85-B-1

A HYBRID METHOD FOR LINEAR PROGRAMMING

Kaoru TONÉ

September 1985

A HYBRID METHOD FOR LINEAR PROGRAMMING

Kaoru TONÉ

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

Abstract: We present a polynomial time algorithm for solving linear programming problems based on a combination of Karmarkar's new LP algorithm and Dantzig's simplex method. Instead of the orthogonal projection of Karmarkar's method, we introduce a projection on a basis system in the projected affine manifold in order to determine the search direction. Then a line search on the direction gives the next point in the iterations. The optimal solution is usually obtained as a basic solution and the dual solution is available at the same time. The proposed method is essentially a reduced gradient method on the projected manifold.
Key words: Linear Programming, Karmarkar's Method, Simplex Method, Reduced Gradient Method, Polynomial Time Algorithm

Introduction

In this paper, we will show an algorithm for solving the linear programming problem with a variable n-vector x:

$$\text{(LP)} \quad \min_{x} c^T x, \text{ subject to } Ax=b,\ x \geqq 0,$$

where A, b and c are a constant (m,n) matrix, a constant m-vector, and a constant n-vector, respectively. In Part 1 of this paper, we define a canonical form LP in the same way as Karmarkar [2], which is expressed as follows:

$$\text{(P)} \quad \min_{x} c^T x, \text{ subject to } Ax=0,\ e^T x=1,\ x \geqq 0,$$

where A is a constant (m,n) matrix and e is an n-vector composed of all 1's. With three assumptions, we show a method for solving the canonical LP. One of the main objects of the paper is the reduction of the computational cost of the orthogonal projection in Karmarkar's method. The validity and computational complexity of the method will be discussed.

In Part 2, we modify the main algorithm in Part 1 so as to solve the general linear programming problem (LP). Firstly we drop the assumption of the known optimum value in Part 1. Then we show a method for solving the unknown starting solution case. Finally, we exhibit preliminary computational results by our algorithm.

Part 1

Canonical Form LP and its Solution

1.1 Problem

The problem is defined as follows:

(1.1.1) (P) $\min_{x} c^T x$, subject to $Ax=0$, $e^T x=1$, $x \geqq 0$,

where A is an (m,n) matrix with $m \leqq n$.

We assume the following three coditions:

(Condition 1) (P) is feasible and $a^0=e/n$ is a feasible solution.

(Condition 2) the rank of A is m, and

(Condition 3) the optimum value of (P), denoted by z^*, is zero.

Let a positive solution to (P) be $x^0=(x_1^0,\ldots,x_n^0)^T$ and let $D=\text{diag}(x_1^0,\ldots,x_n^0)$.

Following Karmarkar, we define the projected problem (P') using x^0 and D.

(1.1.2) (P') $\min_{x'} (Dc)^T x'$, subject to $ADx'=0$, $e^T x'=1$,$x' \geqq 0$.

Throughout the paper, we use the symbols and notations listed below.

$\Omega := \{x \in R^n \mid Ax=0\}$ (an affine space)

$S := \{x \in R^n \mid x \geqq 0,\ e^T x=1\}$ (the simplex)

$F := \{x \in R^n \mid Ax=0,\ e^T x=1,\ x \geqq 0\}$ (the feasible region of Problem (P))

$T :\ x'=D^{-1}x/e^T D^{-1}x$ (a projective transformation)

$T^{-1}:\ x=Dx'/e^T Dx'$ (the inverse transformation)

$\Omega' := \{x' \in R^n \mid ADx'=0\}$ (the tranformed affine space)

$F' := \{x' \in R^n \mid ADx'=0, e^T x'=1,\ x' \geqq 0\}$ (the feasible region of the projected problem)

$a^0 := e/n$ (the center of the simplex S)

$f(x) := \sum_{i=1}^{n} \log(c^T x/x_i)$ (the potential function)

$f'(x') := \sum_{i=1}^{n} \log((Dc)^T x'/x_i')$ (the projected potential function)

$$\bar{A} := \begin{bmatrix} A \\ e^T \end{bmatrix}$$

$$\bar{A}' := \begin{bmatrix} AD \\ e^T \end{bmatrix}$$

$P := I-\bar{A}'^T(\bar{A}'\bar{A}'^T)^{-1}\bar{A}'$ (the projection matrix)

$c' := Dc$

$c_p := Pc'$ (the orthogonal projection of c' onto F')

$z_0 := c'^T a^0$ (the projected objective function value at a^0)

1.2 The Basic Form

Let a basis of the matrix A be B and the remaining part of A be R. Without losing generality, we assume that B is the first m columns of A. Thus, (1.2.1) $A=[B \mid R]$.

Let

(1.2.2) $Y=B^{-1}R=[y_1,\ldots,y_{n-m}]$.

Let a basis of $\bar{A}$ including B be $\bar{B}$. $\bar{B}$ is a regular (m+1,m+1) submatrix of $\bar{A}$ and we assume without losing generality, $\bar{B}$ is the first (m+1) columns of $\bar{A}$. (Note that $\bar{A}$ is full rank by the Conditions 1 and 2.) Let

(1.2.3) $\bar{A}=[\bar{B} \mid \bar{R}]$ and

(1.2.4) $\bar{Y}=\bar{B}^{-1}\bar{R}=[\bar{y}_2,\ldots,\bar{y}_{n-m}]$.

$\bar{Y}$ is an (m+1,n-m-1) matrix.

Throughout this paper, we use several expressions based on the basis matrices B and $\bar{B}$. The notations x_B, x_R, c_B and c_R mean the partitions of x and c corresponding to B and R, respectively. Thus, the problem (P) can be expressed in the following equivalent forms listed in Figures 1(a), (b), (c) and (d). We think that the reader can easily understand the meaning of the formats and symbols. In Figure 1(d), p_j is the (m+1)st element of the vector $\bar{y}_j$ (j=2,...,n-m). Noting that the diagonal matrix D is non-singular, BD_B is a basis of AD where D_B is the submatrix of D corresponding to B. In Figure 2, we show the basic forms of the problem (P') corresponding to Figure 1. We denote the bases for A' (=AD) and $\bar{A}'$ by B' and $\bar{B}'$, respectively. Let $A'=[B' \mid R']$ and $\bar{A}'=[\bar{B}' \mid \bar{R}']$.

Then we have the relations:

(1.2.5) $\zeta_j = (B')^{-1}R_j'$ (j=1,...,n-m)

(1.2.6) $g_j = 1-e^T\zeta_j$ (j=1,...,n-m)

(1.2.7) $d_j = c_{R_j}'-c_B'\zeta_j$ (j=1,...,n-m)

(1.2.8) $H = (\bar{B}')^{-1}\bar{R}' =(\eta_2,\ldots,\eta_{n-m})$

(1.2.9) $\qquad h_j = c_{\bar{R}_j}{}' - c_{\bar{B}}{}'^{T} \eta_j \qquad (j=2,\ldots,n-m)$

(1.2.10) $\qquad Z = D_B^{-1} B^{-1} R D_R.$

The notations $x_{\bar{B}}{}'$, $x_{\bar{R}}{}'$, $c_{\bar{B}}{}'$ and $c_{\bar{R}}{}'$ are the partitions of x' and c' based on the basis $\bar{B}'$.

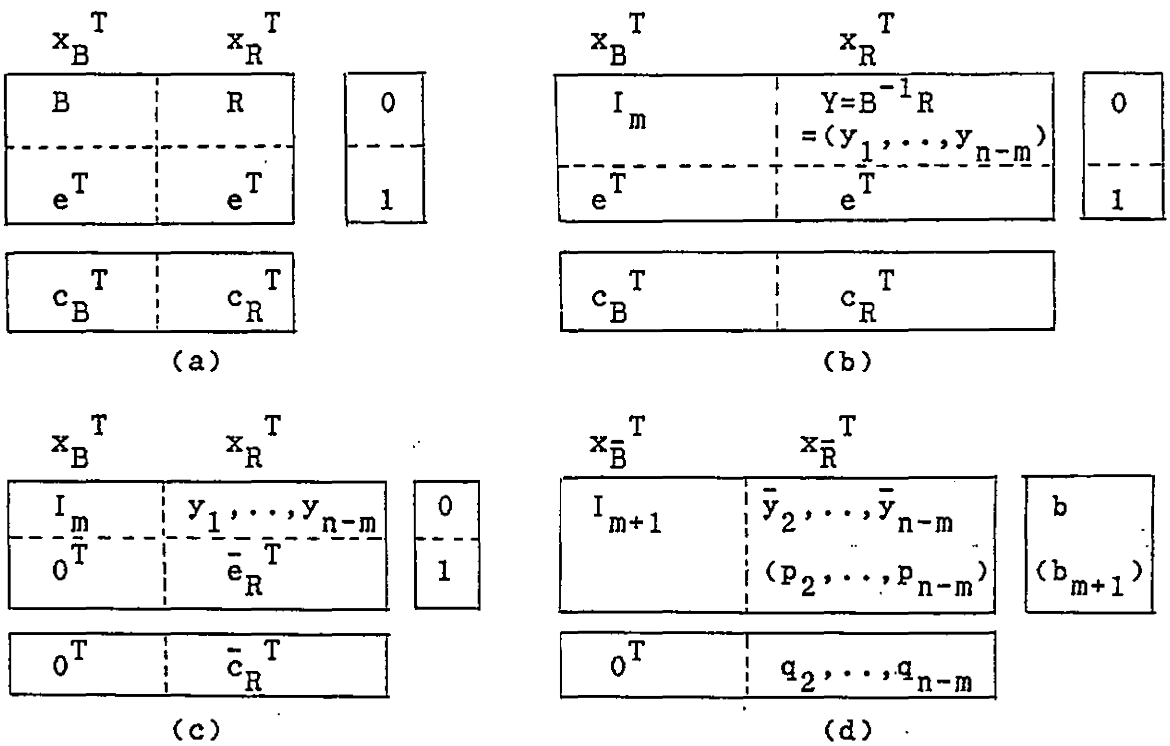
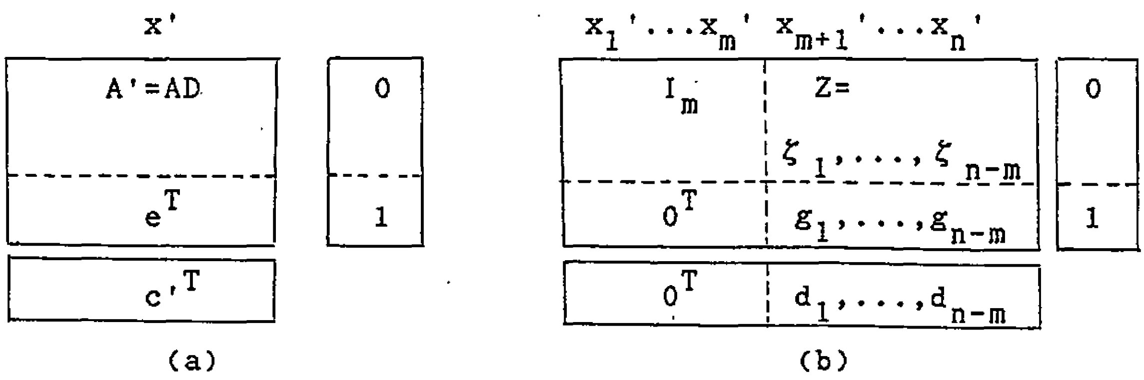


Figure 1. Tableau for (P)



$x_1', \ldots, x_{m+1}'$	$x_{m+2}', \ldots, x_n'$	
I_{m+1}	$H = \eta_2, \ldots, \eta_{n-m}$	β
0	$h_2, \ldots, h_{n-m}$	

(c)

Figure 2. Tableau for (P')

1.3 Method for Solving the Canonical Form Problem

Algorithm A

Step 0. Initialize x by e/n and choose a basis $\bar{B}$ of $\bar{A}$.

Step 1. Test optimality of x and $\bar{B}$.

Step 2. Determine the search direction s.

Step 3. Find $\min_{t>0} f'(a^0+ts)$. (line search)

Step 4. Update x and $\bar{B}$: go back to Step 1.

[Details of the Algorithm A]

Step 0: Any non-singular (m+1,m+1) submatrix $\bar{B}$ of $\bar{A}=\begin{bmatrix} A \\ e^T \end{bmatrix}$ can be used as the starting basis.

Step 1: The optimality tests are as follows:

(a) the basis $\bar{B}$ should be feasible, i.e.,

(1.3.1) $$b=\bar{B}^{-1}e_{m+1} \geqq 0,$$

where e_{m+1} is the (m+1)st unit vector,.

(b) the objective function value of (P) should be zero, i.e.,

(1.3.2) $$c_{\bar{B}}^T b=0.$$

Tests (a) and (b) are for $\bar{B}$. For x, we have the test:

(1.3.3) (c) $|c^T x| < \varepsilon$,

where ε is a convergence tolerance number.

Step 2: Step 2 and Step 3 are related with the projected problem (P'). For the search direction $s=(s_{\bar{B}},s_{\bar{R}})^T$, we take

(1.3.4) $$s_{\bar{R}}=-h=-(h_2,\ldots,h_{n-m})^T$$

and

(1.3.5) $$s_{\bar{B}}= Hh.$$

If the direction thus defined does not satisfy the criterion that will be mentioned in Section 1.5, we will modify the direction.

Step 3: The line search will be carried out with respect to t (>0) on the line segment $a^0+ts\in F'$ to minimize the transformed potential function $f'(a^0+ts)$. Let the optimum value of t be t^*. Then, we define

(1.3.6) $$x'=a^0+t^*s.$$

Step 4: Using the inverse transformation T^{-1}, we have the next iterate x by

(1.3.7) $$x=Dx'/e^T Dx'.$$

For the choice of the new basis $\bar{B}$, we apply the following rule:

[Basis Change Rule]

Use the linear independent (m+1) columns of $\bar{A}$ corresponding to the decreasing order of elements of x obtained above.

1.4 Several Propositions related with the Algorithm

Let us define the vector $\xi_j\in R^n$ (j=2,...,n-m) by

(1.4.1) $$\xi_j = \begin{bmatrix} -\eta_j \\ e_j \end{bmatrix}$$

where

(1.4.2) $$\eta_j = (\bar{B}')^{-1} \bar{R}_j'.$$

<u>Proposition 1.</u> $\{\xi_j\}$ $(j=2,...,n-m)$ spans the affine manifold $\{x' \in R^n \mid ADx'=0,\ e^T x'=1\}$.

<u>Proof</u> Obvious. □

<u>Proposition 2.</u>

(1.4.3) $$\xi_j^T c' = h_j \quad (j=2,...,n-m).$$

<u>Proof</u> Since $c'=(c_{\bar{B}}', c_{\bar{R}}')^T$, we have

$$\xi_j^T c' = -\eta_j^T c_{\bar{B}}' + c_{\bar{R}_j}' = h_j, \text{ by (1.2.9).}$$ □

<u>Proposition 3.</u>

(1.4.4) $$\xi_j^T c_p = h_j \quad (j=2,...,n-m).$$

<u>Proof</u> By the definition of c_p, we have $c_p=Pc'$. Thus,

$$\xi_j^T c_p = \xi_j^T Pc' = (P\xi_j)^T c'.$$

Since $\xi_j \in F'$, $P\xi_j = \xi_j$. So, we have $\xi_j^T c_p = \xi_j^T c' = h_j$ by Proposition 2. □

<u>Proposition 4.</u>

(1.4.5) $$\| c_p \| \leq \| h \|,$$

where $h=(h_2,...,h_{n-m})^T$ and $\| \cdot \|$ means the Euclidean norm of a vector.

<u>Proof</u> Since $\{\xi_j\}$ spans F', there exist $\lambda_2,...,\lambda_{n-m}$ such that

$$c_p = \sum_{j=2}^{n} \lambda_j \xi_j .$$

Then, we have

$$\| c_p \|^2 = (\Sigma \lambda_j \xi_j)^T c_p = \Sigma \lambda_j \xi_j^T c_p = \Sigma \lambda_j h_j$$

$\leqq \Sigma |\lambda_j| |h_j| \leqq ((\Sigma \lambda_j^2)(\Sigma h_j^2))^{1/2} \leqq \|c_p\| \|h\|$.
Thus, we have $\|c_p\| \leqq \|h\|$. □

The following proposition is essentially proved by Kojima [4] and Todd and Burrell [6].

<u>Proposition 5.</u> Assume that the minimum value of (P) is non-positive. If $\|c_p\|>0$, then the projected objective function value z_0 at a^0 is not greater than $\|c_p\|$, i.e.,

(1.4.6) $\|c_p\| \geqq z_0 \ (=c'a^0)$.

<u>Proof</u> The dual to (P') is

(D') max v_{m+1}, subject to $v^TA'+v_{m+1}e^T \leqq c'^T$,

where $A'=AD$, $c'=Dc$, $v\in R^m$ and $v_{m+1}\in R$. For any v^T, $(v^T, \bar{v}_{m+1})$ with $\bar{v}_{m+1}=\min_j(c'^T-v^TA')_j$, is feasible to (D'). From the assumption in the proposition, we have $\min\{c^Tx \mid x\in F\}\leqq 0$. Since $c^Tx=c'^Tx'/e^TDx'$, the minimum objective function value to (P') is nonpositive, i.e.,

$$\min \{c'^Tx' \mid x'\in F'\}\leqq 0.$$

Thus $\bar{v}_{m+1}\leqq 0$.

Let P_e be the projection matrix $I-ee^T/n$ onto the null space of e and $P_{A'}$ be the projection matrix $I-A'^T(A'A'^T)^{-1}A'$ onto the null space of A'. We have

$$c_p=Pc'=P_eP_{A'}c'=P(c'-A'^T(A'A'^T)^{-1}A'c')$$
$$=c'-A'^T(A'A'^T)^{-1}A'c'-ee^T(c'-A'^T(A'A'^T)^{-1}A'c')/n.$$

Since $A'e=0$, we have

$$c_p=c'-A'^T(A'A'^T)^{-1}A'c'-(c'^Te/n)e.$$

Also, from the duality relation,

$c'^{T}e/n \geqq \bar{v}_{m+1}$.

Now, let v be $(A'A'^{T})^{-1}A'c'$. Then, for some i, we have

$\bar{v}_{m+1} = (c' - A'^{T}(A'A'^{T})^{-1}A'c')_{i}$.

Hence

$(c_p)_i = \bar{v}_{m+1} - c'^{T}e/n \leqq 0$.

Therefore

$\| c_p \| \geqq |(c_p)_i| = c'^{T}e/n - \bar{v}_{m+1} \geqq c'^{T}e/n = z_0$,

which completes the proof. □

<u>Proposition 6.</u> Under the same assumption with Proposition 5, if $\| c_p \| > 0$, then we have,

(1.4.7) $\| h \| \geqq c'a^{0} = z_0$.

<u>Proof</u> From Propositions 4 and 5, the conclusion follows. □

1.5 Validity of Algorithm A and some Additional Features of the Algorithm

The search direction $s=(s_{\bar{B}}, s_{\bar{R}})^{T}$ defined in Step 2, can be expressed as follows:

(1.5.1) $$s = \begin{bmatrix} s_{\bar{B}} \\ s_{\bar{R}} \end{bmatrix} = -\begin{bmatrix} -H \\ I \end{bmatrix} h = -\sum_{j=2}^{n-m} h_j \xi_j .$$

By moving a unit length on this direction, the projected objective function $c'^{T}x'$ can be reduced by the amount $-\Delta z'$,

(1.5.2) $$\Delta z' = c'^{T}(-\Sigma h_j \xi_j / \| s \|) = -(\Sigma h_j^{2}) / \| s \| = -\| h \|^{2} / \| s \| .$$

Now let us define

(1.5.3) $$\rho = \| h \| / \| s \| .$$

Then we have

(1.5.4) $\Delta z' = -\rho \| h \|$.

From Propositions 4 and 5, the following relations hold.

(1.5.5) $\Delta z' = -\rho \| h \| \leqq -\rho \| c_p \| \leqq -\rho z_0$.

Thus we have the lemma.

Lemma 1.1 In the algorithm A, if we move from a^0 to $a^0 + s/\| s \|$, the projected objective function z' of (P') reduces at least by $-\rho z_0$. Therefore we have

(1.5.6) $c'^T(a^0 + (\alpha/n)s/\| s \|) \leqq (1-\rho\alpha/n)z_0$,

where $z_0 = c'^T a^0$, $\rho = \| h \| / \| s \|$ and $0 < \alpha < 1$.

Karmarkar [2] proved the following two lemmas.

Lemma 1.2 If $|\varepsilon| \leqq \alpha < 1$, then

(1.5.7) $\varepsilon - \varepsilon^2/2(1-\alpha)^2 \leqq \log(1+\varepsilon) \leqq \varepsilon$.

Lemma 1.3 If $\| x - a^0 \| \leqq \alpha/n < 1$ and $e^T x = 1$, then

(1.5.8) $0 \leqq -\sum_{j=1}^{n} \log x_j - n \log n \leqq \alpha^2/2(1-\alpha)^2$.

Theorem 1.1 With notations as in Lemma 1.1, we have

(1.5.9) $f'(a^0 + (\alpha/n)s/\| s \|) \leqq f'(a^0) - \delta$,

where

(1.5.10) $\delta = \rho\alpha - \alpha^2/2(1-\alpha)^2$.

Proof $f'(a^0 + (\alpha/n)s/\| s \|) = n \log c'^T(a^0 + (\alpha/n)s/\| s \|)$

$- \sum_{j=1}^{n} \log (a^0 + (\alpha/n)s/\| s \|)_j$

$\leqq n \log (1-\rho\alpha/n)z_0 - \Sigma \log (a^0 + (\alpha/n)s/\| s \|)_j$ (by Lemma 1.1)

$\leqq n \log z_0 - \rho\alpha - \Sigma \log (a^0 + (\alpha/n)s/\|s\|)_j$ (by Lemma 1.2)

$= \Sigma \log c'^T a^0/a_j^0 - \rho\alpha - \Sigma \log(a^0 + (\alpha/n)s/\|s\|)_j + \Sigma \log a_j^0$

$= f'(a^0) - \rho\alpha - \{\Sigma \log(a^0 + (\alpha/n)s/\|s\|)_j + n \log n\}$

$\leqq f'(a^0) - \rho\alpha + \alpha^2/2(1-\alpha)^2.$ (by Lemma 1.3) □

Several sampled values for ρ, α and δ are as follows:

ρ	α	δ
3/4	1/4	19/144=.131944
2/3	1/4	1/9 =.111111
1/2	1/4	5/72 =.069444
1/3	1/4	1/36 =.027777
1/4	1/4	1/44 =.022727
1/4	1/5	3/160 =.01875
1/5	1/5	7/800 =.00875
1/10	1/10	.003827

Table 1.

For any value of ρ, the maximum of δ is attained at α^* such that

(1.5.11) $\alpha^*/(1-\alpha^*)^3 = \rho$.

The maximum value of δ is given by

(1.5.12) $\delta^* = \alpha^{*2}(1+\alpha^*)/2(1-\alpha^*)^3 = \rho(1+\alpha^*)/2.$

In Table 2, we show several values of interest for ρ, α^* and δ^*. They will be used in the algorithm in Part 2 so that a certain reduction of the potential function should be expected. Although $\delta^*>0$ for any value of ρ, a very small ρ results in a small reduction of f', as is seen from (1.5.12). This reflects the possibility that the angle between the true steepest descent direction ($-c_p$)

and the search direction s becomes nearly orthogonal since

(1.5.13) $\cos\theta = -c_p^T s/\|c_p\|\|s\| = -c_p^T(-\Sigma\xi_j h_j)/\|c_p\|\|s\|$
$= \Sigma h_j^2/\|c_p\|\|s\| = \|h\|^2/\|c_p\|\|s\| = \rho\|h\|/\|c_p\|.$

ρ	α^*	δ^*
1.0	.317672	.658836
.2	.13117	.113117
.1	.0783006	.053915
.05	.043724	.0260931
.01	.00970874	.00504854
.005	.00492611	.00251232

Table 2.

So, if ρ goes down below some value ρ_0, say .1, we switch to another search directions. They include:

[Search Direction Policy]

(a) <u>Orthonormalization</u>: From the coordinate system $\{\xi_j\}$ (j=2,..., n-m), we build up an orthonormalized system $\{\bar{\xi}_j\}$ (j=2,...,n-m). Let

(1.5.14) $\bar{\lambda}_j = \bar{\xi}_j^T c'$ (j=2,...,n-m).

Then c_p is given by

(1.5.15) $c_p = \sum_{j=2}^{n-m} \bar{\lambda}_j \bar{\xi}_j.$

In this case, we have the steepest descent direction itself. It is needless to say that the resulting algorithm is just the same as Karmarkar's. The main cost of computation is that of orthonormalization. It costs about $(n-m)^2 m/2$ arithmetic operations. However, we can stop the orthonormalization as soon as, for some k, the following relation becomes to be satisfied.

$$(1.5.16)\quad (\sum_{j=2}^{k} \bar{\lambda}_j^{\,2})^{1/2} / \| c' \| \geqq \rho_0 .$$

Then, the search direction s given by

$$(1.5.17)\quad s = -\sum_{j=2}^{k} \bar{\lambda}_j \bar{\xi}_j$$

ensures the reduction of the projected objective function value by $\rho_0 z_0$, since

$$\Delta z' = c'^T s / \| s \| = -\sum_{j=2}^{k} \bar{\lambda}_j c'^T \bar{\xi}_j / \| s \|$$

$$= -(\sum_{j=2}^{k} \bar{\lambda}_j^{\,2}) / (\sum_{j=2}^{k} \bar{\lambda}_j^{\,2})^{1/2} = -(\sum_{j=2}^{k} \bar{\lambda}_j^{\,2})^{1/2}$$

$$\leqq -\rho_0 \| c' \| \leqq -\rho_0 \| c_p \| \leqq -\rho_0 z_0 .$$

Therefore, we can get at least $\rho_0 z_0$ reduction in the projected objective function value.

Note that even if the inequality (1.5.16) is not satisfied for k= n-m, we have, by the definition of c_p and Proposition 5,

$$\Delta z' = -\| c_p \| \leqq -z_0 \leqq -\rho_0 z_0 .$$

Hence, we can get at least $\rho_0 z_0$ reduction in the projected objective function value.

(b) <u>Normal Equations</u>: Since the system $\{\xi_j\}$ (j=2,...,n-m) spans F', we have, for some $\{\lambda_j\}$,

$$(1.5.18)\quad c_p = \sum_{j=2}^{n-m} \lambda_j \xi_j .$$

From this relation and $\xi_j^T c_p = \xi_j^T c' = h_j$, we have a system of equations

(1.5.19) $\sum_{j=2}^{n-m} (\xi_i{}^T \xi_j) \lambda_j = h_i$. (i=2,...,n-m)

This is the normal equations to determine $\{\lambda_j\}$. By solving the system for $\{\lambda_j\}$, we can get the steepest descent direction $-c_p$. The cost of this computation is about $(n-m)^2(n+2m)/6$ arithmetic operations.

Applications of the search direction policy mentioned above pay back to get a good reduction in the potential function value.

Corollary 1.1 With notations as in Lemma 1.1, if we use the search direction s augmented by the search direction policy (a) or (b), we have

(1.5.20) $f'(a^0+(\alpha/n)s/\|s\|) \leqq f'(a^0)-\delta_0$,

where

(1.5.21) $\delta_0=\rho_0\alpha-\alpha^2/2(1-\alpha)^2$.

Lemma 1.4 Let x^0 and x^1 be the images of a^0 and $a^0+(\alpha/n)s/\|s\|$ by the transformation T^{-1}, then we have

(1.5.22) $f(x^1) \leqq f(x^0)-\delta$

where

(1.5.23) $\delta=\rho\alpha-\alpha^2/2(1-\alpha)^2$.

Proof Straightly from the definitions of T^{-1} and f. □

Theorem 1.2 If we start from a feasible solution $x^0=e/n$ to (P) and apply the algorithm A augmented by the search direction policy (a) or (b) to get x^k after k iterations, then we have

(1.5.24) $c^Tx^K \leqq \exp(-k\delta_0/n)c^Tx^0$, where

(1.5.25) $\delta_0=\rho_0\alpha-\alpha^2/2(1-\alpha)^2$.

Proof Since $f(x^1)\leqq f(x^0)-\delta_0$, we have

$$\Sigma \log c^Tx^1/x_j^1 \leqq \Sigma \log c^Tx^0/x_j^0-\delta_0.$$

Hence $n \log c^Tx^1\leqq n \log c^Tx^0-\delta_0+\Sigma(\log n+\log x_j^0)$.

In the same way, using Lemma 1.4, we have

$$n \log c^Tx^{p+1}\leqq n \log c^Tx^p-\delta_0+\Sigma(-\log x_j^p+\log x_j^{p+1})$$

for $p=1,\ldots,k-1$.

Thus, we have

$$n \log c^Tx^k\leqq n \log c^Tx^0-k\delta_0+\Sigma(\log n+\log x_j^k)$$

$$\log c^Tx^k\leqq \log c^Tx^0-k\delta_0/n+\log n+\Sigma \log(x_j^k)^{1/n}$$

$$c^Tx^k\leqq \exp(-k\delta_0/n)c^Tx^0+\log \prod_{j=1}^{n}(nx_j^k)^{1/n}.$$

Now nx_j^k's are positive numbers with arithmetic mean 1. Thus their geometric mean is at most 1 which implies that the last term of the above inequality is nonpositive.

Hence we have

$$c^Tx^k\leqq \exp(-k\delta_0/n)c^Tx^0. \quad \square$$

In Step 3 of the algorithm A, we apply a line search to find out the minimum point of the potential function f' on the line segment. The function f',

$$f'(x')=\Sigma \log c'^Tx'/x'_j$$

has a favorable feature as noted below.

Lemma 1.5 (Martos [5], p.62)

If a scalar function p(x) is convex and nonnegative and a scalar function q(x) is concave and positive in the convex set $X \subset R^n$, then

$$r(x)=p(x)/q(x)$$

is strictly quasiconvex.

Lemma 1.6 (Martos [5], p.59)

If a scalar function r(x) is strictly quasiconvex in the convex set $X \subset R^n$, and a scalar function ϕ is strictly increasing, then $\phi(r(x))$ is strictly quasiconvex in X.

From Lemmas 1.5 and 1.6, we have

Theorem 1.3 The potential function f' is strictly quasiconvex in F'.

Corollary 1.2 Since f' is strictly quasiconvex in F', if it has a stationary point on the line segment $a^0+ts \in F'$, then it is a minimizer.

As to the choice of the basis $\bar{B}$, any basis of $\bar{A}$ can be, basically, used for our purpose from the theoretical point of view. However, the basis change rule proposed in Section 1.3 is most fitted from the computational point of view, as pointed out below:

(a) The optimal basis test has the biggest chance of success, if we choose the basis corresponding to up to the (m+1)st largest elements of x, keeping the linear independency.

(b) If the problem (P) has a nondegenerate optimal basic solution and x^0 is sufficiently close to the solution, the elements of the matrix H are small in the absolute value, since

$$H=(\bar{B}')^{-1}\bar{R}'=D_{\bar{B}}^{-1}\bar{B}^{-1}\bar{R}D_{\bar{R}}.$$

Then the coordinate system $\{\xi_j\}$ defined by (1.4.1) becomes nearly orthogonal. Hence the ratio

$$\rho = \| h \| / \| s \|$$

tends to unity, as $\| s \| \fallingdotseq \| h \|$. Thus, we can expect a bigger reduction in the potential function value than by using other basis system. Even in the degenerate case, the elements of H remain in a moderate scale, because the operation $D_{\bar{B}}^{-1}(.)D_{\bar{R}}$ does not magnify its elements so much as in other basis choice rules.

Kojima [4] proposed a method for determining basic variables of optimum solutions in Karmarkar's algorithm. We can incorporate the idea into our framework as follows:

<u>Theorem 1.4</u> Assume that the minimum value of the objective function of (P) is nonpositive. If $x_j^*=0$ in the optimal solution x^* to (P), then for every $\mu>0$, x^* is also an optimal solution to the problem

(P") min $c''^T x$, subject to $Ax=0$, $e^T x=1$, $x \geqq 0$,

where

(1.5.26) $c''=c+\mu e_j$.

Therefore, for any basis $\bar{B}$ and any feasible solution $x^0(>0)$, we have the inequality

(1.5.27) $\| h(\mu) \| \geqq z_0(\mu)$,

where $h(\mu)$ is the reduced costs vector for c" related with $\bar{B}$ in the projected problem (P'), and $z_0(\mu)=(Dc'')^T a^0$.

<u>Proof</u> From Proposition 6, the conclusion follows. □

<u>Theorem 1.5</u> (Basic Variable Test)

With assumption and notations as in Theorem 1.4, for any basis B to (P) and any feasible solution $x^0(>0)$ to (P) with $c^Tx^0>0$, let

(1.5.28) $h(\mu)=h+\mu\bar{h}$,

(1.5.29) $u_j=\|\bar{h}\|^2-(x_j^0)^2/n^2$,

(1.5.30) $v_j=h^T\bar{h}-z_0x_j^0/n$ and

(1.5.31) $w=\|h\|^2-z_0^2$,

where $z_0=(Dc)^Ta^0$.

If (a) $u_j<0$, (b) $u_j=0$, $v_j<0$ or (c) $u_j>0$, $v_j<0$ and $v_j^2-u_jw>0$, then x_j^* must be positive in any optimal solution to (P).

Proof $z_0(\mu)=(D(c+\mu e_j))^Ta^0=c^TDa^0+\mu e_j{}^TDa^0$

$=c^Tx^0/n+\mu x_j^0/n$.

Therefore, if $x_j^*=0$ in the optimal solution x^* to (P), we have, for every $\mu>0$,

(1.5.32) $\|h+\mu\bar{h}\|\geq c^Tx^0/n+\mu x_j^0/n>0$,

by Theorem 1.4 and by the assumption $c^Tx^0>0$.

Hence $\|h+\mu\bar{h}\|^2\geq(c^Tx^0/n+\mu x_j^0/n)^2$.

The above inequality is quadratic in μ. Rearranging the inequality with respect to μ and considering the conditions to hold the inequality for every $\mu>0$, the conclusion follows. □

1.6 Computational Complexity of the Algorithm

The main computational costs of our algorithm are those of Step 2 (finding the search direction s) and Step 4 (updating the basis B). Therefore, we will discuss the complexity of Step 2 and Step 4 and then the overall complexity of the algorithm.

Step 2. If we keep Y $(=B^{-1}R)$ at each iteration, we can get the matrix $Z=[\zeta_1,\ldots,\zeta_{n-m}]=D_B^{-1}B^{-1}RD_R$ by 2m(n-m) multiplications and divisions and $\{g_i\}$ and $\{d_i\}$ in the figure 2 by m(n-m) multiplications. The matrix H costs m(n-m+1) multiplications and divisions. $s_{\bar{B}}$ (=Hh) costs (m+1)(n-m-1) multiplications. Therefore, we need approximately 5m(n-m) multiplications and divisions. However, if we apply the search direction policy, it costs about $(n-m)^2m/2$ multiplications and divisions. Thus, in the worst case, the computational complexity of Step 2 is $O(n^3)$, if m is O(n).

Step 4. Instead of updating $\bar{B}$ and $\bar{B}^{-1}$ directly, we can update $Y=B^{-1}R$ at each iteration if a basis change occurs. It costs $O(n^2)$ arithmetic operations for each pair of basic and nonbasic variables.

Hence, the complexity of the algorithm per iteration is $O(n^3)$ in the worst case. Total number of iterations to finish the algorithm is O(n), since the rate of convergence is at least linear by Theorem 1.2. Therefore the overall complexity of the algorithm is $O(n^4)$ in the worst case. However, the line search in Step 3 reduces the potential function value significantly and we may chance to get the optimal basis long before the algorithm converges to the optimum point. The cost of the basic variable check proposed in Theorm 1.5 is O(n) per variable.