INSTITUTE FOR POLICY SCIENCE RESEARCH REPORT NO. 87-B-4

AN IMPLEMENTATION OF A REVISED KARMARKAR'S METHOD

by

K. Tone

March 25, 1987

An Implementation of a Revised Karmarkar's Method

Kaoru Tone

Graduate School for Policy Science, Saitama University
Urawa, Saitama 338, Japan

Abstract
We will show a variant of the Karmarkar's algorithm for LPs with sparse matrices. We deal with the standard form LP. Starting from an initial interior point, one interation of our method consists of choice of a basis, factorization of the basis, optimality test, reduced gradient, conjugate gradient method and determination of the next point of iterate. A combination of the reduced gradient and the conjugate gradient method is used for generating the steepest descent direction of the transformed objective function. Bases which are maintained and updated throughout the iterations are effectively utilized. As a basis, we choose the linearly independent columns of the coefficient matrix corresponding to the decreasing order of the variables. The basis is then factorized in the LU-form which is used in the computations throughout the iteration. Preliminary numerical experiments will be reported. Emphasis is laid on the implementational issues of the sparse basis.

Key words: Linear programming, logarithmic barrier function, interior point method, Karmarkar's algorithm, LU-factorization, sparse basis, reduced gradient, conjugate gradient method, implementation, numerical example

1. Outline of the Revised Karmarkar's Method

We will introduce the outline of the affine version of the revised Karmarkar's method (RK method) along with explanations of notations and symbols. (For further details, see [7] and [10].) The problem we are going to solve is:

(L) $\min c^T x$ (1.1)

subject to $Ax=b,\ x \geqq 0$,

where A is an (m,n) matrix.

The feasible region X of (L) is defined as

$$X=\{x\in R^n:\ Ax=b,\ x\geqq 0\}. \tag{1.2}$$

We impose three assumptions on (L):

A1) X is nonempty and an interior point $x^0\in X$ $(x^0>0)$ is known,

A2) X is bounded

and

A3) rank(A)=m.

A logarithmic barrier-function method to (L) is:

$$\text{(L')}\ \min F(x)=c^Tx-\mu\sum_{j=1}^{n}\ln(x_j) \tag{1.3}$$

subject to Ax=b,

where μ $(\mu>0)$ is the barrier parameter.

If $x^*(\mu)$ is the solution of (L'), then $x^*(\mu)\rightarrow x^*$(the solution of (L)) as $\mu\rightarrow 0$ (see Fiacco and McCormick [2]).

Based on an interior point x^0 of X, we define the affine transformation from x to y as

$$y=D^{-1}x,\quad D=\mathrm{diag}(x^0_j) \tag{1.4}$$

Notice that x^0 corresponds to $e=(1,\ldots,1)^T$ by D.

We have the transformed problems of (L) and (L') as

$$\text{(M)}\quad \min \bar{c}^Ty \tag{1.5}$$

subject to $\bar{A}y=b,\ y\geqq 0$

where $\bar{A}=AD,\ \bar{c}=DC,$ (1.6)

and

$$\text{(M')}\ \min G(x)=\bar{c}^Ty-\mu(\Sigma\ln(y_j)+\ln(\det(D))) \tag{1.7}$$

subject to $\bar{A}y=b$.

The feasible region of (M) is:

$$Y=\{y\in R^n:\bar{A}y=b,\ y\geqq 0\}. \qquad (1.8)$$

The steepest descent direction d of the $g(=\nabla G(y))$ on Y is given as

$$d=-P_Y g \qquad (1.9)$$

where

$$P_Y=I - \bar{A}^T(\bar{A}\bar{A}^T)^{-1}\bar{A} \qquad (1.10)$$

is the projection matrix, and

$$g=Dc-\mu e. \qquad (1.11)$$

We move from e along the direction d to obtain

$$a=e+\alpha\sigma^* d \qquad (1.12)$$

where

$$\sigma^*=\sup\{\sigma\in R: e+\sigma d\geqq 0\} \qquad (1.13)$$

and

$$\alpha\in(0,1). \qquad (1.14)$$

The transformation (1.4) is applied inversely to return a to an interior point of X. Thus, we have

$$x^1=Da. \qquad (1.15)$$

x^1 will replace x^0 until the iterations converge.

Another look at the orthogonal projection (1.9) is as follows: A least squares problem equivalent to (1.9) is

$$\text{(LSQ)}\quad \min \|g-z\|^2 \qquad (1.16)$$

$$\text{subject to } \bar{A}z=0. \qquad (1.17)$$

Furthermore, by introducing a bais $\bar{B}$ of $\bar{A}(=[\bar{B}\mid\bar{N}])$, the constraint (1.15) is decomposed into

$$\bar{A}z=\bar{B}z_B+\bar{N}z_N=0 \qquad (1.18)$$

and hence

$$z_B=-\bar{B}^{-1}\bar{N}z_N. \tag{1.19}$$

Then, (LSQ) is reduced to an unconstrained least squares problem:

$$\text{(LS)} \quad \min \| \tilde{g} - \begin{pmatrix} -\bar{\bar{N}} \\ I \end{pmatrix} z_N \|^2 \tag{1.20}$$

where

$$\tilde{g}=[0,\tilde{g}_N]^T \tag{1.21}$$

$$\tilde{g}_N{}^T=g_N{}^T-g_B{}^T\bar{\bar{N}} \tag{1.22}$$

$$\bar{\bar{N}}=\bar{B}^{-1}\bar{N}. \tag{1.23}$$

The RK method solves the problem (LS) by the conjugate gradient methods using $\tilde{g}_N$ as the starting value of z_N. Th CG process converges fast to the optimal solution $z_N{}^*$, if the basis $\bar{B}$ becomes nearly optimal and the optimal solution of (L) is nondegenerate.

Then, we have

$$d= -\begin{pmatrix} -\bar{\bar{N}} \\ I \end{pmatrix} z_N{}^*$$

[Remark 1] Although the affine version explained above is simpler than the projective version (see [7]) in that no lower bound to the optimal value to (L) is needed, it has no guarantee of polynomial time convergence.

[Remark 2] The logarithmic barrier function is first suggested by Frisch([3]). Since we consider it as an extension of our previous works([7],[10]), we name it the affine version of the RK method rather than the logarithmic barrier function method.
It is quite similar to the method by Gill et al.([4]) but is

different in the implementation.

2. Impementation of the RK Method with Sparse Matrices

The RK method starts from the scaling of the input data A,b and c and the initialization for Phase 1 follows. Then we proceed iteratively until convergence occurs. One iteration consists of

(1) Choice of the basis

(2) LU-decomposition of the basis

(3) Optimality test

(4) Computation of the reduced gradient

(5) Conjugate gradient method

and

(6) Updata of the iterate x.

Our implementation deals with MPS format data. Non-zero elements are stored in one dimensional arrays with appropriate pointers to segment the data. We will explain the main parts of the implementation.

a. Scaling

The rows are scaled first by the largest coefficient (absolute value) and then columns. Also the vectors b and c are scaled so that the largest elements in absolute value are 1s.

b. Initialization and Phase 1/Phase 2

In order to have an initial interior point, we use the Phase 1 problem:

$$\text{(L1)} \qquad \min\ (c^T x + M x_0) \qquad \text{subject to } Ax + (b - Ae)x_0 = b, \quad x \geqq 0,\ x_0 \geqq 0 \tag{2.1}$$

where M is a sufficiently large positive number (penalty).
It is easy to see that $x=e$ and $x_0=1$ is an initial interior point to (L1). We start Phase 1 of the RK method from this value. If x_0 comes less than 0, then we interpolate the value of x corresponding to $x_0=0$, which will be used as the starting interior point of the Phase 2 problem, after deletion of x_0.

<u>c. Basis and its LU-Decomposition</u>

Since the bases play a big role in the RK method, choice of a basis at each iteration is crucial to the efficiency of the algorithm. If the problem has a nondegenerate optimal solution, it is quite natural to choose m linearly independent columns of A corresponding to the decreasing order of elements of the iterate x^0. This choice will make the convergence of the RK method fast. The following rule is applied in the implementation.

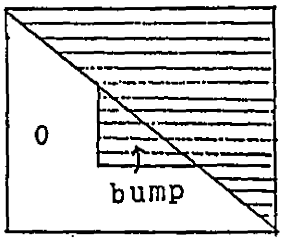
We assume that the indices of the iterate x^0 are reordered so that

$$x^0{}_1 \geqq x^0{}_2 \geqq \ldots\ldots \geqq x^0{}_n (>0). \tag{2.2}$$

(c1) If there are slack variables among $\{x_1,\ldots,x_m\}$, they are included into the basis. Let s be the number of slack variables thus chosen.

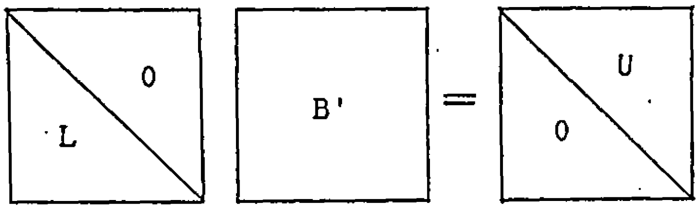
(c2) We make a 'tentative' basis B by adding (m-s) columns of A to the set, according to the index order, which are not apparently linearly dependent each other. Thus, B has m candidates of pivots but is not assured to be nonsingular.

(c3) The Reid's preconditioning ([8]) is applied to B so as to curb 'fill in' in the LU-decomposition. Thus, we have:

B'=QBR=  (2.3)

where Q and R are permutation matrices.

(c4) LU-decomposition is applied. However, since linear independence is not assured in 'bump', zero pivots may occur in the process. In such a case, we put a 'dummy 1' (actually a unit vector) at the pivot position and continue the decomposition, which results in:



(c5) If there are 'dummy 1' columns after the decomposition is over, we exile the columns from the basis by using the η vectors(a technique in the simplex method). Thus B^{-1} is usually expressed by Q, R, L, U and η vectors.

d. Computation of the Reduced Gradient

A simple calculation shows

$$\tilde{g}_N^T=(c_N^T-c_B^T(B^{-1}N))D_N-\mu(e_N^T-e_B^T D_B^{-1}B^{-1}ND_N). \quad (2.4)$$

e. Conjugate Gradient Method

We use Hestenes and Stiefel's version of the conjugate gradient method for least squares problems ([5]). If we rewrite the system (LS) as

$$\min \; \| h-Pz \|^{2}, \qquad (2.5)$$

then the algorithm is as follows:

```
1. $z^0=h_N$
2. $r^0=h-Pz^0$
3. $s^0=P^T r^0$
4. $\gamma^0=\| s^0 \|_2^2$
5. $p^1=s^0$
6. For k=1,2,... repeat the following:
    a. $q^k=Pp^k$
    b. $\alpha^k=\gamma^{k-1}/\| q^k \|_2^2$
    c. $z^k=z^{k-1}+\alpha^k p^k$
    d. $r^k=r^{k-1}-\alpha^k q^k$
    e. $s^k=P^T r^k$
    f. $\gamma^k=\| s^k \|_2^2$
    g. $\beta^k=\gamma^k/\gamma^{k-1}$
    h. $p^{k+1}=s^k+\beta^k p^k$.
```

<u>f. Choice of α</u>

α is chosen from [0.85,0.95].

<u>g. Reduction of μ</u>

$\mu^0=0.1$ and $\mu^{k+1}=\beta\,\mu^k$ ($\beta=0.1$) for k=0,1,2,...

<u>h. Changeover to the Simpelx Method</u>

We define the infeasibility of a basis as the sum of primal infeasibility and dual infeasibility. The infeasibility usually becomes small at the last few steps of the iterations. A changeover to the simplex method is efficient to obtain a fast convergence to an optimal basis if the infeasibility becomes less than a tolerance, say (m+n)/25.

We use a perturbation of RHS and a parametric LP as follows:

Let $\Delta b=-\sum_{i\in I_-} \bar{b}_i B_i$ (2.6)

where

$I_-=\{i: \bar{b}_i<0\}$ (2.7)

$\bar{b}_i=(B^{-1}b)_i$ (2.8)

and

$B_i=$ the i-th column of B (basis). (2.9)

If we perturb b by adding Δb, then B is primal feasible for the perturbed problem. Then we obtain an optimal solution of the perturbed problem by the primal simplex method and a parametric LP is applied to find an optimal solution of the original problem.

3. Sample Problem and Computational Results

The following is a variation of a scheduling production and inventory (multiperiod and staircase) problem by V. Chvatal [1]:

$$\min \sum_{j=1}^{k} (20y_j+8z_j+t_j) \tag{3.1}$$

subject to
$$x_j-x_{j-1}\leqq 800$$
$$-x_j+x_{j-1}\leqq 800$$
$$y_j\leqq 0.3x_j$$
$$z_{j-1}+x_j+y_j=d_j+z_j$$
$$t_j\geqq 15(x_j-x_{j-1})$$
$$t_j\geqq 21(x_{j-1}-x_j)$$

$x_0=5800$

$z_0=0$

$x_j, y_j, z_j, t_j \geqq 0$

for $j=1,\dots,k$.

We solved the problem for several ks with the demand $\{d_j\}(j=1,..,k)$ fixed randomly within a certain range. Table 1 shows the computational results along with comparisons with MINOS 3.0 and MPS2.

M	N	RK		MINOS 3.0		MPS2	
		ITR(1,2,3)	CPU(CG)	ITR(1,2)	CPU	ITR(1,2)	CPU
36	53	11(2,8,1)	0.69(0.33)	31(14,17)	1.47	17(14,3)	0.72
72	107	8(2,4,2)	1.63(0.98)	75(36,37)	3.00	52(39,13)	1.14
144	215	11(3,4,4)	4.19(2.43)	148(74,74)	7.06	94(80,14)	1.91
288	431	16(4,5,7)	13.74(7.96)	263(137,126)	17.81	187(164,23)	4.58
576	863	17(4,6,7)	47.26(30.34)	520(268,252)	56.98	383(337,46)	13.91

Table 1.

Notes to the Table 1:

1) M=no. of rows(=6k), N=no. of columns. Nos. of non-zero elements of the coefficient matrix A are 120(for M=36), 246(72), 498(144), 1002(288), and 2010(576). Densities of

non-zero elements are 11%(for M=36), 5.5%(72), 2.8%(144), 1.4%(288) and 0.7%(576).

2) RK:the numbers in RK are averages on ten random sample problems. All the iterations finished after having found an optimal basis. Double precision arithmetics were used for the experiments of the RK method. Th CG method stops when the relative change in the norm of residuals of the succeeding two steps becomes less than 10^{-4}(for M=36,72), 10^{-5}(144) 10^{-6}(288),and 10^{-8}(576).

3) "Fill in" in the LU-factorization of the RK method and η-vectors appeared rarely.

4) MINOS and MPS2: the numbers in MINOS and MPS2 are averages on two random sample problems.

5) ITR(1,2,3) means no. of iterations , Phase 1/Phase 2/Phase 3 iterations. Phase 3 iterations in RK mean those of the simplex method after the changeover. We switched to the simplex method if the infeasibility of the basis became less than (M+N)/25. Nos. of CG iterations are 27(M=36),44(72), 56(144), 96(288), and 184(576).

6) The values of α (step length ratio) in (1.12) are 0.95(M=36,72, 144,288),and 0.90(576).

7) CPU(CG): CPU(CG) time in seconds on a HITAC M-260D.

8) RK and MINOS 3.0 are written by Fortran while the language of MPS2 is unknown to the author.

[Remark 3] The above experiments were done just for a trial to see how the algorithm works. We need more extensive experiments to have any decisive conclusions on the efficiency of the method.

4. Observations and Future Research Subjects

In this paper, we showed an implementation of the revised Karmarkar's method with emphasis on the sparse basis handlings. From the very limited experiences, we observe the followings:

(1) The CG method becomes unstable for large m. Some preconditionings of the CG method will be needed for such cases.

(2) The choice of α (the step length) in (1.12) and μ (the barrier parameter) in (1.3) is sensitive to the performances of the algorithm. Some other works [4], [9] on the subjects should be consulted in the future implementation.

(3) More than 60 percent of the CPU time is spent in the CG computations. Zenios and Mulvey [11] have succeeded in the vectorization of the CG computation for nonlinear network programming problems. Their implementation deals with the sparse basis and suggests the future possibility of vectorization of our algorithm.

(4) Changeover to the simplex method is effective in speeding up the convergence to the optimal basis and in saving the CPU time.

(5) Efficient updating methods of the sparse basis factorization along with the problem reduction by finding the basic or nonbasic variables are future research subjects.

(6) Protections against degenerate optimal solutions and null variables should be crucial subjects to be studied further. We need more theoretical and empirical studies.

References

[1]V.Chvatal, Linear Programming (Freeman,1983).

[2]A.V.Fiacco and G.P.McCormick,Nonlinear Programming:Sequential Unconstrained Minimization Techniques (John Wiley and Sons, 1968).

[3]K.R.Frisch,"The logarithmic potential method of convex programming," University Institute of Economics (Oslo, Norway,1955).

[4]P.E.Gill,W.Murray,M.A.Saunders,J.A.Tomlin, and M.H.Wright, "On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective methods," Mathematical Programming, 36(1986)183-209.

[5]M.T.Heath,"Numerical methods for large sparse linear least squares problems," SIAM J.Sci.Stat.Comput., 5(1984)497-513.

[6]N.Karmarkar,"A new polynomial-time algorithm for linear programming," Combinatorica, 4(1984b)373-395.

[7]M.Kojima and K.Tone,"An efficient implementation of Karmarkar's new LP algorithm," Research Reports B-180, Dept. of Information Sciences, Tokyo Institute of Technology, 1986.

[8]J.K.Reid,"A sparsity exploiting variant of the Bartels-Golub decomposition for linear programming bases," Mathematical Programming, 24(1982)55-69.

[9]J.Renegar,"A polynomial-time algorithm, based on Newton method, for linear programming," MSRI 07118-86, Mathematical Sciences Research Institute, Berkeley, California, 1986.

[10]K.Tone and M.Kojima,"The revised Karmarkar algorithm and its computational experiments,"(in Japanese) Proceedings of the 7th Mathematical Programming Symposium, Japan,1986.

[11]S.A.Zenios and J.M.Mulvey,"Nonlinear network programming on vector supercomputers: a study on the CRAY X-MP," Report EES-85-13, Princeton University, 1986.