Journal of the Operations Research
Society of Japan
Vol. 33, No. 2, June 1990

A DUAL ALGORITHM FOR FINDING THE MINIMUM-NORM POINT IN A POLYTOPE

Satoru Fujishige Ping Zhan
University of Tsukuba

(Received August 30, 1989)

Abstract We give a dual algorithm for the problem of finding the minimum-norm point in the convex hull of a given finite set of points in a Euclidean space. Our algorithm repeatedly rotates a separating supporting-hyperplane and in finitely many steps finds the farthest separating supporting-hyperplane, whose minimum-norm point is the desired minimum-norm point in the polytope. During the execution of the algorithm the distance of the separating supporting-hyperplane monotonically increases. The algorithm is closely related to P. Wolfe's primal algorithm which finds a sequence of norm-decreasing points in the given polytope. Computational experiments are carried out to show the behavior of our algorithm.

1. Introduction

P. Wolfe [11] developed a finite algorithm for finding the minimum-norm point in a convex polytope expressed as the convex hull of finitely many given points in an n-dimensional Euclidean space $\mathbf{R}^n$. The problem is to solve the following quadratic progromming problem:

$$\text{Minimize } \|X\| \tag{1.1a}$$

$$\text{subject to } X = \sum_{j=1}^{m} P_j w_j \tag{1.1b}$$

$$\sum_{j=1}^{m} w_j = 1 \tag{1.1c}$$

$$w_j \geq 0, \quad (j = 1, 2, \cdots, m), \tag{1.1d}$$

where P_j $(j = 1, 2, \cdots, m)$ are given points in $\mathbf{R}^n$ and $\| \bullet \|$ is the Euclidean norm in $\mathbf{R}^n$ Wolfe's algorithm finds a sequence of norm-decreasing points in the polytope by generating simplices contained in the polytope.

It should be noted here that the polytope under consideration is expressed as the convex hull of points but not as the intersection of halfspaces.

The minimum-norm point problem arises directly or indirectly in a lot of problems such as pattern recognition [1], the problem of mixing [5], the nondifferentiable function minimization (see [6]), and the submodular function minimization [3].

The minimum-norm point problem (1.1) is a special case of the quadratic programming problem (see [10]). The minimum-norm point problem and its variants have been consid-

ered by P. Wolfe [11], E. G. Gilbert [4], R. O. Barr [2], B. F. Mitchell, V. F. Dem'yanov and V. N. Malozemov [6], K. G. Murty and Y. Fathi [7], and others.

We propose a dual algorithm for the minimum-norm point problem in Section 2. Our algorithm keeps a supporting hyperplane which separates the polytope and the origin, repeatedly rotates the separating supporting-hyperplane in such a way that the distance of the hyperplane monotonically increases, and in finitely many steps finds the farthest separarting-hyperplane, whose minimum-norm point is the desired minimum-norm point in the polytope. Our dual approach also works for more general nonlinear objective functions.

Our dual algorithm initially requires a separating hyperplane between the polytope and the origin. In Section 4 we describe methods for treating the case where a separating hyperplane is difficult to find or may not exist. We also discuss the relationship between our algorithm and Wolfe's primal one.

In Section 5 we give results of computational experiments to show the behavior of the proposed algorithm.

2. A Dual Algorithm for the Minimum-Norm Point Problem

For any set Q of points in $\mathbf{R}^n$ we denote the *affine hull* of Q by $\mathrm{A}(Q)$ and the *convex hull* of Q by $\mathrm{C}(Q)$. For the definitions of affine hull, convex hull etc. in convex analysis see [8] and [9].

Consider the minimum-norm point problem described by (1.1) in Section 1. We denote by P the set of the given m points P_j $(j = 1, 2, \cdots, m)$ in $\mathbf{R}^n$.

We identify a hyperplane with the associated linear equation $C^{\top}X$ $(= c_1x_1 + \cdots + c_nx_n) = d$, where $C^{\top} = (c_1, \cdots, c_n)$, $X^{\top} = (x_1, \cdots, x_n)$ and $\top$ denotes the transpose. A hyperplane $C^{\top}X = d$ is called a *separating hyperplane* of polytope $\mathrm{C}(P)$ if either (i) $0 \leq d$ and $C^{\top}X \geq d$ for every X in $\mathrm{C}(P)$, or (ii) $0 \geq d$ and $C^{\top}X \leq d$ for every X in $\mathrm{C}(P)$. That is, $C^{\top}X = d$ is a separating hyperplane of $\mathrm{C}(P)$ if and only if the hyperplane *separates* $\mathrm{C}(P)$ and the origin in $\mathbf{R}^n$.

Also, a hyperplane $C^{\top}X = d$ is called a *supporting hyperplane* of $\mathrm{C}(P)$ if $C^{\top}X \geq d$ for every X in $\mathrm{C}(P)$ and there exists a point X_0 in $\mathrm{C}(P)$ such that $C^{\top}X_0 = d$. For a supporting hyperplane $C^{\top}X = d$ of $\mathrm{C}(P)$, a point X_0 in $\mathrm{C}(P)$ satisfying $C^{\top}X_0 = d$ is called a *supporting point*. A *separating supporting-hyperplane* of $\mathrm{C}(P)$ is a supporting hyperplane of $\mathrm{C}(P)$ which is also a separating hyperplane of $\mathrm{C}(P)$.

For any point X_0 in $\mathbf{R}^n$ we denote by $\mathrm{H}(X_0)$ the hyperplane $X_0^{\top}X = \| X_0 \|^2$.

Now, a dual algorithm is furnished as follows.

A Dual Algorithm

Input: A set P of points P_j $(j = 1, 2, \cdots, m)$ in $\mathbf{R}^n$, a separating supporting-hyperplane $C^{\top}X = d$ of $\mathrm{C}(P)$ with $\| C \| = 1$, and its supporting point $X_0 = P_J$ in P.

Output: The minimum-norm point X_0 of $\mathrm{C}(P)$.

Step 0: Put $Q := P_J$.

Step 1: If $\mathrm{H}(X_0)$ is a separating hyperplane, then stop (X_0 is the minimum-norm point in $\mathrm{C}(P)$). Otherwise, find the maximum value of λ $(0 \leq \lambda \leq 1)$ for which the hyperplane $\pi(\lambda)$ expressed by

$$((1 - \lambda)C + \lambda X_0)^{\top}(X - X_0) = 0 \qquad (0 \leq \lambda \leq 1) \tag{2.1}$$

is a separating hyperplane of $C(P)$. Let $\hat{\lambda}$ be the maximum value of such λ. Choose a point P_J in P lying on $\pi(\hat{\lambda}) \cap C(P)$ such that

$$X_0^{\top} P_J = \min \{ X_0^{\top} P_j \mid \quad j = 1, 2, \cdots, m; \quad P_j \in \pi(\hat{\lambda}) \cap C(P) \} \tag{2.2}$$

Put $Q := Q \cup \{P_J\}$, and if $\hat{\lambda} > 0$, then put $\hat{C} = (1 - \hat{\lambda})C + \hat{\lambda} X_0$ and $C := \hat{C} / \| \hat{C} \|$.

Step 2: Let Y be the minimum-norm point in $A(Q)$. If Y is in the relative interior of $C(Q)$, then put $X_0 := Y$ and go to Step 1. Otherwise go to Step 3.

Step 3: Let Z be the point on the line segment $C(Q) \cap \overline{X_0 Y}$ which is nearest to Y. Delete from Q the points not in the minimal face of $C(Q)$ on which Z lies. Put $X_0 := Z$ and go to Step 2.

(End)

Step 1 rotates the current separating supporting-hyperplane while keeping X_0 to be a supporting point of $C(P)$. The point, P_J, in (2.2) prevents the hyperplane from rotating. Moreover, it should be noted that, for P_J chosen in Step 1 by (2.2), $X = P_J$ minimizes the following linear function in X for a sufficiently small positive number ε :

$$((1 - (\hat{\lambda} + \varepsilon))C + (\hat{\lambda} + \varepsilon) X_0)^{\top} X. \tag{2.3}$$

Note that when (2.3) is minimized,

(1) X must lie on the hyperplane $\pi(\hat{\lambda})$ since ε is a sufficiently small positive number, and

(2) X minimizes $(X_0 - C)^{\top} X$ on $\pi(\hat{\lambda}) \cap C(P)$.

Since we have $0 \leq \lambda < 1$, minimizing $(X_0 - C)^{\top} X$ on $\pi(\hat{\lambda}) \cap C(P)$ is equivalent to minimizing $X_0^{\top} X$; to see this, eliminate $C^{\top} X$ in $(X_0 - C)^{\top} X$ by using the equation for $\pi(\hat{\lambda})$.

Informally, P_J in (2.2) is the point which prevents the current supporting hyperplane from rotating and which is farthest from the rotation axis through X_0.

It should also be noted that if $\hat{\lambda} > 0$ in Step 1, (2.2) is equivalent to

$$C^{\top} P_J = \max \{ C^{\top} P_j \mid \quad j = 1, 2, \cdots, m; \quad P_j \in \pi(\hat{\lambda}) \cap C(P) \} \tag{2.4}$$

since $0 < \hat{\lambda} < 1$ and $\pi(\hat{\lambda})$ is given by (2.1) with $\lambda = \hat{\lambda}$.

In Step 1 X_0 is in the relative interior of $C(Q)$. After augmenting Q as $Q := Q \cup \{P_J\}$ in Step 1, Steps 2 and 3 try to find a new point X_0 such that new X_0 is in the relative interior of the new $C(Q)$. For these new Q and X_0 the current separating supporting-hyperplane may be rotated.

By the rotation of the separating suporting-hyperplane the distance of the hyperplane (from the origin) increases.

We call the cycle formed by Steps 2 and 3 a *minor cycle* and the one formed by Step 1 and possible repeated minor cycles a *major cycle*.

The behavior of the algorithm is illustrated in Figure 2.1 and Table 2.1 for a simple two-dimensional example.

The computationally most dominating part of the algorithm is to find the minimum-norm point in the affine hull $A(Q)$ of Q in Step 2, where points in Q are affinely independent. As shown in [11], the minimum-norm point in $A(Q)$ is found by solving the following system of equations.

$$e^{\top} w = 1, \tag{2.5}$$

$$e\lambda + Q^{\top} Q w = 0, \tag{2.6}$$

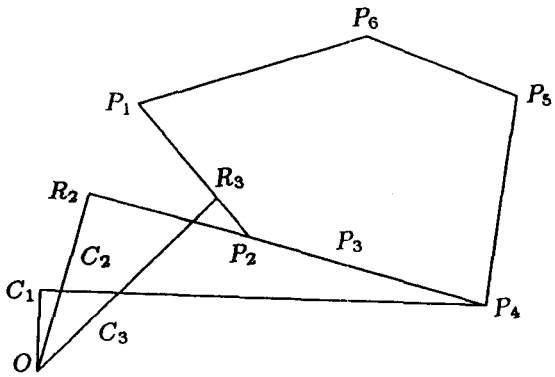

Figure 2.1

Table 2.1

number	Step	X	C	Q	Y
1	0	P_4	C_1	P_4	
2	1	P_4	C_1	P_4, P_2	
3	2	P_4	C_1	P_4, P_2	R_2
4	3	P_2	C_1	P_2	
5	2	P_2	C_2	P_2	P_2
6	1	P_2	C_2	P_2, P_1	
7	2	R_3	C_3	P_2, P_1	R_3
8	1			Stop	

where $e^T = (1, 1, \cdots, 1)$, Q should be regarded as the matrix consisting of column vectors which correspond to points in Q, and λ and $w = (w_1, \cdots, w_n)^T$ are, respectively, scalar and vector variables to be determined. Here Qw will be the minimum-norm point in $A(Q)$. As suggested by Wolfe [11], the system of equations (2.5) and (2.6) is solved by Cholesky decomposition with efficient updating when Q is changed (see [11]).

It should also be noted that in our dual algorithm we keep a separating supporting hyperplane $C^T X = d$ ($\| C \| = 1$) and a supporting point X_0 and that d and $\| X_0 \|$ are, respectively, the lower and the upper bound of the norm of the minimum-norm point in $C(P)$.

3. The Validity of the Dual Algorithm

In this section we show that the proposed dual algorithm finds the minimum-norm point in $C(P)$ in a finite number of steps.

Lemma 3.1: *Executing each major cycle does not decrease the distance of the separating supporting-hyperplane.*

(Proof) When the separating supporting-hyperplane is rotated (i.e., $\hat{\lambda} > 0$) in Step 1, the distance of the hyperplane increases, since the distance of the hyperplane $X_0^T (X - X_0) = 0$ is greater than that of the hyperplane $C^T (X - X_0) = 0$. Also Steps 2 and 3 do not change the current separating supporting-hyperplane.

Lemma 3.2: *Executing each major cycle decreases the norm* $\| X_0 \|$ *of* X_0 *unless the algorithm terminates.*

(Proof) Suppose that we are now carrying out Step 1 and the algorithm does not terminate there. Then for the point Y obtained in the next Step 2 we have $\| Y \| < \| X_0 \|$, since $X_0^\top (P_J - X_0) < 0$, i.e., P_J lies in the near side of the halfspace determined by the hyperplane $X_0^\top (X - X_0) = 0$. If we further move to Step 3, Z satisfies $\| Z \| < \| X_0 \|$ since $\| Y \| < \| X_0 \|$, so that $\| X_0 \|$ decreases. When we go back to Step 2, the current X_0 is in the relative interior of $\mathrm{C}(Q)$ and the new Y in Step 2 satisfies $\| Y \| \leq \| X_0 \|$. If Y is not in the relative interior of $\mathrm{C}(Q)$, then we have $\| Y \| < \| X_0 \|$. Therefore, the present lemma follows by induction.

□

At the begining of Step 1 we have a set $Q \subseteq P$ and a point X_0 in the relative interior of $\mathrm{C}(Q)$ such that X_0 is the minimum-norm point in the affine hull $\mathrm{A}(Q)$ of Q. Hence, the hyperplane $\pi(\lambda)$ in Step 1 for each λ $(0 \leq \lambda \leq 1)$ contains Q. We call such a set Q a *corral* (see [11]).

Theorem 3.3: *The dual algorithm terminates in a finite number of steps.*

(Proof) Each major cycle completes with at most $|Q|$ times repeating minor cycles of Steps 2 and 3, where Q is the one obtained at the begining of the major cycle and $|Q|$ denotes the number of points in Q. This is because each Step 3 reduces $|Q|$ at least by one.

From Lemma 3.2 each major cycle decreases $\| X_0 \|$, each corral Q obtained at the beginning of Step 1 uniquely determines X_0 (the minimum-norm point of the $\mathrm{A}(Q)$), and there are only finitely many corrals for the given set P. Hence the algorithm terminates in a finite number of steps. □

4. Initial Separating Supporting-Hyperplanes

The dual algorithm described in Section 2 requires an initial separating supporting-hyperplane and its supporting point. We shall show some methods to deal with the case where a separating supporting-hyperplane is difficult to find or may not exist.

We consider the $(n+1)$-dimensional Euclidean space $\mathbf{R}^{n+1}$ by increasing the dimension by one. For each point $P_j = (p_{1j}, \cdots, p_{nj})^\top$ in $\mathbf{R}^n$ define $P'_j = \{ (p_{1j}, \cdots, p_{nj}, 1) \}^\top$ $(j = 1, 2, \cdots, m)$. Note that if we are given the minimum-norm point $X'_0 = (x_{10}, \cdots, x_{n0}, 1)^\top$ in $\mathrm{C}(P')$ for $P' = \{ P'_1, \cdots, P'_m \}$, then $X_0 = (x_{10}, \cdots, x_{n0})^\top$ is the minimum-norm point in $\mathrm{C}(P)$.

Now, let $C^\top X = d$ be any supporting hyperplane of $\mathrm{C}(P)$ with X_0 being its supporting point. Then $C^\top X - dx_{n+1} = 0$ is a separating supporting-hyperplane of $\mathrm{C}(P')$ and $X'_0 = (X_0^\top, 1)^\top$ is a supporting point of $\mathrm{C}(P')$. For example, a supporting-hyperplane of $\mathrm{C}(P)$ can be given as follows. For any $k \in \{ 1, \cdots, n \}$ suppose

$$\min \{ p_{kj} \mid j = 1, 2, \cdots, m \} = p_{kj*} \tag{4.1}$$

for some $j* \in \{ 1, 2, \cdots, m \}$. Then, $x_k = p_{kj*}$ is a supporting hyperplane of $\mathrm{C}(P)$ with a supporting point P_{j*}. It should be noted that if $p_{kj*} \geq 0$ in (4.1), then $x_k = p_{kj*}$ is a separating supporting-hyperplane of $\mathrm{C}(P)$

We can also choose a separating supporting-hyperplane of $\mathrm{C}(P')$ in $\mathbf{R}^{n+1}$ as follows. Note that the hyperplane $x_{n+1} = 1$ is a separating supporting-hyperplane of $\mathrm{C}(P')$

and that any point P_j' is a supporting point ($j = 1, 2, \cdots, m$). If we choose a separating supporting-hyperplane in this way, the dual algorithm becomes essentially the same as Wolfe's primal algorithm and the separating supporting-hyperplane is not rotated throughout the algorithm. It is interesting to see that Wolfe's primal algorithm correspond to the dual algorithm with the above reduction.

5. Computational Experiments

We carry out computational experiments for two types of problems where m points P_j ($j = 0, 1, \cdots, m$) are given as follows.

Type 1: m points $P_j = (p_{1j}, \cdots, p_{nj})^\top$ ($j = 1, 2, \cdots, m$) are chosen at random from the sample space where each component p_{kj} ($k = 1, 2, \cdots, n$; $j = 1, 2, \cdots, m$) is uniformly distributed on $\{1, 2, \cdots, 50\}$.

Type 2: m points $P_j = (p_{1j}, \cdots, p_{nj})^\top$ ($j = 1, 2, \cdots, m$) are chosen at random from the sample space where each first component p_{1j} ($j = 1, 2, \cdots, m$) is uniformly distributed on [0.01-0.001, 0.01+0.001] and other components p_{kj} ($k = 2, 3, \cdots, n$; $j = 1, 2, \cdots, m$) are uniformly distributed on [-0.001, 0.001]. (This problem is also treated by Wolfe [11].)

Initial separating supporting-hyperplanes are found by the method described in Section 4. Note that we have $p_{kj*} \geq 0$ in (4.1) for the problems of types 1 and 2.

Figure 5.1 shows a sample behavior of vlues $\| X_0 \|$ and $C^\top X_0$ obtained by the dual algorithm for the problem of type 2; the average behavior is also shown in Figure 5.2.

Figure 5.3 shows the average number of major cycles for the problem of type 1, and Figure 5.4 for the problem of type 2. The dual algorithm requires more major cycles than the primal one for the problem of type 2. However, as shown in Figure 5.5, each set Q for the dual algorithm consists of significantly smaller number of points than for the primal one. Note that the reduction of the size of Q strongly contributes to the reduction of time for solving the system of equations (2.5) and (2.6).

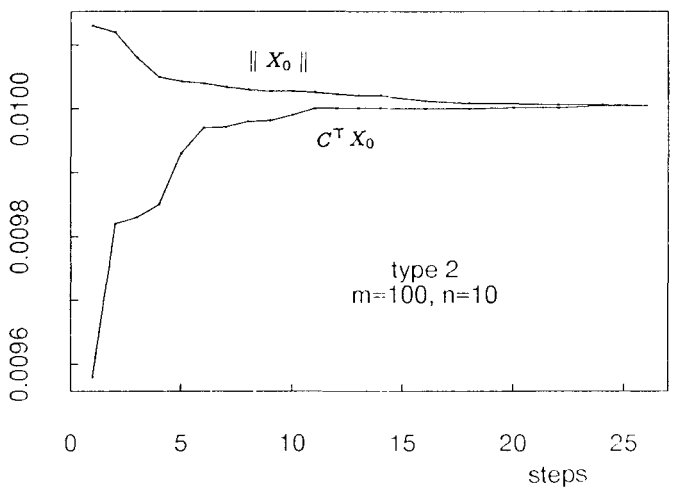

Figure 5.1.

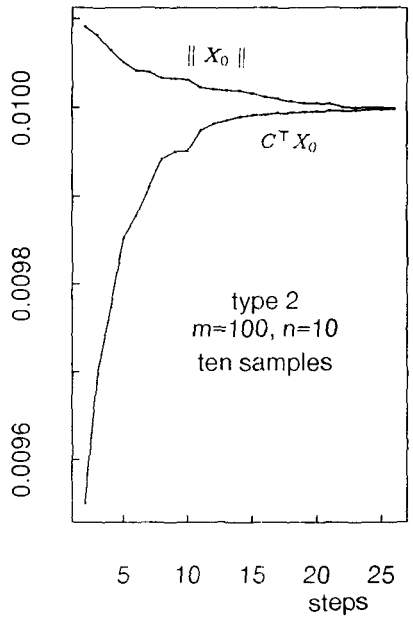

Figure 5.2.

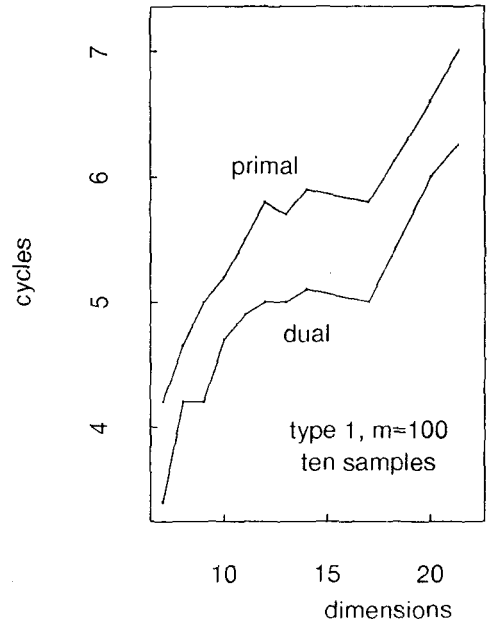

Figure 5.3.

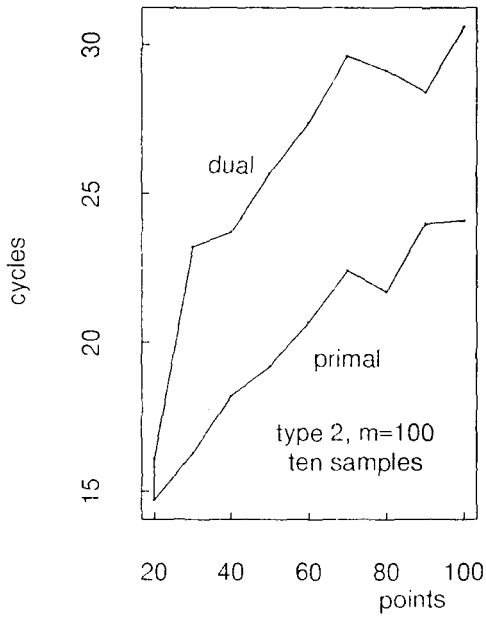

Figure 5.4.

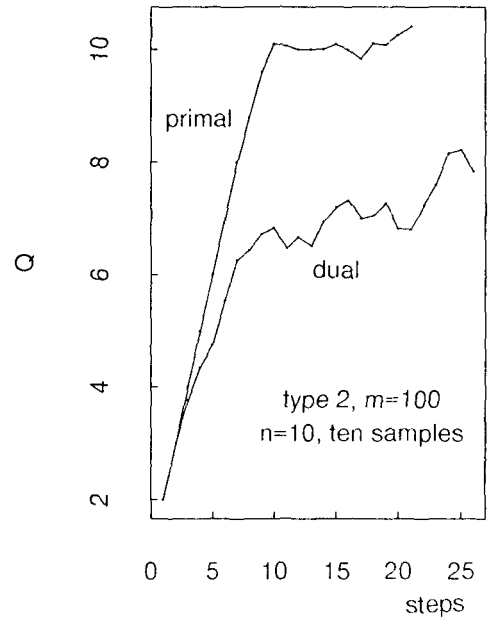

Figure 5.5.

We see from these computational experiments that the efficiency of the dual algorithm is comparable to primal one. However, the efficiency heavily depends on the way of choosing an initial separating supporting-hyperplane.

The dual algorithm may be effective in the sensitivity or parametric analysis with respect to the change of the data P_j $(j = 1, 2, \cdots, m)$.

It should also be noted that, though Wolfe's primal and the proposed dual algorithms are practically efficient, they may require exponential time as the simplex methods for linear and quadratic programs.

Acknowledgment: The present work is partly supported by a grant in aid of the Ministry of Education, Science and Culture of Japan.

References

[1] L. C. Barbosa and E. Wong: On a class of iterative algorithms for linear inequalities with applications to pattern classification, in: *Proceedings of the First Annual Princeton Conference of Information Sciences and Systems*, 1967, pp. 86 ~ 89.

[2] R. O. Barr: An efficient computional procedure for a generalized quadratic programming problem, *SIAM Journal on Control* **7** (1969) 415 ~ 429.

[3] S. Fujishige: Submodular systems and related topics, *Mathematical Programming Study* **22** (1984) 113 ~ 131.

[4] E. G. Gilbert: An iterative procedure for computing the minimum of a quadratic form on a convex set, *SIAM Journal on Control* **4** (1966) 61 ~ 80.

[5] E. Klafszky, J. Mayer and T. Terlaky: On mathematical programming model of mixing, *European Journal of Operation Research* **42** (1989) 254 ~ 267

[6] B. F. Mitchell, V. F. Dem'yanov and V. N. Malozemov: Finding the point of a polyhedron closest to the origin, *SIAM Journal on Control* **12** (1974) 19 ~ 26.

[7] K. G. Murty and Y. Fathi: A critical index algorithm for nearest point problems on simplicial cones, *Mathematical Programming* **23** (1978) 206 ~ 215.

[8] R. T. Rockafellar: *Convex Analysis* (Princeton University Press, Princeton, N. J., 1970).

[9] J. Stoer and C. Witzgall: *Convexity and Optimization in Finite Dimentions I* (Spring-Verlag Berlin, Heidelberg, New York, 1970)

[10] C. van de Panne: *Methods for Linear and Quadratic Programming* (North-Holland, Amsterdam, 1975).

[11] P. Wolfe: Finding the nearest point in a polytope, *Mathematical Programming* **11** (1976) 128 ~ 149.

Satoru FUJISHIGE and Ping ZHAN
Institute of Socio-Economic Planning,
University of Tsukuba,
Tsukuba, Ibaraki 305, Japan