c & c - A^t e_m + e_n \\ \hline
e_n^t & e_m^t & e_m^t & e_n^t & 1 & -k & k - (2m+2n+1) \\ \hline
e_n^t & e_m^t & e_m^t & e_n^t & 1 & 1 & 1
\end{array}\right]$$

where k is to be found/supplied such that the sum of the values of all the variables $\le k$.

Example (Symbolic KLP for a symbolic LP). Consider the LP where $A = (a_{ij})$, is an 2×4 matrix, $b^t = (b_1, b_2)$, $c^t = (c_1, c_2, c_3, c_4)$. Convert this LP to a KLP (Karmarkar Form of Linear Program) in matrix-vector form.

Solution In this case we have that $m = 2$ and $n = 4$. Thus Karmarkar form (KLP) for this case is Min y_{15}, s.t., $\mathcal{A}y = 0, y \ge 0$ where $y^t = [y_1, y_2, \cdots, y_{15}]$ and $\mathcal{A}$ is a 9×15 matrix given by

$$\mathcal{A} = \left[\begin{array}{rrrr|rr|rr|rrrr|r|r|r}
c_1 & c_2 & c_3 & c_4 & -b_1 & -b_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & b_1 + b_2 - c_1 - c_2 - c_3 - c_4 \\ \hline
a_{11} & a_{12} & a_{13} & a_{14} & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -b_1 & -a_{11} - a_{12} - a_{13} - a_{14} + b_1 - 1 \\
a_{21} & a_{22} & a_{23} & a_{24} & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -b_2 & -a_{21} - a_{22} - a_{23} - a_{24} + b_2 - 1 \\ \hline
0 & 0 & 0 & 0 & a_{11} & a_{21} & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -c_1 & -a_{11} - a_{21} + c_1 + 1 \\
0 & 0 & 0 & 0 & a_{12} & a_{22} & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -c_2 & -a_{12} - a_{22} + c_2 + 1 \\
0 & 0 & 0 & 0 & a_{13} & a_{23} & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -c_3 & -a_{13} - a_{23} + c_3 + 1 \\
0 & 0 & 0 & 0 & a_{14} & a_{24} & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -c_4 & -a_{14} - a_{24} + c_4 + 1 \\ \hline
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -k & k - 13 \\ \hline
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{array}\right]$$

Example (Numerical KLP for a numerical LP). Consider a LP where $A = \begin{bmatrix} 1 & 3 & 1 & 0 \\ 0 & 2 & 5 & 3 \end{bmatrix}$,

$b = \begin{bmatrix} 1 \\ 6 \end{bmatrix}$, $c = \begin{bmatrix} 2 \\ 2 \\ 9 \\ 7 \end{bmatrix}$ then write the corresponding KLP

Solution. The corresponding KLP is

$$\text{Min } y_{15}, \text{ s.t.}, \mathcal{A}y = 0, y \geq 0, \quad y = e_{15}/15 \text{ is feasible}$$

where $w^t = [y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10}, y_{11}, y_{12}, y_{13}, y_{14}, y_{15}]$ and $\mathcal{A}$ is 9×15 matrix given by

$$\mathcal{A} = \left[\begin{array}{cccc|cc|cc|cccc|c|c|c}
2 & 2 & 9 & 7 & -1 & -6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -13 \\ \hline
1 & 3 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -5 \\
0 & 2 & 5 & 3 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -6 & -5 \\ \hline
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -2 & 2 \\
0 & 0 & 0 & 0 & 3 & 2 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -2 & -2 \\
0 & 0 & 0 & 0 & 1 & 5 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -9 & 4 \\
0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -7 & 5 \\ \hline
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -24 & 11 \\ \hline
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{array}\right]$$

3. KARMARKAR ALGORITHM

Consider the general form of KLP: Min $z = c^t x$ subject to $Ax = 0$, $e_n^t x = 1$, $x \geq 0$, $c = e_n/n$ is feasible, minimal z-value $= 0$.

We assume that a feasible solution having a minimal z-value $< \epsilon$ (ϵ is a small positive value compared to the average element of A, b and c) is acceptable. The KA is then as follows.

(S_1) Input A, b, c, m, n. Set n-vector $e = [1\ 1 \ldots 1]^t$.

(S_2) Set the feasible point (solution) $x^0 = e/n$, the iterate $k = 0$.

(S_3) If $c^t x^k < \epsilon$ then stop else go to Step 4.

(S_4) Compute the new point (an n-vector) y^{k+1} in the transfomed n-dimensional unit simplex S (S is the set of points y satisfying $e^t y = 1$, $x \geq 0$) given by $y^{k+1} = x^0 - \alpha \dfrac{c_p}{\sqrt{n(n-1)}\,||c_p||}$ where $c_p = \left(I_n - P^t(PP^t)^+ P\right)\left[\operatorname{diag}\left(x^k\right)\right] c$, $P = \begin{bmatrix} A\left[\operatorname{diag}\left(x^k\right)\right] \\ e^t \end{bmatrix}$, $0 < \alpha < 1$. $\alpha = 0.25$ is known to ensure convergence. P is the$(m+1) \times n$ matrix whose last row e^t is a vector of ls. $(PP^t)^+$ is the pseudo-inverse of the matrix PP^t.

(S_5) Compute now a new point x^{k+1} in the original space using the Karmarkar Centring transformation to deterrnine the point corresponding to the point y^{k+1}: $x^{k+1} = \dfrac{q}{e^t q}$, $q = \left[\operatorname{diag}\left(x^k\right)\right] y^{k+1}$. Increase k by 1 and return to Step 3.

A Mathematica version program for the KA is presented below for the reader to readily check the algorithm for different kinds of LP including extreme ones and get a feel of it. No effort has been made to make the program more efficient so as to differ from the KA presented here. The inputs to this program are A, b, c, k (a parameter that differs from problem to problem), m and n. At the end of this program we use

Maximize command from Mathematica to compare with the one we obtain using KA. In order to conserve space we have eliminated comments, indentations, and we have written more than one statement per line.

```
"LP to KLP CONVERSION "; ClearAll["Global`*"]; LPΞA = ({{1, 2, 1}, {−4, −2, 3}}); LPΞb = {1, 2}; LPΞc =
{−2, −7, 2}; {m, n} = Dimensions[LPΞA]; k = Input[]; KLPΞA = Table[0, {i, 3 + m + n}, {j, 2m + 2n + 3}]; KLPΞB =
Table[0, {i, 3+m+n}]; KLPΞC = Table[0, {i, 2n+2m+3}]; sumΞLPΞA = Total[Transpose[LPΞA]]; sumΞAΞTranspose =
Total[LPΞA]; e = Table[1, {i, 2m + 2n + 3}]; For[i = 1, i<=(n), KLPΞA[[1, i]] = LPΞc[[i]]; i++]; For[i =
(1), i<=(m), KLPΞA[[1, n + i]] = −LPΞb[[i]]; i++]; For[i = (1), i<=(n), For[j = (1), j<=(m), KLPΞA[[1 + j, i]] =
LPΞA[[j, i]]; KLPΞA[[1 + m + i, n + j]] = LPΞA[[j, i]]; KLPΞA[[1 + j, n + m + j]] = 1; KLPΞA[[1 + m + i, 2m + n + i]] =
−1; ; j++]; i++]; KLPΞA[[1, 2m + 2n + 3]] = Total[LPΞb] − Total[LPΞc]; For[i = (1), i<=(m), KLPΞA[[1 + i, 2m + 2n +
2]] = −LPΞb[[i]]; KLPΞA[[1 + i, 2m + 2n + 3]] = −(sumΞLPΞA[[i]] + 1 + KLPΞA[[1 + i, 2m + 2n + 2]]); i++]; For[i =
(1), i<=(n), KLPΞA[[1 + m + i, 2m + 2n + 2]] = −LPΞc[[i]]; KLPΞA[[1 + m + i, 2m + 2n + 3]] =
−(sumΞAΞTranspose[[i]] − 1 + KLPΞA[[1 + m + i, 2m + 2n + 2]]); ; i++]; KLPΞA[[2 + m + n]] =
Table[1, {i, Dimensions[KLPΞA][[2]]}]; KLPΞA[[2+m+n, 2m+2n+2]] = −k; KLPΞA[[2+m+n, 2m+2n+3]] = −(2n+
2m + 1 − k); KLPΞA[[3 + m + n]] = e; KLPΞB[[3 + m + n]] = 1; KLPΞC[[2n + 2m + 3]] = 1; Print[MatrixForm[KLPΞA]];
"KARMARKAR PROCEDURE"; b = KLPΞB; c = KLPΞC; A = KLPΞA; itern = 3000; {m, n} =
Dimensions[A]; e = Table[1, {i, n}]; x0 = e/n; x = x0; alp = 0.7; IM = IdentityMatrix[n]; eps =
0.00005(Total[Total[Abs[A]]] + Total[Abs[b]] + Total[Abs[c]])/(mn + m + n); n2 = Sqrt[(n(n − 1))]; Q = {}; For[j =
(1), j ≤ (itern), If[(k + 1)c.x < eps, Print["eps = ", eps]; Print["Iteration no.", j]; Print["x=", x]; Break[]]; P =
Append[A.DiagonalMatrix[x], e]; cp = (IM − Transpose[P].PseudoInverse[P.Transpose[P]].P).DiagonalMatrix[x].c; y =
x0 − alpcp/(n2Norm[cp]); q = DiagonalMatrix[x].y; x = q/(e.q); j++]; xt =
(k + 1)x; Print["The iteration No. is ", j]; Print["The true solution is: "]; Print["xt = ", Take[xt, Dimensions[LPΞA][[2]]]];
Print["Obj. value =", Take[xt, Dimensions[LPΞA][[2]]].LPΞc]; Print["The epsilon is: ", eps];
"SOLUTION WITH MATHEMATICA"; {m, n} = Dimensions[LPΞA]; mx = Array[xx, n]; SS = {}; aux =
LPΞA.mx; For[i = 1, i ≤ m, SS = Join[SS, {aux[[i]] ≤ LPΞb[[i]]}]; i++]; For[i = 1, i ≤ n, SS = Join[SS, {xx[i] ≥
0}]; i++]; SS; Maximize[LPΞc.mx, SS, mx]
```

---

4. NUMERICAL EXPERIMENTS: PARAMETER α VERSUS NUMBER OF ITERATIONS

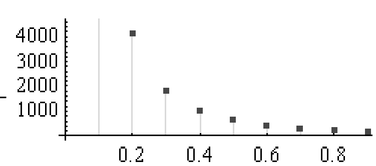
We have experimented with several distinct LPs including those with unique solution, infeasibility, multiple solutions, and unbounded solution. However, to conserve space we have included the following typical numerical examples to illustrate how the values of α influence the number of iterations for a specified accuracy.

1. *Unique solution* (i) Consider the LP Max $z = c^t x$ s.t. $Ax \leq b, x \geq 0$, where

$$A = \begin{pmatrix} -\frac{2}{3} & -\frac{11}{3} & \frac{22}{3} & \frac{5}{3} & -\frac{19}{9} \\ -\frac{9}{2} & -\frac{23}{4} & -\frac{5}{2} & \frac{17}{8} & 7 \\ -\frac{23}{4} & -\frac{26}{3} & -\frac{19}{3} & \frac{28}{3} & -\frac{29}{3} \end{pmatrix}, b^t = \left(\frac{929}{18}, -\frac{607}{24}, -\frac{3071}{36}\right), c^t = \left(-\frac{22}{3}, -\frac{39}{4}, \frac{15}{4}, -\frac{9}{8}, -\frac{31}{4}\right)$$

After using KA with $k = 62$, we have the following table which records the value of α versus the number of iterations. Besides, we have included a graph correspondig to this information.

α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
No. Iter	16489	4108	1815	1014	642	443	323	248	195



The optimal solution vector is
$x^t = (0.000228791, 2.22513, 8.51096, 0.0000686293, 1.25354)$and objective function value is $z = 0.504345$. The exact solution according to Mathematica 6.0 is

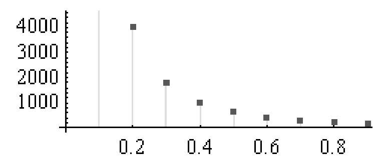
$$z^* = \frac{332593}{653648},\ x^{*t} = \left(0, \frac{272645}{122559}, \frac{1390827}{163412}, 0, \frac{51228}{40853}\right) \approx (0, 2.2246, 8.51117, 0, 1.25396)$$

(ii) Consider the LP Max $z = c^t x$ s.t. $Ax \le b, x \ge 0$, where

$$A = \begin{pmatrix} -\frac{57}{7} & 4 & -\frac{23}{3} & \frac{13}{2} & -\frac{4}{3} \\ -\frac{10}{3} & -9 & -\frac{3}{2} & \frac{31}{4} & \frac{29}{3} \\ \frac{39}{5} & \frac{15}{4} & \frac{15}{4} & -\frac{8}{3} & \frac{44}{5} \end{pmatrix},\ b^t = \left(-\frac{28013}{840}, -\frac{5197}{240}, \frac{1267}{15}\right),\ c^t = \left(-\frac{25}{3}, 6, -\frac{22}{3}, -\frac{25}{4}, 3\right)$$

After using KA with $k = 64$, we have the following table which records the value of α versus the number of iterations. Besides, we have included a graph correspondig to this information.

α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
No. Iter	16264	4050	1790	998	633	434	318	230	193



The optimal solution vector is
$x^t = (0.000195737, 5.86614, 6.67014, 0.0000707261, 4.25589)$ and objective function value is $z = -0.951884$. The exact solution according to Mathematica 6.0 is

$$z^* = -\frac{461603}{486360},\ x^{*t} = \left(0, \frac{2852989}{486360}, \frac{1081361}{162120}, 0, \frac{46003}{10808}\right) \approx (0, 5.866, 6.67013, 0, 4.25638)$$

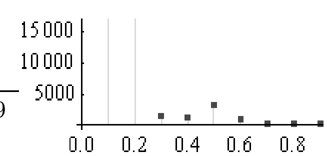
We did more than 10 experiments for the unique solution case and $\alpha = 0.9$ was the value which gave the best perfomance in terms of less number of iterations.

2. *Infeasible LP* Consider the LP Max $z = c^t x$ s.t. $Ax \le b, x \ge 0$, where

$$A = \begin{pmatrix} \frac{3}{2} & \frac{6}{5} & \frac{13}{2} & -3 & -\frac{29}{5} \\ -\frac{21}{4} & 7 & \frac{17}{2} & \frac{23}{4} & -\frac{27}{4} \\ \frac{21}{4} & -7 & -\frac{17}{2} & -\frac{23}{4} & \frac{27}{4} \end{pmatrix}, b^t = \left(-\frac{823}{60}, \frac{209}{15}, -20\right), c^t = \left(-\frac{25}{4}, -\frac{14}{3}, -\frac{25}{3}, \frac{7}{3}, -\frac{19}{2}\right)$$

After using KA with $k = 66$, we have the following table:

α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
No. Iter	18427	≥ 25000	1604	1175	3323	950	352	267	209



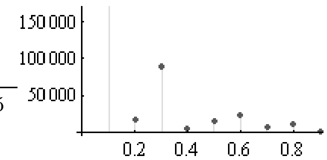
KA indicates the infeasibility of the LP. Mathematica 6.0 also indicates the infeasibility. Notice that with $\alpha = 0.9$ we get the minimum number of iterations.

3. *Unbounded LP* (i) Consider the LP Max $z = c^t x$ s.t. $Ax \le b, x \ge 0$, where

$$A = \begin{pmatrix} -\frac{3}{4} & -\frac{15}{2} & \frac{11}{2} & -\frac{22}{3} \\ \frac{1}{2} & \frac{31}{5} & -\frac{19}{5} & -\frac{33}{4} \\ 8 & \frac{37}{4} & \frac{14}{3} & -\frac{16}{3} \end{pmatrix}, b^t = \left(-\frac{203}{2}, \frac{753}{10}, \frac{3739}{48}\right), c^t = \left(-\frac{31}{6}, -\frac{19}{2}, \frac{34}{5}, 4\right)$$

After using KA with $k = 52$, we have the following table which records the value of α versus the number of iterations. Besides, we have included a graph correspondig to this information.

α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
No. Iter	≥ 200000	17547	88990	5704	14870	23615	7020	11092	455

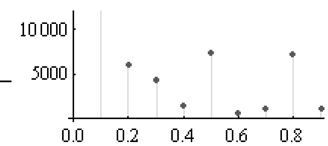


(ii) Consider the LP (taken from [1]) Max $z = c^t x$ s.t. $Ax \leq b, x \geq 0$, where

$$A = \begin{pmatrix} 2 & -4 & -1 & 1 \\ 1 & 1 & 2 & -3 \\ 1 & -1 & -4 & 1 \end{pmatrix}, b^t = (8, 10, 3), c^t = (3, 2, -1, 1)$$

After using KA with $k = 32$, we have the following table which records the value of α versus the number of iterations. Besides, we have included a graph correspondig to this information.

α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
No. Iter	24879	6112	4325	1561	7349	705	1222	7303	1074



In both KA and Mathematica 6.0 the elements of the solution vector grow indefinitely in magnitude. Thus the LPs are unbounded.

4. *Multiple Solution LP* (i) Consider the LP Max $z = c^t x$ s.t. $Ax \leq b, x \geq 0$, where

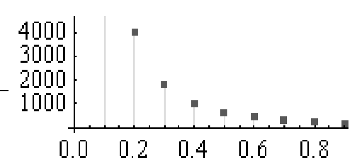
$$A = \begin{pmatrix} -\frac{9}{5} & -2 & \frac{35}{4} & \frac{19}{4} & \frac{19}{3} \\ -\frac{5}{2} & -\frac{29}{3} & -\frac{19}{3} & 5 & \frac{26}{3} \\ -\frac{29}{3} & -\frac{1}{3} & -\frac{23}{4} & -\frac{41}{6} & -\frac{1}{4} \end{pmatrix}, b^t = \left(\frac{43}{48}, -\frac{343}{12}, -\frac{11441}{144}\right), c^t = \left(-\frac{9}{5}, -2, \frac{35}{4}, \frac{19}{4}, \frac{19}{3}\right)$$

Using Mathematica 6.0 we find $z^* = 43/48 \approx 0.895833$ and more than one solution, for example:

$$x_1^t = \left(\frac{3139201}{432126}, \frac{120091}{4321260}, \frac{6927649}{4321260}, 0, 0\right), x_2^t = \left(\frac{4150045}{4608}, \frac{7}{512}, \frac{7}{128}, \frac{3}{64}, 256\right)$$

After using KA with $k = 60$, we have the following table which records the value of α versus the number of iterations. Besides, we have included a graph correspondig to this information.

α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
No. iter	16227	4053	1797	1008	644	444	323	250	193



In this case we arrive at the (approximate) solution

$$x^t = (7.85132, 4.79658, 0.656678, 1.61743, 1.76767), \quad z = 0.898363$$

(ii) Consider the LP Max $z = c^t x$ s.t. $Ax \leq b, x \geq 0$, where

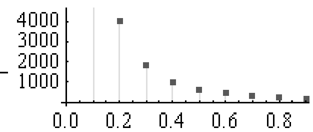
$$A = \begin{pmatrix} 1 & \frac{27}{4} & -\frac{11}{2} & -\frac{1}{9} & -\frac{13}{2} \\ -\frac{20}{3} & -\frac{3}{2} & \frac{19}{2} & \frac{15}{2} & \frac{8}{7} \\ \frac{5}{2} & \frac{20}{3} & 5 & -\frac{39}{5} & -10 \end{pmatrix}, b^t = \left(\frac{13}{24}, \frac{319}{126}, -\frac{2003}{30}\right), c^t = \left(1, \frac{27}{4}, -\frac{11}{2}, -\frac{1}{9}, -\frac{13}{2}\right)$$

Using Mathematica 6.0 we find $z^* = 13/24 \approx 0.541667$ and more than one solution, for example:

$$x_1^t = \left(\frac{71592746}{6239739}, 0, 0, \frac{21437571}{2079913}, \frac{37579711}{24958956}\right), x_2^t = \left(\frac{281725}{24576}, \frac{5}{128}, \frac{7}{2048}, \frac{84321}{8192}, \frac{1579}{1024}\right)$$

After using KA with $k = 60$, we have the following table which records the value of α versus the number of iterations. Besides, we have included a graph correspondig to this information.

α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
No. iter	16157	4039	1794	1008	645	447	328	251	198



We arrive at the solution $x^t = (13.0829, 8.76017, 1.52122, 9.59719, 9.5756)$, $z = 0.539584$.

5. CONCLUSIONS

When we convert an LP to KLP the dimension of the $m \times n$ matrix A increases to $(m + n + 3) \times (2m + 2n + 3)$. If $m = 15$ and $n = 25$, then the matrix A in LP would require 375 locations while the correspondig matrix $\mathcal{A}$ in KLP would be needing 3569 locations. This implies that KLP requires nearly 10 times the locations for this dimension. In general, we would be needing over 8 times the storage locations in KLP for a given LP in an inequality form, where m is close to n. The following table depicts the storage locations required by KA in terms of the size of given matrix A.

n/m	5	10	15	20	25	30	35	40
5	11.96							
10	11.88	9.89						
15	13.19	9.89	9.24					
20	14.84	10.4	9.25	8.92				
25	16.63	11.1	9.52	8.93	8.73			
30	18.49	11.9	9.92	9.1	8.74	8.61		
35	20.39	12.75	10.4	9.36	8.86	8.61	8.52	
40	22.32	13.65	10.92	9.69	9.04	8.7	8.52	8.46

In both simplex as well as Karmarkar methods the objective function value will be monotonically decreasing with increasing number of iterations in general. But there is one difference. In simplex method the basic variables will go on changing at each iteration while it will not be so in KA. Further unlike simplex methods which produce an optimal basic solution, KA could produce an optimal non-basic solution. So the KA has the potential to allow us to detect basic variables after sufficient number of iterations (without completing all the iterations required to get the desired accuracy of the solution vector).

The study of the effect of the parameter α on the number of iterations in KA is certainly of academic interest. However this algorithm still has the potential to be appropriately modified by some future researcher so that the modified version could be competitive with the current efficient polynomial time interior point methods.

From the several examples that we have experimented on, we see that α for a unique optimal solution case as well as a multiple solution case is close to 1, that minimizes the number of iterations assuming sufficiently large precision of the computer. However, for infeasible and unbounded solution cases we have observed some kind of irregular oscillations in the number of iterations.

REFERENCES

[1] Bazaraa, M. S., Jarvis, J. J., and Sherali, H., Linear Programming and Network Flows, 2nd Edition, John Wiley & Sons N.Y. 1990.

[2] Beale, E. M. L., Cycling in the dual simplex algorithm, Naval Research Logrstics Quarterly 2 1955, 269–275.

[3] Vajda, S., Theory of Linear and Non-liner Programming, Longman, London, 1974.

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

[5] Barnes, E. R., A variation on Karmarkar's algorithm for solving linear programming problems, Mathematical Programming, 36 1986, 174-182.

[6] Terlaky, T., Interior-point Methods in Mathematical Programming, Springer Berlin, 1996.

[7] Lakshmikantham, V.; Maullo, A. K.; Sen, S. K.; Sivasundaram, S., Solving linear programming problems exactly, Appl. Maths. Comput. 81, 1997, 69-80.

[8] S. K. Sen and S. Sen, Karmarkar Form of Linear Program and Algorithm: Precise Presentation Proc. 45th International Congress of the Indian Society of Theoretical and Applied Mechanics, India, 2000, 88–96.

[9] Winston, W. L., Operations Research: Applications and Algorithms, Wadsworth Publishing Company, Duxbury, Belmont, California, 2004.