Journal of the Korean
OR/ MS Society
Vol. 19, No. 3, December 1994

A Formulation of Hybrid Algorithm for Linear Programming

Koonchan Kim*

Abstract

This paper introduces an effective hybridization of the usual simplex method and an interior point method in the convergent framework of Dembo and Sahi. We formulate a specific and detailed algorithm(HYBRID) and report the results of some preliminary testing on small dense problems for its viability. By piercing through the feasible region, the newly developed hybrid algorithm avoids the combinatorial structure of linear programs, and several other interesting and important characteristics of this algorithm are also discussed.

1. Introduction

Besides the well-known classical method of simplex by Dantzig[3] for solving linear programming problems, especially in operations research and management science, there are a number of interior point methods that have been developed since 1984, beginning with the projective scaling method of Karmarkar[7]. These interior point methods can be divided largely into three categories and they are *projective scaling method*[7] pioneered by Karmarkar, *path-following method*[13] originated by Megiddo, and *affine scaling method*[5] proposed by Dikin.

Many variants of the affine scaling method have been further developed, proposed, and implemented on practical problems by numerous researchers ; primal affine scaling[2, 20], dual affine scaling[1], and primal-dual affine scaling methods[14], to name a few. Nazareth[16] has proposed a null-space version of primal affine scaling method(called *primal null-space affine scaling method*), and its properties and the results of the implementation on some of the

* Department of Mathematics, Kei-myung University. This paper was partially supported by university affiliated research institute, Korea Research Foundation, 1993.

non-trivial problems have been studied and reported in Kim and Nazareth[9].

Basically, these two types, the simplex method and interior point methods, comprise the solution techniques of linear programming. Some characteristics of the simplex method are vertex following, finite processing, and dependence of the combinatorial structure of a linear program. On the other hand, some of the characteristics of an interior point method are polynomial time complexity, infinite processing, and independence of the combinatorial structure of a linear program. Hence, each type has its distinctive characteristics, merits, and shortcomings.

Recently, a number of researchers have discussed and studied the possibility of combining the simplex method with interior-point techniques in order to produce a more effective linear programming method by utilizing and taking advantage of their special characteristics. For example, Nazareth[16] discussed the merits of augmenting one linear programming technique by another and proposed a way to hybridize the primal simplex method with an affine scaling method, but no extensive experiment or research has been performed in this direction. In [8], an attempt has been made in combining the simplex method with an interior point method in the setting of Dantzig-Wolfe decomposition. It illustrates the use of interior points of subproblems in decomposition procedures to alleviate some of the computational difficulties that plague decomposition algorithms. Kortanek and Shi[10] studied a hybrid algorithm that consists of an interior point algorithm followed by a purification algorithm to obtain a dual basic optimal solution. However, the effectiveness of correlating the two central types of technique of linear programming has not been studied fully and realized yet.

The purpose of this paper is to investigate the effective hybridization of the two techniques, specifically, the primal simplex method and the primal null-space affine scaling method using the idea of Nazareth[16]. From the structural point of view, an important advantage of the primal null-space affine scaling method is that it integrates well and naturally with the primal simplex method, as can be seen in later sections. These two methods will be integrated in the algorithmic convergent framework of Dembo and Sahi[4] to form a hybrid algorithm(HYBRID) for linear programming. The actual convergence proof of the algorithm will be studied and reported in the subsequent paper. Rather, we state some of the features possessed by this hybrid algorithm which will be formulated in section 3.2. The method

- can be initiated with any suitable feasible point
  (e.g., on a *facet* of the feasible polytope),
- avoids the combinatorial structure of the problems,
- utilizes as many variables corresponding to the negative reduced cost as possible in defining descent directions (relaxing steps),
- does not have to use all the variables in defining descent directions (restricted steps) as in

any interior point method, and

• contains the usual simplex method and a combination of simplex and interior point method (as determined by the value of κ defined in section 3.1).

This paper is organized as follows. In section 2, we define *relaxing* and *restricted* steps that can be fitted into the convergent framework of Dembo and Sahi. In section 3, we first describe an overview of the hybrid method and then formulate a specific and detailed algorithm(HYBRID). The results of the preliminary experiment on the viability of this hybrid method are reported and discussed in section 4. Finally, a few concluding remarks are given in section 5, with directions for further research in this area.

2. Relaxing and Restricted Steps for Hybrid Models

In this section we review the two methods that become an essential part of forming a hybrid model and redefine them as *relaxing* and *restricted* steps. Consider the following standard form of linear program(LP):

$$\begin{aligned} &\text{minimize } c^T x \\ &\text{s. t. } Ax = b \\ &\qquad x \geq 0, \end{aligned} \tag{1}$$

where c and x are n-dimensional vectors, and b is an m-dimensional vector, and A is an $m \times n$ matrix of full row rank with $m<n$.

2.1 Relaxing Steps

In the primal simplex method, the coefficient matrix A in (1) is partitioned into $A \equiv [B|N]$ where columns of B and N correspond to the basic and nonbasic variables, respectively. Without loss of generality, we assume that B consists of first m linearly independent columns of A. Let $x^\circ = [x^\circ_B x^\circ_N]^T$ be a nondegenerate basic feasible solution of (1), i.e., $x^\circ_B > 0$ and $x^\circ_N = 0$, and let $z^\circ = c^T x^\circ$

It can be shown that the simplex method can be interpreted as a method of coordinate descent in a reduced space, see [16], by making a transformation of variables

$$x = x^\circ + Z \Delta x_N \tag{2}$$

where Z is an $n \times (n-m)$ matrix of full rank for which $AZ=0$, i.e., Z spans the null space of A. Z is defined to be

$$Z \equiv \begin{bmatrix} -B^{-1}N \\ I_{\bar{n}\times\bar{n}} \end{bmatrix} \tag{3}$$

where $\bar{n}=n-m$. The resulting reduced space, called *linear local reduced model*, is given as

$$\text{minimize}(Z^Tc)^T\Delta x_N+z^\circ$$
$$x^\circ_N+x_N\geq 0 \tag{4}$$

By solving (4), the associated descent direction in the original space (1) is given by

$$\Delta x=x-x^\circ=Z\Delta x_N \tag{5}$$

In determining a descent direction at each iteration, the usual primal simplex algorithm chooses a single coordinate, for example, the one with the most negative reduced cost. However, the simplex method can be extended to determine a descent direction by changing as many coordinates simultaneously as possible. This can be done by incorporating one or more coordinates corresponding to the negative reduced cost(From now on, we shall understand *extended simplex method* to mean what we have just described). For peggied variable[17], i.e., for a nonbasic variable which is not at a zero level, we want to increase the value of the nonbasic variable $(x^\circ_N)_\kappa$ if $(Z^Tc)^T_\kappa < 0$ and to decrease the value of the nonbasic variable $(x^\circ_N)_\kappa$ from positive value if $(Z^Tc)^T_\kappa > 0$. Then, a combination of these κ's will give us a descent direction in the extended simplex method. We call this step a *relaxing step*.

2.2 Restricted Steps

We let $x^\circ > 0$ be an interior point of (1). In like manner, we partition $A\equiv[B|S]$, where columns of B correspond to the basic variables and columns of S correspond to the so-called *superbasic* variables. Of course, x° can be partitioned as $x^\circ=[x^\circ_B x^\circ_S]^T$, and note that both $x^\circ_B > 0$ and $x^\circ_S > 0$. Let $z^\circ=c^Tx^\circ$.

By making a similar transformation of variables

$$x=x^\circ+Z\Delta x_S, \tag{6}$$

where Z now is defined to be

$$Z \equiv \begin{bmatrix} -B^{-1}S \\ I_{\bar{n}\times\bar{n}} \end{bmatrix} \tag{7}$$

such that $AZ=0$, we obtain the following *quadratic local reduced model* around x°

$$\text{minimize}_{\Delta x_S\in R^{n-m}}\ (Z^Tc)^T\Delta x_S+\frac{1}{2}\Delta x_S^TZ^TD_o^{-2}Z\Delta x_S+z^\circ \tag{8}$$

where $D_o\equiv\text{diag}(x^\circ_1, x^\circ_2,\cdots,x^\circ_n)$. The readers are referred to [16] for more detailed steps.

The solution for the quadratic local reduced model (8) is explicitly given as

$$\Delta x_S = -([Z^T D_o^{-2} Z])^{-1} Z^T c \tag{9}$$

by direct differentiation, and the associated descent direction in the original space (1) is given by

$$\Delta x = x - x^\circ = Z \Delta x_S \tag{10}$$

We will call this a *restricted step*. By substituting (7) in (9) and rearranging, we obtain a more convenient computational form

$$[I + M_S]\Delta \bar{x}_S = -\bar{c}, \tag{11}$$

where $\Delta \bar{x}_S = D_S^{-1} \Delta x_S$, $\bar{c} = D_S Z_C^T$, $M_S = D_S S^T B^{-T} D_B^{-2} B^{-1} S D_S$, and $D_B \equiv \mathrm{diag}(x_1^o, x_2^o, \cdots, x_m^o)$ and $D_S \equiv \mathrm{diag}(x_{m+1}^o, x_{m+2}^o, \cdots, x_n^o)$.

Some properties of this algorithm, such as the property of descent direction when Δx in (10) is approximated (actually, $\Delta \bar{x}_S$ in (11) is approximated by inexact conjugate gradient method and this yields an approximation of Δx_S since $\Delta x_S = D_S \Delta \bar{x}_S$) and the experimental implementation on a number of non-trivial problems are investigated in [9] in conjunction with employing a *basis change strategy*. It changes basis B periodically in every p, *say* $p = 5$, iterations just as in the case of the simplex method which changes or updates its basis in every iteration. The change is based on the size of the current iterate and the sparsity of B.

2.3 Similarity in Structures

In the previous subsections, we mentioned about the partitioning of the coefficient matrix A. The partition is done not only for the simplex method($A = [B|N]$) but also for the primal null-space affine scaling method($A = [B|S]$). Moreover, the basis matrix B is changing periodically towards an optimal basis B^* for both cases; every iteration in the simplex method and every p iteration in the primal null-space algorithm. These observations clearly show that the structure of the two methods is very similar, and this is one of the reasons behind the motivation of hybridizing the two methods in a way to form an effective linear programming technique. Fortunately, the Dembo-Sahi convergent framework given below provides an excellent arena in which the two methods can be fitted in very naturally.

2.4 Dembo-Sahi Convergent Framework

We present the convergent framework of Dembo-Sahi[4] in order to describe the overall picture of how our hybridized algorithm is structured.

A1. Dembo-Sahi Framework

START with x° feasible

(Major iteration: index k)

IF optimal at x_k **THEN** exit.

(Minor iteration)

ELSE compute a *relaxing step*, p_k

and an acceptable point $x_k^+ = x_k + \alpha_k p_k$

set $y \leftarrow x_k^+$;

WHILE a constraint relaxation condition is not satisfied at y,

compute a *restricted step* p

and an acceptable point $y^+ = y + \alpha p$,

set $y = y^+$ and repeat.

ELSE $k \leftarrow k + 1$

$x_k = y$

Start a new major iteration

End

3. Hybrid Procedures

The idea of formulating the hybrid algorithm(HYBRID) is as follows. In section 2.1 we defined a *relaxing step* as a descent direction taken by the *extended* primal simplex method. We let this step be coincided with the *relaxing step* in the Dembo-Sahi convergent framework. We also defined a *restricted step* as a descent direction taken by the primal null-space affine scaling method(described in full detail in HYBRID). This time we let this step be coincided with the *restricted step* in the Dembo-Sahi framework. In this way, the two algorithms are naturally fitted into the convergent framework of Dembo and Sahi. The relaxing step is mainly being used to adjust many variables in the active set, i.e., to give a good descent direction, and the restricted steps are being used for suboptimizing within the current active set, i.e. , identifying a good basis.

3.1 Combining Techniques

Suppose a linear program is given in the standard form (1). Let x° be any initial feasible

solution. Here, x° need not be a vertex nor a strictly interior point. Let $0 < \delta < 1$ be a feasibility tolerance by which a variable x_j is classified as so-called *basic*, *superbasic* and *nonbasic* variables(terminology of Murtagh and Saunders[15]) and $0 \leq \kappa \leq 1$ be a fraction used in step 2 of the hybrid algorithm to determine the number of coordinates to be used in defining the descent direction as a relaxing step. It can be seen that by setting $\kappa = 1$ and starting at a vertex, we obtain the usual simplex method, and by setting $\kappa = 0$, all the negative reduced cost variables are involved in defining a descent search direction.

The algorithm is initiated with x° and a partition

$$A \equiv [B|S|N] \tag{12}$$

where B, S and N correspond to the columns of *basic*, *superbasic* and *nonbasic* variables, respectively. Here, x_j° is a nonbasic variable if $0 \leq x_j^\circ \leq \delta$ and it is a basic or a superbasic variable if $x_j^\circ > \delta$. We assume that there are at least m variables $x_j^\circ > \delta$ among which the basis matrix B consisting of m linearly independent columns can be selected. Let m, s and t be the number of columns in B, S and N, respectively. Note that then $n = m + s + t$.

We define

$$Z \equiv [Z_S Z_N] \equiv \begin{bmatrix} -B^{-1}S & -B^{-1}N \\ I_{s\times s} & 0 \\ 0 & I_{t\times t} \end{bmatrix} \quad \text{and} \quad \Delta x \equiv \begin{bmatrix} \Delta x_B \\ \Delta x_S \\ \Delta x_N \end{bmatrix} \tag{13}$$

and make the convention that $s=0$ means that the matrix S and Z_S do not exist, and we have an analogous convention when $t=0$. I stands for identity matrix. Clearly, $AZ_S=0$ and $AZ_N=0$ and hence Z_S and Z_N span the null space of A.

Now, a relaxing step is determined from the linear local reduced model (4) by considering only the *basic* and *nonbasic* variables from the foregoing partition (12). The computed step Δx_N from the local reduced model (4), in turn, gives a good descent direction in the original space. It is obtained by replacing Z in (2), (3), and (4) by Z_N in (13). A feasible step is taken in this direction and a new point is determined. Then, starting at this current point, several restricted steps Δx_S are determined from the quadratic local reduced model (8) by considering this time only the *basic* and *superbasic* variables in (12). In this case, Z in (6), (7), and (8) is replaced by Z_S in (13). These steps identify a good basis. The associated descent direction in the original space for each step is given by $\Delta x = Z_N \Delta x_N$ and $\Delta x = Z_S \Delta x_S$, respectively.

We note here that in both models the same basis matrix B is used in defining steps, and $\Delta x_S=0$ when the linear local reduced model is initiated and $\Delta x_N=0$ when the qradratic local reduced model is applied. These two steps then give one cycle of the combined method and also corresponds to one major iteration in the Dembo-Sahi framework. The process is repeated until a stopping criterion is satisfied.

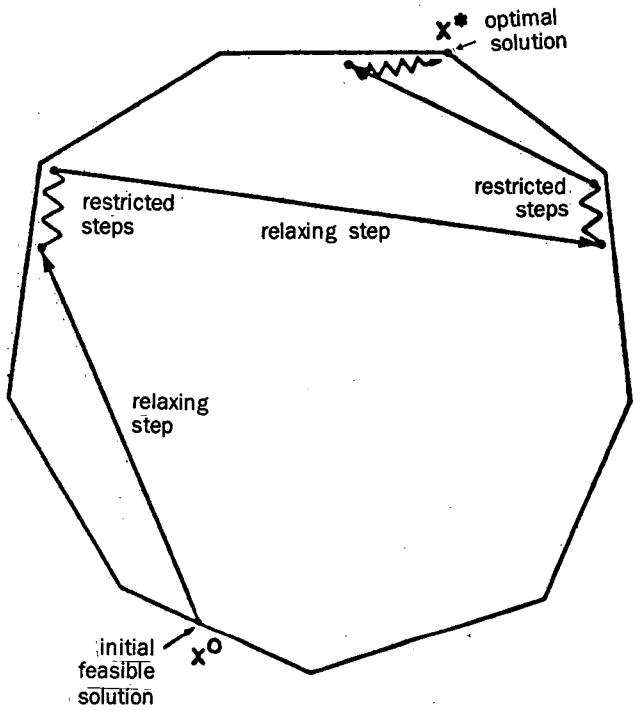
To see this more clearly, we illustrate the scheme in 2-dimensional space (see Figure 1). Let x° be a starting point and x^* be an optimal solution. Starting from x°, relaxing step and restricted steps taken in order are shown in the Figure 1. Repeated applicatioan of these two steps will eventually find an approximate optimal solution. It is apparent that each cycle of HYBRID algorithm gives a descent direction since each of the two methods, simplex and affine scaling, has descent properties.

3.2 The Hybrid Algorithm

Finally, we outline the algorithm HYBRID below by integrating the extended simplex method and the primal null-space affine scaling method in the Dembo-Sahi framework.

Algorithm HYBRID
Given *NSMIN*, *IMAX*, *JMAX*, ε, ε_{CGA}, and θ
START with x^1 feasible,
STEP 0 : k ← 1
(Major iteration ; index k)
(Optimality Criterion)
STEP 1 : IF$(Z_N^Tc)_j \geq -\varepsilon$, $j=1,\cdots,t$ and $|\ (Z_S^Tc)_j\ | \leq \varepsilon$, j=1,⋯,sTHEN exit.
ELSEIF$(Z_N^Tc)_j \geq -\varepsilon$, $j=1,\cdots,t$ GOTO STEP 4
ELSE compute a relaxing direction p_κ,
STEP 2 :
2A : $(\Delta x_N)_j \leftarrow \max\ [0,-(Z_N^Tc)_j]$, $j=1,\cdots,t$; $\sigma_{\min} \leftarrow \min_{j=1,\cdots,t}(Z_N^Tc)_j$

Figure 1. Relaxing and Restricted Steps in 2-D Space



IF $(\Delta x_N)_j < \kappa \mid \sigma_{\min} \mid$ THEN

$(\Delta x_N)_j \leftarrow 0,\ j=1,\cdots,t$

2B : $p_\kappa \leftarrow Z_N\ \Delta x_N$

Compute an acceptable point $y=x^\kappa+\alpha_\kappa p_\kappa$ by taking as large a step as possible along p_κ without violating a constraint.

2C : $\alpha_\kappa \leftarrow \max[\alpha>0 \mid x^\kappa+\alpha p_\kappa \geq 0]$

2D : $y \leftarrow x^\kappa+\alpha_\kappa p_\kappa$

STEP 3 : Update basis matrix and revise the partition of variables, namely s, t, S, and N (see comments below)

(Relaxation Condition)

If there are at least NSMIN superbasic variables, then start a minor iteration.

Else, start a major iteration.(typically NSMIN=2)

IF NS(# of superbasic variables) < NSMIN GOTO STEP 7

ELSE start minor iteration.

STEP 4 : (Minor iteration ; index i)

$i \leftarrow 1$

4A : DO WHILE $i \leq$ IMAX

$D_B \leftarrow \text{diag}(x_1,\cdots,x_m)$, $D_s \leftarrow \text{diag}(x_{m+1},\cdots x_n)$

(D_B and D_S are defined in section 2.2.)

4B : Compute a restricted direction p by the CG algorithm.

STEP CG0 : Let $\Delta x_S \leftarrow 0$

$r \leftarrow -D_S Z_S^T c$

$d \leftarrow r$

(*CG iteration; index j*)

$j \leftarrow 1$

STEP CG1 : Form $q \leftarrow [I+D_S S^T B^{-1} D_B^{-2} B^{-1} S D_S]d$ by the following steps :

$w \leftarrow SD_S d$

Solve $Bv=w$

$w \leftarrow D_B^{-2} v$

solve $B^T v=w$

$q \leftarrow D_S S^T v+d$

STEP CG2 : $\beta \leftarrow \frac{r^T r}{d^T q}$

$\Delta x_S \leftarrow \Delta x_S+\beta d$

$r_+ \leftarrow r-\beta q$

If($\|r_+\|<\varepsilon_{CGA}$ or $j \geq$ JMAX)exit

$\gamma \leftarrow \frac{r_+^T r_+}{r^T r}$

$d \leftarrow r_+ +\gamma d$

$r \leftarrow r_+$

$j=j+1$

Go to STEP CG1

4C : $p \leftarrow Z_S D_S \Delta x_S$

Find an acceptable point $y_+ \leftarrow y+\alpha_+ p$ by taking the largest possible step along p without violating feasibility.

STEP 5 : $\alpha_+ \leftarrow \max[\alpha>0 | y+\alpha p \geq 0]$

$y_+ \leftarrow y+\theta\alpha_+ p$($\theta$ *is a pullback constant,* $0<\theta<1$)

$i=i+1$

$y \leftarrow y_+$

GOTO STEP 4A

SETP 6 : Select a new basis using the *basis change strategy* and revise the partitions.

STEP 7 : Start a new major iteration :

$x^k \leftarrow y$

$k=k+1$

GOTO STEP 1

END

3.3 Comments

In step 1, the reduced costs are computed efficiently as follows: Solve $B^T\pi=c_B$ for π ; $Z_S^Tc=c_S-S^T\pi$; $Z_N^Tc=c_N-N^T\pi$(the vector c is also partitioned as $c^T=[c_B\ c_S\ c_N]^T$

In step 2A, by setting $\kappa=0$ we change as many variables simultaneously as possible (i.e., all the variables corresponding to the negative reduced cost are involved). By setting $\kappa=1$ and starting at a vertex, we obtain the usual simplex procedure with entering variable chosen to have the most negative reduced cost (resolved arbitrarily if there is a tie). When $0<\kappa<1$, then only a subset of the variables corresponding to the negative reduced cost are involved in defining a descent direction.

In step 2B, p_κ can be also efficiently computed as follows : Form $\Delta\tilde{x}_N=N\Delta x_N$; solve $Bv=\Delta\tilde{x}_N$; then

$$p_\kappa = \begin{bmatrix} -v \\ \Delta x_N \end{bmatrix}$$

In step 3, when a relaxing step is taken, usually only one basic variable will drop to zero if problems are nondegenerate and some nonbasic variables become superbasic variables. A *leaving variable* from the basis can be identified if the value of any basic variable is less than δ. An *entering variable* can be chosen from the superbasic variables whose value is greater than δ, and at the same time, is linearly independent with the rest of the columns of the basis. This can be accomplished by a standard dual simplex procedure of defining a price vector $\pi^T=e_l^TB^{-1}$ and finding a column of S, say s_j, for which π^Ts_j is nonzero (here e_l denotes a unit vector with nonzero element in position l, and index l corresponds to the basic variable that leaves the basis). Once the entering column has been found, the basis factorization can be updated. It could happen that more than one basic variable becomes tight at this step, in which case the updating procedure is repeated.

Step 4 gives the restricted steps by the primal null-space affine scaling algorithm, see [9]. The system of equation in (11) is approximated by the iterative method of conjugate gradient(CG).

In step 6, a new basis is determined based on the size of the current iterate and the sparsity of columns to be included in a new basis.

4. Experimental Results

Our objective in this section is to demonstrate the viability of the combined method developed in the previous section by applying on small dense problems. We make this experiment a basis for the development of more effective combined techniques (interior point methods and the simplex method) for linear programming.

4.1 Test Problems

We utilize the variants of Kuhn-Quandt problems. These problems are of the form

$$\begin{aligned} &\text{minimize } c^T x \\ &\text{s. t. } Ax \leq b \\ &\qquad x \geq 0, \end{aligned} \tag{14}$$

where A is a small dense $m \times n$ matrix with integer elements chosen at random in the range 1 to 100. The numerical range for integer elements of the vector b is from 5000 to 10000 and for the vector c is -1 to -9. This choice will guarantee that the problems are bounded and that an initial feasible solution is readily available. Also, many of the problems created in this procedure are nondegenerate. By adding slack variables, (14) can be expressed in the standard form of (1).

4.2 Results and Discussions

Table 1 displays the detail of the problems tested and the results obtained. The size of selected test problems is given in the first column, and column 2 shows the value of the parameter δ used for each problem. In each of the test problem, the matrix A includes an appropriate identity matrix, and κ is set to zero in all cases. We used the origin as an initial starting point and set the maximum number of minor steps $IMAX$ and the maximum number of CG steps $JMAX$ to be 5 in each test problem. We assigned $NSMIN = 1$ for the constraint relaxation condition, i.e., the minor iteration in HYBRID is started if there is at least one superbasic variable. The stopping criterion for the main iteration(outer loop) is

specifically given in step 1 of HYBRID ; here, the optimality tolerance $\varepsilon=1.0\times10^{-6}$ is used. We set $\varepsilon_{CGA}=1.0\times10^{-4}$ and $\theta=0.9$ throughout the whole operations.

We have coded our hybrid algorithm in GAUSS(similar to MATLAB). Column 3 gives the number of cycles of the algorithm HYBRID. Each cycle of HYBRID consists of several CG iterations, and the last column presents an average number of CG iterations for each HYBRID cycle.

Table 1
HYBRID Test Statistics

Size of A	δ	HYBRID	Avg. CG
10×30	0.1	6	17
10×40	0.01	4	14
20×40	0.01	4	15
20×50	0.01	5	17
20×60	0.1	5	19
30×60	0.01	7	24
30×70	0.1	7	20
30×80	0.1	5	18
40×80	0.1	7	24

For these test problems, though small and limited, the number of cycles of HYBRID for each problem was low as we expected, and 7 was the largest number of cycles we observed among these results. It appears that the number of cycles is increasing very slowly as the size of the problems is increased. Also, as the solution approached an optimal point, the number of CG iterations, not shown here, was reduced considerably in several cases. The solutions(objective function values) we obtained were at least within 4 significant digits accuracy in comparison with the results obtained by the pure simplex method(also implemented by GAUSS but not shown here).

To be a comparable method to any other existing method for linear programming, HYBRID should have the following properties: low number of cycles required even if the size of problems is increased and low number of CG iterations required in each HYBRID cycle. The hybrid algorithm described here seems to have the above properties, and it appears to be of worth further study on practical problems.

One disadvantage of the combined method is that if a problem is degenerate, in fact, many practical problems are degenerate, choosing an appropriate δ is rather a difficult task. One

possible remedy to overcome this difficulty would be to perturb data slightly, but again much research has to be done in this direction. In this experiment, we used $\kappa=0$ for all the problems tested, but when different values of κ were used, we obtained the similar behavior.

5. Conclusions and Further Research

In this paper, we formulated and studied a hybrid algorithm based on the simplex method and an interior point method. The hybridized method has some distinctive features and the results obtained from the preliminary computational experiments are encouraging. We point out especially that the successful formulation and implementation of this hybridized method is due to the particular characteristics of both primal simplex method and primal null-space affine scaling method. Even though the hybridized method is tested only on small dense nondegenerate problems and further development and testing are definitely needed, it provided a foundation work on the development and effective integration of interior-point techniques with the simplex method. Some directions for further research in this area includes finding a feasible point quickly and cheaply(not necessarily a vertex nor an interior point), obtaining a better constraint relaxation condition in HYBRID, level-2 implementation(see [18] for terminology) on practical problems, and the theoretical study of convergence of the algorithm HYBRID.

References

[1] Adler, I., N. Karmarkar, M. G. C. Resende and G. Veiga, "*An Implementation of Karmarkar's Algorithm for Linear Programming*", Mathematical Programming 44 (1986), 297-336.

[2] Barnes, E. R., "*A Variation on Karmarkar's Algorithm for Solving Linear Programming Problems*" , Mathematical Programming 36 (1986), 174-182.

[3] Dantzig, G. B., "*Linear Programming and Extensions*", Princeton University Press, Princeton, N. J (1963).

[4] Dembo, R. and S. Sahi, "*A Convergent Framework for Constrained Nonlinear Optimization*", School of Organization and Management Working Paper, Series B #69 Yale University (1984).

[5] Dikin, I. I., "*Iterative Solution of Problems of Linear and Quadratic Programming*", Soviet Mathematics Doklady 8 (1967), 674-675.

[6] Gill, P. E., M. A. Saunders and M. H. Wright, "*On Projected Newton Barrier Methods for Linear Programming and an Equivalence to Karmarkar's Projective Method*", Mathematical Programming 36 (1986), 183-209.

[7] Karmarkar, N. K., "*A New Polynomial-Time Algorithm for Linear Programming*", Algorithmica 1 (1984), 455-482.

[8] Kim, K. and J. L. Nazareth, "*The Decomposition Principle and Algorithms for Linear Programming*", Linear Algebra and Its Applications 152 (1991), 119-133.

[9] Kim, K. and J. L. Nazareth, "*A Primal Null-Space Affine Scaling Method*", to appear in : acm Transactions on Mathematical Software.

[10] Kortanek, K. O. and M. Shi, "*Convergence Results and Numerical Experiments on a Linear Programming Hybrid Algorithm*", European Journal of Operational Research 32 (1987), 47-61.

[11] Lustig, I. J., "*Feasibility Issues in a Primal-Dual Interior Point Method for Linear Programming*", Mathematical Programming 49 (1991), 145-162.

[12] McShane, K. A., C. L. Monma and D. F. Shanno, "*An Implementation of a Primal-Dual Interior Point Method for Linear Programming*", ORSA Journal on Computing 1 (1989), 70-83.

[13] Megiddo, N., "*Pathways to the Optimal Set in Linear Programming*", in : N. Megiddo, ed., Progress in Mathematical Programming, Springer, New York (1988), pp. 131-158.

[14] Monteiro, R. D. C. and I. Adler, "*Interior Path Following Primal-Dual Algorithms. Part I : Linear Programming*", Mathematical Programming 44 (1989), 27-41.

[15] Murtagh, B. A. and M. A. Saunders, "*MINOS User's Guide*", Technical Report SOL 83-20, Stanford Optimization Laboratory, Stanford, CA (1983).

[16] Nazareth, J. L., "*Pricing Criteria in Linear Programming*", in : N. Megiddo, ed., "Progress in Mathematical Programming", Springer, New York (1988), 105-129.

[17] Nazareth, J. L., "*Computer Solution of Linear Programs*", Oxford University Press, Oxford and New York (1987).

[18] Nazareth, J. L., "*Hierarchical Implementation of Optimization Methods*", In Numerical Optimization, 1984, P. Boggs, R. Byrd and R. Schnabel, Eds., SIAM Philadelphia, PA (1985), 199-210.

[19] Todd, M. J., "*A Low Complexity Interior- Point Algorithm for Linear Programming*", SIAM Journal on Optimization 2 (1992), 198-219.

[20] Vanderbei, R. J., M. S. Meketon and B. A. Freedman, "*A Modification of Karmarkar's Algorithm*", Algorithmica 1 (1986), 395-409.