Journal of the Operations Research
Society of Japan
Vol. 35, No. 4, December 1992

A DUAL ALGORITHM FOR FINDING A NEAREST PAIR OF POINTS IN TWO POLYTOPES

Satoru Fujishige   Ping Zhan
University of Tsukuba

(Received August 27, 1991; Revised April 8, 1992)

Abstract We propose a separating-hyperplane algorithm for finding a nearest pair of points in two polytopes, where each polytope is expressed as the convex hull of given points in a Euclidian space. The proposed algorithm is an extension of the authors dual algorithm for finding the minimum-norm point in a polytope.

1. Introduction

A primal algorithm for the problem of finding the minimum-norm point in a polytope is given by Wolfe [9] and a dual one for the problem by Fujishige and Zhan [3]. Very recently, Sekitani and Yamamoto [6] have devised a variant of our dual algorithm, which does not require an initial separating supporting-hyperplane but generates the same sequence of separating supporting-hyperplanes as ours after finding one. Also, an interior point algorithm is given by Takehara [8]. The minimum-norm point problem has a lot of applications such as pattern recognition [1], the problem of mixing [4], the nondifferentiable function minimization, the submodular function minimization [2] and the portfolio selection [8].

In this paper we consider a generalization of the minimum-norm point problem, i.e., the problem of finding a nearest pair of points in two polytopes, where each polytope is expressed as the convex hull of given points in a Euclidean space. This problem also has applications in the fields similar to those above mentioned for the minimum-norm point problem. The problem can be reduced to that of finding the minimum-norm point in a single polytope obtained by taking a vector difference of the given two polytopes. Direct implementation of this approach, however, requires a significant increase of the data size. We shall propose a dual algorithm that avoids this difficulty by means of separating hyperplanes, which is an extension of our algorithm given in [3].

In Section 2 we precisely describe the problem of finding a nearest pair of points in two polytopes and show some preliminary results. Based on the results, we propose, in Section 3, a dual algorithm and prove the validity of the algorithm. In Section 4 we also carry out computational experiments to examine the behavior of the algorithm.

2. Preliminaries

Given two finite point sets $Q = \{q_i \mid i \in I\}$ and $R = \{r_j \mid j \in J\}$ in the n-dimensional Euclidian space $\mathbf{R}^n$, we consider the problem of finding a nearest pair of points in two polytopes $\mathrm{C}(Q)$ and $\mathrm{C}(R)$, the convex hulls of Q and R, respectively (see [5] and [7] for terminology about convex analysis). That is, the problem is to solve:

(2.1)
$$\begin{aligned} &\text{Minimize} && \| x - y \| \\ &\text{subject to} && x = \sum_{i\in I} v_i q_i, \qquad y = \sum_{j\in J} w_j r_j, \end{aligned}$$

$$\sum_{i\in I} v_i = 1, \qquad \sum_{j\in J} w_j = 1,$$
$$v_i \geq 0 \quad (i \in I), \quad w_j \geq 0 \quad (j \in J).$$

Here, $\| \cdot \|$ denotes the Euclidean norm and each vector in $\mathbf{R}^n$ is considered as an n-dimensional column vector. When either Q or R consists of only one point, the present problem becomes that of finding the minimum-norm point in a polytope, for which we gave a dual algorithm in [3]. In this paper we shall extend the dual algorithm to solve the above problem (2.1).

Define

(2.2) $$g(d) = \min \{d^{\mathsf{T}}(q_i - r_j) \mid i \in I,\ j \in J\}$$

for $d \in \mathbf{R}^n$ with $\| d \| = 1$, where $^{\mathsf{T}}$ denotes the matrix transpose. When $g(d) \geq 0$, d is normal to a plane which separates $\mathrm{C}(Q)$ from $\mathrm{C}(R)$.

Lemma 2.1: *For any $x \in \mathrm{C}(Q)$, $y \in \mathrm{C}(R)$ and $d \in \mathbf{R}^n$ with $\| d \| = 1$ we have*

(2.3) $$g(d) \leq \| x - y \| .$$

If $x \in \mathrm{C}(Q)$, $y \in \mathrm{C}(R)$ and $d \in \mathbf{R}^n$ with $\| d \| = 1$ satisfy $g(d) = \| x - y \|$, then the pair of x and y is an optimal solution of Problem (2.1).

(Proof) Choose any $x \in \mathrm{C}(Q)$, $y \in \mathrm{C}(R)$ and $d \in \mathbf{R}^n$ with $\| d \| = 1$. Let x and y be expressed as convex combinations $x = \sum_{i\in I} v_i q_i$ and $y = \sum_{j\in J} w_j r_j$ of Q and R, respectively, as in (2.1). Then, we have

(2.4) $$\begin{aligned} \| x - y \| &\geq d^{\mathsf{T}}(x - y) \\ &= d^{\mathsf{T}}(\sum_{i\in I} v_i q_i - \sum_{j\in J} w_j r_j) \\ &= \sum_{i\in I}\sum_{j\in J} v_i w_j d^{\mathsf{T}}(q_i - r_j) \\ &\geq \min\{d^{\mathsf{T}}(q_i - r_j) \mid i \in I,\ j \in J\} \\ &= g(d), \end{aligned}$$

where note that $\sum_{i\in I}\sum_{j\in J} v_i w_j = 1$ and $v_i w_j \geq 0$ $(i \in I,\ j \in J)$. Inequality (2.3) holds for any such x, y and d. Therefore, if we have $g(d) = \| x - y \|$, the pair (x, y) attains the minimum of (2.1) (and d attains the maximum of g). □

The following lemma (Lemma 2.2) shows a relation between Problem (2.1) and its dual problem of maximizing (2.2) and provides a sufficient condition for the optimality of the solutions of Problem (2.1).

For a polytope W and a point $x_0 \in W$ we define $\mathrm{N}(x_0, W)$ by

(2.5) $$\mathrm{N}(x_0, W) = \{d \mid d \in \mathbf{R}^n,\ \| d \| = 1,\ \forall x \in W : d^{\mathsf{T}}(x - x_0) \leq 0.\}$$

When $\mathrm{N}(x_0, W) \neq \emptyset$, $d \in \mathrm{N}(x_0, W)$ is a *normal vector* of a supporting hyperplane of the polytope W with a supporting point x_0. The following (i) and (ii) hold:

(i) $\mathrm{N}(x_0, W)$ is not empty if and only if x_0 is a boundary point of W.

(ii) $\mathrm{N}(x_0, W)$ contains more than one vector when x_0 is an extreme point of polytope W.

Now, we show the following lemma.

Lemma 2.2 (Optimality): *Suppose that we have $x_0 \in \mathrm{C}(Q)$ and $y_0 \in \mathrm{C}(R)$ such that $x_0 \neq y_0$. Define $d = (x_0 - y_0)/ \| x_0 - y_0 \|$. Then, the pair of x_0 and y_0 is a nearest pair of points of polytopes $\mathrm{C}(Q)$ and $\mathrm{C}(R)$ if $-d \in \mathrm{N}(x_0, \mathrm{C}(Q))$ and $d \in \mathrm{N}(y_0, \mathrm{C}(R))$.*

(Proof) Suppose $-d \in \mathrm{N}(x_0, \mathrm{C}(Q))$ and $d \in \mathrm{N}(y_0, \mathrm{C}(R))$. From the definition of d we have

$$(2.6) \qquad d^{\mathsf{T}}(q_i - x_0) \geq 0 \qquad (i \in I)$$

and hence

$$(2.7) \qquad \min\{d^{\mathsf{T}}(q_i - x_0) \mid i \in I\} \geq 0.$$

On the other hand, since x_0 is expressed as a convex combination $x_0 = \sum_{i \in I} v_i q_i$ of Q, we have

$$(2.8) \qquad 0 = \sum_{i \in I} v_i d^{\mathsf{T}}(q_i - x_0) \geq \min\{d^{\mathsf{T}}(q_i - x_0) \mid i \in I\}.$$

From (2.7) and (2.8),

$$(2.9) \qquad \min\{d^{\mathsf{T}}(q_i - x_0) \mid i \in I\} = 0.$$

Similarly, we have

$$(2.10) \qquad \min\{-d^{\mathsf{T}}(r_j - y_0) \mid j \in J\} = 0.$$

From (2.9) and (2.10) we have

$$(2.11) \qquad \min\{d^{\mathsf{T}}(q_i - x_0) \mid i \in I\} + \min\{-d^{\mathsf{T}}(r_j - y_0) \mid j \in J\} = 0,$$

i.e.,

$$(2.12) \qquad \min\{d^{\mathsf{T}}(q_i - r_j) \mid i \in I,\ j \in J\} = d^{\mathsf{T}}(x_0 - y_0).$$

It follows from (2.12) and the definition of d that we have $g(d) = \| x_0 - y_0 \|$. We see from Lemma 2.1 that (x_0, y_0) is a nearest pair of points of $\mathrm{C}(Q)$ and $\mathrm{C}(R)$. □

Note that Problem (2.1) may have more than one optimal solution. We are interested only in finding one optimal solution which satisfies the optimality condition of Lemma 2.2.

Wolfe's algorithm [9] and ours [3] can solve Problem (2.1) by considering the set P of $| I | \times | J |$ points p_{ij} given by

$$(2.13) \qquad p_{ij} = q_i - r_j \quad (i \in I,\ \ j \in J)$$

due to the following.

Proposition 2.3 (Carathéodory): *For $x_0 \in \mathrm{C}(Q)$ and $y_0 \in \mathrm{C}(R)$, we have*

$$(2.14) \qquad (x_0 - y_0)^{\mathsf{T}}((q_i - r_j) - (x_0 - y_0)) \geq 0 \quad (i \in I,\ j \in J)$$

if and only if

$$(2.15) \qquad (x_0 - y_0)^{\mathsf{T}}(q_i - x_0) \geq 0 \qquad (i \in I)$$

and

$$(2.16) \qquad (y_0 - x_0)^{\mathsf{T}}(r_j - y_0) \geq 0 \qquad (j \in J).$$

□

The dual algorithm given in the next section will work more efficiently than the dual algorithm which simply uses the set P of points of (2.13) and then applies the dual algorithm [3] to the polytope $\mathrm{C}(P)$.

3. A Dual Algorithm for Finding a Nearest Pair of Points

A set W of points in $\mathbf{R}^n$ is called a *corral* ([9]) if W is affinely independent and the minimum-norm point of the affine hull $\mathrm{A}(W)$ of W lies in the relative interior of $\mathrm{C}(W)$ (see [7] for the basic terminology). For $K \subseteq I \times J$, define $I_K = \{i \mid (i,j) \in K\}$ and $J_K = \{j \mid (i,j) \in K\}$. For $K \subseteq I \times J$, we call $Co(K) = \{q_i - r_j \mid (i,j) \in K\}$ a *pair corral* for Q and R if $Co(K)$ is a corral and for some $x_0 \in \mathrm{C}(\{q_i \mid i \in I_K\})$ and $y_0 \in \mathrm{C}(\{r_j \mid j \in J_K\})$ the following (3.1) and (3.2) hold:

$$(3.1) \qquad (x_0 - y_0)^\top (q_i - x_0) \geq 0 \qquad (i \in I_K),$$

$$(3.2) \qquad (y_0 - x_0)^\top (r_j - y_0) \geq 0 \qquad (j \in J_K).$$

We have (3.1) and (3.2) if and only if (x_0, y_0) is a nearest pair of points of polytopes $\mathrm{C}(\{q_i \mid i \in I_K\})$ and $\mathrm{C}(\{r_j \mid j \in J_K\})$. Here, note that we may not have $K = I_K \times J_K$ in general.

Now, we give a dual algorithm for finding a nearest pair of points of $\mathrm{C}(Q)$ and $\mathrm{C}(R)$. We assume that we are given an initial pair of separating supporting-hyperplanes of Q and R. We will show how to get one such pair at the end of this section.

A Dual Algorithm

Input: Point sets $Q = \{q_i \mid i \in I\}$ and $R = \{r_j \mid j \in J\}$ in $\mathbf{R}^n$. Two separating supporting-hyperplanes $c^\top(x - x_0) = 0$ and $(-c)^\top(y - y_0) = 0$ (x, y: variable vectors) with supporting points $x_0 = q_{i_0}$ ($i_0 \in I$) and $y_0 = r_{j_0}$ ($j_0 \in J$) such that $\| c \| = 1$ and $c^\top(q_i - y_0) \geq 0 \geq c^\top(r_j - x_0)$ for all $i \in I$ and $j \in J$.
Output: A nearest pair of points $\hat{x}_0 \in \mathrm{C}(Q)$ and $\hat{y}_0 \in \mathrm{C}(R)$; $\hat{x}_0$ and $\hat{y}_0$ satisfy

$$(3.3) \qquad (\hat{x}_0 - \hat{y}_0)^\top (q_i - \hat{x}_0) \geq 0 \qquad (i \in I),$$

$$(3.4) \qquad (\hat{y}_0 - \hat{x}_0)^\top (r_j - \hat{y}_0) \geq 0 \qquad (j \in J).$$

Step 0: Put $K := \{(i_0, j_0)\}$ and $Co := \{q_{i_0} - r_{j_0}\}$.
Step 1: (a) If (3.3) and (3.4) hold, then $(\hat{x}_0, \hat{y}_0) = (x_0, y_0)$ is a nearest pair of points of $\mathrm{C}(Q)$ and $\mathrm{C}(R)$ and stop.

(b) If Co is a pair corral, simultaneously rotate the current supporting hyperplanes $\pi_Q(\lambda)$ ($\lambda = 0$) and $\pi_R(\lambda)$ ($\lambda = 0$) around x_0 and y_0, respectively, as

$$(3.5) \qquad \pi_Q(\lambda): \quad ((1-\lambda)c + \lambda(x_0 - y_0))^\top (x - x_0) = 0,$$

$$(3.6) \qquad \pi_R(\lambda): \quad ((1-\lambda)(-c) + \lambda(y_0 - x_0))^\top (y - y_0) = 0$$

by increasing λ from 0 to 1 so that

$$(3.7) \qquad ((1-\lambda)c + \lambda(x_0 - y_0))^\top (q_i - x_0) \geq 0 \ (i \in I),$$

$$(3.8) \qquad ((1-\lambda)(-c) + \lambda(y_0 - x_0))^\top (r_j - y_0) \geq 0 \ (j \in J).$$

Choose the maximum value of λ which satisfies (3.7) and (3.8), and denote it by $\hat{\lambda}$. Choose a point q_{i^*} in Q lying on $\pi_Q(\hat{\lambda}) \cap \mathrm{C}(Q)$ which satisfies

$$(3.9) \qquad (x_0 - y_0)^\top q_{i^*} = \min\{(x_0 - y_0)^\top q_i \mid i \in I, \ q_i \in \pi_Q(\hat{\lambda}) \cap \mathrm{C}(Q)\}$$

and choose r_{j^*} in R lying in $\pi_R(\hat{\lambda}) \cap \mathrm{C}(R)$ which satisfies

$$(3.10) \qquad (y_0 - x_0)^\top r_{j^*} = \min\{(y_0 - x_0)^\top r_j \mid j \in J, \ r_j \in \pi_R(\hat{\lambda}) \cap \mathrm{C}(R)\}.$$

Put $Co := Co \cup \{q_{i^*} - r_{j^*}\}$, $K := K \cup \{(i^*, j^*)\}$, $\hat{c} = (1-\hat{\lambda})c + \hat{\lambda}(x_0 - y_0)$ and $c := \hat{c} / \parallel \hat{c} \parallel$. Go to Step 2.

(c) If Co is not a pair corral, choose q_{i_1} in $\{q_i \mid i \in I_K\}$ and r_{j_1} in $\{r_j \mid j \in J_K\}$ which satisfy

$$(x_0 - y_0)^\top q_{i_1} = \min\{(x_0 - y_0)^\top q_i \mid i \in I_K\}, \tag{3.11}$$

$$(y_0 - x_0)^\top r_{j_1} = \min\{(y_0 - x_0)^\top r_j \mid j \in J_K\}. \tag{3.12}$$

(Here, we do not change the current supporting hyperplanes.) Put $Co := Co \cup \{q_{i_1} - r_{j_1}\}$, $K := K \cup \{(i_1, j_1)\}$ and go to Step 2.

Step 2: (a) Let z be the minimum-norm point in $\mathrm{A}(Co)$.

(b) If z is not in the relative interior of $\mathrm{C}(Co)$, go to Step 3.

(c) Let z be expressed as a convex combination $z = \sum_{(i,j)\in K} w_{ij}(q_i - r_j)$, and put $x_0 = \sum_{(i,j)\in K} w_{ij} q_i$ and $y_0 = \sum_{(i,j)\in K} w_{ij} r_j$. Go to Step 1.

Step 3: Let w be the point on the line segment $\mathrm{C}(Co) \cap \overline{(x_0 - y_0)z}$ which is nearest to z, where $\overline{(x_0 - y_0)z}$ denotes the line segment with end points $x_0 - y_0$ and z. Delete from Co the points not in the minimal face of $\mathrm{C}(Co)$ on which w lies, and update K. Go to Step 2. (End)

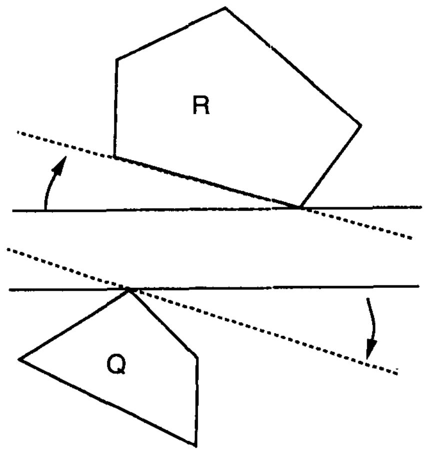

Figure 3.1

In Step 1 (b) we rotate the current supporting hyperplanes $\pi_Q(\lambda)$ and $\pi_R(\lambda)$ around x_0 and y_0, respectively, by increasing λ to obtain a farther pair of supporting hyperplanes (see Figure 3.1). When we obtain the farthest pair of supporting hyperplanes, the corresponding primal pair (x_0, y_0) is a nearest one. If Co is not a pair corral (in Step 1 (c)), rotating the current supporting hyperplanes $\pi_Q(\lambda)$ and $\pi_R(\lambda)$ around x_0 and y_0 may not yield supporting hyperplanes, so that in Steps 2 and 3 we change Co into a pair corral, a subset of Co.

Now, we show the validity of the algorithm.

Lemma 3.1: *At the beginning of each execution of Step 1 we have for current x_0, y_0 and $Co = \{q_i - r_j \mid (i,j) \in K\}$*

(1) *q_i ($i \in I_K$) and r_j ($j \in J_K$) lie on the current separating hyperplanes for Q and R, respectively,*

(2) *Co is affinely independent,*

(3) *$x_0 - y_0$ is the minimum-norm point of $\mathrm{A}(Co)$ lying in the relative interior of $\mathrm{C}(Co)$.*

(Proof) (1)~(3) clearly hold at the beginning of the first execution of Step 1. So, suppose that (1)~(3) hold at the beginning of the kth execution of Step 1 ($k \geq 1$). We show that (1)~(3) hold at the beginning of the $(k+1)$th execution of Step 1.

Proof of (1): From (3) we have

$$(x_0 - y_0)^{\mathsf{T}}((q_i - r_j) - (x_0 - y_0)) = 0 \qquad ((i,j) \in K). \tag{3.13}$$

Also, from (1),

$$c^{\mathsf{T}}(q_i - x_0) = 0 \qquad (i \in I_K), \tag{3.14}$$

$$c^{\mathsf{T}}(r_j - y_0) = 0 \qquad (j \in J_K). \tag{3.15}$$

For $\hat{\lambda}$ determined in Step 1 (b) define

$$\hat{c} = (1 - \hat{\lambda})c + \hat{\lambda}(x_0 - y_0). \tag{3.16}$$

It follows from (3.13) ~ (3.16) that

$$\hat{c}^{\mathsf{T}}((q_i - r_j) - (x_0 - y_0)) = 0 \qquad ((i,j) \in K). \tag{3.17}$$

We have from (3.7), (3.8) and (3.17)

$$\hat{c}^{\mathsf{T}}((q_i - x_0) = 0 \qquad (i \in I_K), \tag{3.18}$$

$$\hat{c}^{\mathsf{T}}(r_j - y_0) = 0 \qquad (j \in J_K). \tag{3.19}$$

This implies that (1) holds at the end of the execution of Step 1 (b). During the execution of Steps 1 (c), 2 and 3 only the deletion of elements of I_K and J_K is possible in Step 3 and the current separating hyperplanes $\pi_Q(\hat{\lambda})$ and $\pi_R(\hat{\lambda})$ are not changed. Hence, (1) holds at the beginning of the $(k+1)$th execution of Step 1.

Proof of (2): In Step 1 (b), since (x_0, y_0) is not a nearest pair of C(Q) and C(R), we have

$$(x_0 - y_0)^{\mathsf{T}}(q_{i^*} - x_0) \leq 0, \tag{3.20}$$

$$(y_0 - x_0)^{\mathsf{T}}(r_{j^*} - x_0) \leq 0, \tag{3.21}$$

where at least one of (3.20) and (3.21) holds with strict inequality. Hence,

$$(x_0 - y_0)^{\mathsf{T}}((q_{i^*} - r_{j^*}) - (x_0 - y_0)) < 0. \tag{3.22}$$

It follows from (3.13) and (3.22) that Co augmented at the end of Step 1 (b) is affinely independent.

In Step 1 (c), Co is a corral but not a pair corral. Hence for the current x_0 and y_0 we have

$$(x_0 - y_0)^{\mathsf{T}}((q_i - r_j) - (x_0 - y_0)) = 0 \qquad ((i,j) \in K). \tag{3.23}$$

From (3.23) and the definitions of q_{i_1} and r_{j_1} in (3.11) and (3.12), we have for any $(i,j) \in K$

$$\begin{aligned} 0 &= (x_0 - y_0)^{\mathsf{T}}(q_i - x_0) + (y_0 - x_0)^{\mathsf{T}}(r_j - y_0) \\ &\geq (x_0 - y_0)^{\mathsf{T}}(q_{i_1} - x_0) + (y_0 - x_0)^{\mathsf{T}}(r_{j_1} - y_0) \end{aligned} \tag{3.24}$$

If the value of the right-hand side of (3.24) is negative, then Co augmented in Step 1 (c) is affinely independent because of (3.23). Otherwise we would have

$$(x_0 - y_0)^{\mathsf{T}}(q_{i_1} - x_0) + (y_0 - x_0)^{\mathsf{T}}(r_{j_1} - y_0) = 0, \tag{3.25}$$

which would lead us to a contradiction as follows. Since Co is not a pair corral, one of the two terms in the left-hand side of (3.25) is negative and, hence, the other term is positive. So, suppose

$$(x_0 - y_0)^\top (q_{i_1} - x_0) > 0. \tag{3.26}$$

(The case where $(y_0 - x_0)^\top (r_{j_1} - y_0) > 0$ can be treated similarly.) From (3.11) and (3.26) we have

$$(x_0 - y_0)^\top (q_i - x_0) > 0 \qquad (i \in I_K). \tag{3.27}$$

However, since $x_0 = \sum_{(i,j)\in K} w_{ij} q_i$, (3.27) would imply

$$0 < \sum_{(i,j)\in K} w_{ij}(x_0 - y_0)^\top (q_i - x_0) = (x_0 - y_0)^\top (x_0 - x_0) = 0, \tag{3.28}$$

a contradiction.

Proof of (3): When we go from Step 2 to Step 1, (3) clearly holds. □

We see from (1) in Lemma 3.1 that x_0 and y_0 in Step 1 (b) are supporting points of C(Q) and C(R), respectively.

Lemma 3.2: *The value of $\|x_0 - y_0\|$ is made decrease every time x_0 and y_0 are updated in Step 2.* □

The proof of Lemma 3.2 is the same as given in [9] and [3].

The value of $\| x_0 - y_0 \|$ is uniquely determined for each corral Co and there are only finitely many corrals Co. It follows from Lemma 3.2 that the algorithm terminates in a finite number of steps.

It should be noted that the distance between the separating supporting-hyperplanes increases if they rotate, i.e., $\hat{\lambda} > 0$ in Step 1 (b). Also, note that in (3.9) and (3.10) we search the minima on at most $| I |$ and $| J |$ points in Q and R, respectively.

If separating supporting-hyperplanes of Q and R are not available (or may not exist), then we consider the $(n-1)$-dimensional Euclidean space $\mathbf{R}^{n+1}$ and place Q in the hyperplane $x(n+1) = 0$ and R in the hyperplane $x(n+1) = \gamma > 0$ in $\mathbf{R}^{n+1}$, i.e., we consider $Q' = \{(q^\top, 0)^\top \mid q \in Q\}$ and $R' = \{(r^\top, \gamma)^\top \mid r \in R\}$ in $\mathbf{R}^{n+1}$ (see Figure 3.2). These two

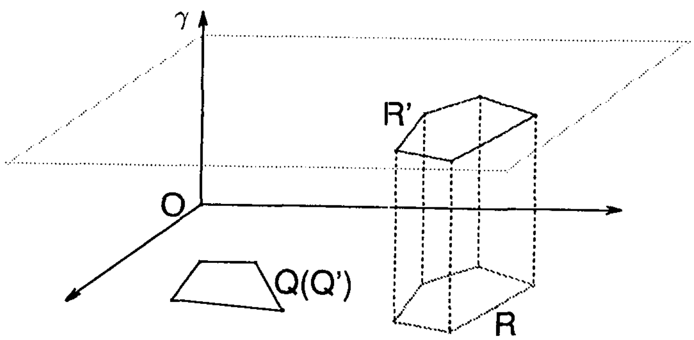

Figure 3.2

hyperplanes $x(n+1)=0$ and $x(n+1)=\gamma$ are trivial separating supporting-hyperplanes of Q' and R', but other separating supporting-hyperplanes can also easily be found by choosing sufficiently large constant γ for the hyperplane $x(n+1)=\gamma$. We see that finding a nearest pair $(x_0,\ y_0)$ of points of Q' and R' in $\mathbf{R}^{n+1}$ gives the pair of vectors $(x_0(1),\cdots,x_0(n))^{\top}$ and $(y_0(1),\cdots,y_0(n))^{\top}$, which is a nearest one for Q and R in $\mathbf{R}^n$, and that if $x_0(k)=y_0(k)$ $(k=1,\cdots,n)$, there is no separating hyperplane of Q and R.

4. Computational Experiments

We carry out computational experiments to examine the behavior of the proposed dual algorithm. we consider the following two types of data:

Type 1: l points $q_i=(q_{i1},\cdots,q_{in})^{\top}$ $(i=1,2,\cdots,l)$ are independently chosen in such a way that q_{i1} is uniformly distributed on [10,30] and that other components q_{ik} $(k=2,\cdots,l)$ are uniformly distributed on [0,20]. Also, m points $r_j=(r_{j1},\cdots,r_{jn})^{\top}$ $(j=1,2,\cdots,m)$ are independently chosen in such a way that r_{j1} is uniformly distributed on $[-30,-10]$ and that r_{jk} $(k=2,\cdots,m)$ are uniformly distributed on [0,20].

Type 2: l points $q_i=(q_{i1},\cdots,q_{in})^{\top}$ $(i=1,2,\cdots,l)$ are independently chosen in such a way that q_{i1} is uniformly distributed on $[-10^{-5}+10^{-2},10^{-5}+10^{-2}]$ and that other components q_{ik} $(k=2,\cdots,l)$ are uniformly distributed on $[-10^{-3},10^{-3}]$. Also m points $(r_{j1},\cdots,r_{jn})^{\top}$ $(j=1,2,\cdots,m)$ are uniformly chosen in such a way that r_{j1} is uniformly distributed on $[-10^{-5}-10^{-2},10^{-5}-10^{-2}]$ and that r_{jk} $(k=2,\cdots,m)$ are uniformly distributed on $[-10^{-3},10^{-3}]$.

We choose $c=(1,0,\cdots,0)^{\top}$ as the normal vector of the initial separating hyperplanes for both Types 1 and 2.

Figures 4.1 and 4.2 show sample behaviors of the algorithm for problems of Types 1 and 2 with $l=m=50$ and $n=10$. Average behaviors of the corresponding ten samples is given in Figures 4.3 and 4.4. These figures show that the duality gap, $\| x_0-y_0 \| -c^{\top}(x_0-y_0)$, rapidly diminishes.

Figure 4.5 shows ten-sample average behaviors of the number of major cycles (including Step 1 (b)) versus the number $l(=m)$ of points of Q (and R), for Types 1 and 2. It is seen that the number of major cycles grows sublinearly with respect to the number of points.

We also carried out experiments on larger-scale problems. Figure 4.6 shows a sample behavior of the algorithm for a problem of Type 1 with $l=m=1000$ and $n=50$. Its ten-sample average behavior is given in Figure 4.7. Figure 4.8 illustrates a ten-sample average behavior of $l(=m)$ versus the number of major cycles for Type 1. The running time required for solving the problems is 39 seconds for $l(=m)=300$ and n =50, 75 seconds for $l(=m)$=500 and n =50, and 150 seconds for $l(=m)$=1000 and n =50 by SUN 4/260 in the average of ten samples.

The above experimental results show the practicality of our dual algorithm. As are dual algorithms for the ordinary mathematical programming problems, our dual algorithm is also effective for the sensitivity or parametric analysis with respect to the change of the data points of Q or R. Note that if Q or R is changed in such a way that one of the final optimal separating supporting-hyperplanes is still separating, then we can start our algorithm with the separating hyperplane. Deleting some points of Q or R is a typical such changing.

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

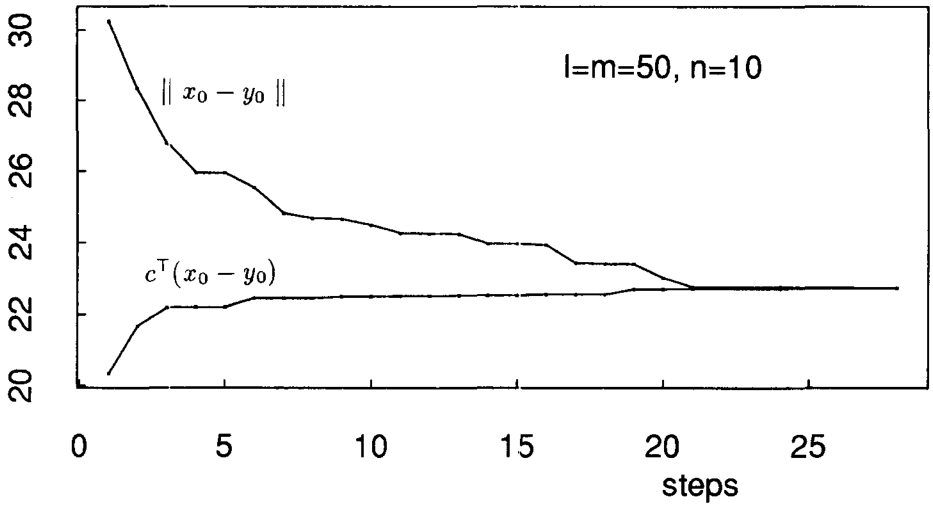

Figure 4.1. A sample behavior for Type 1.

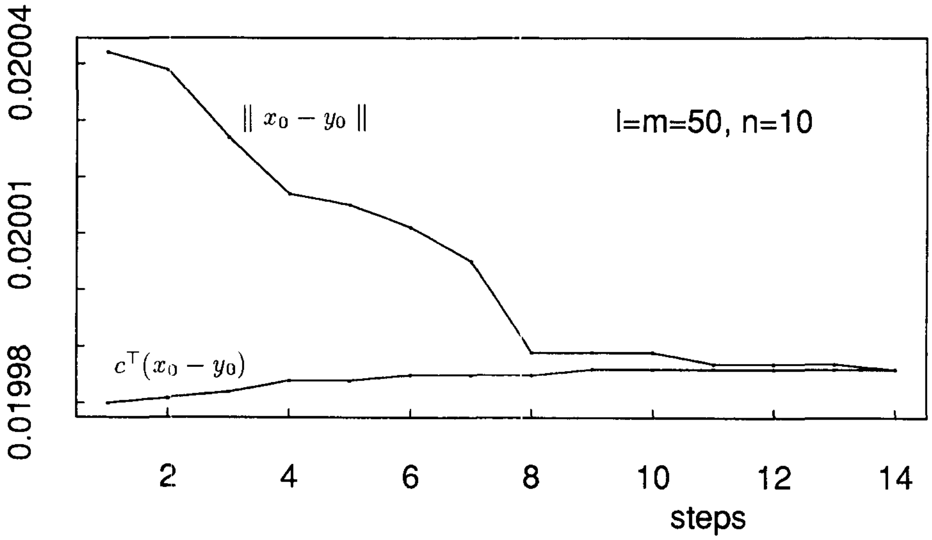

Figure 4.2.A sample behavior for Type 2.

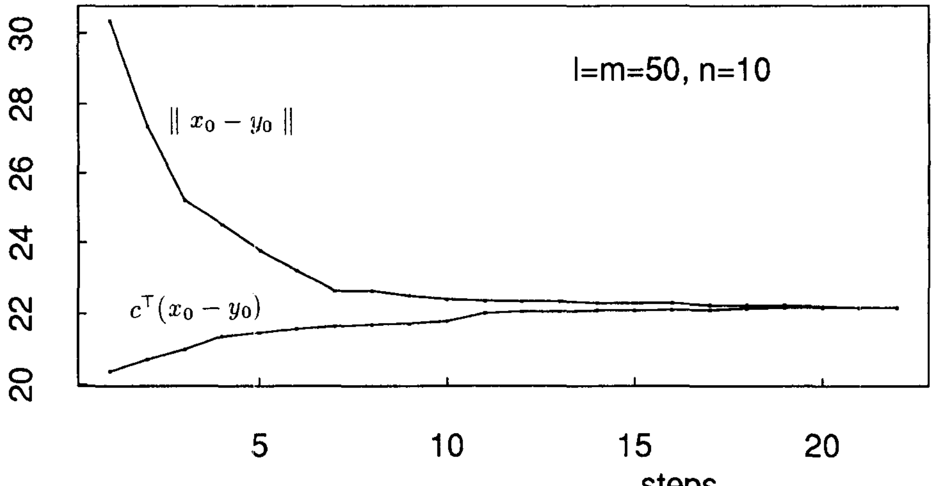

Figure 4.3. A ten-sample average behavior for Type 1.

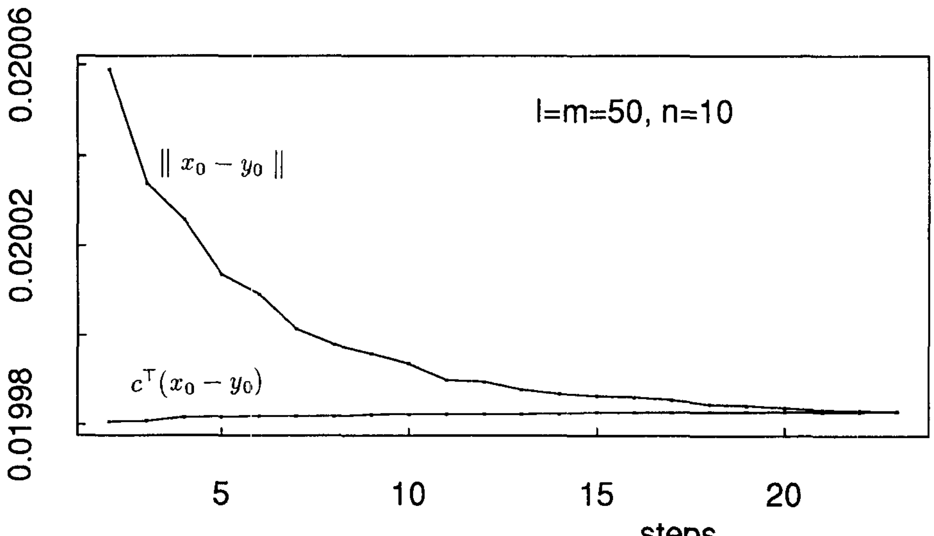

Figure 4.4. A ten-sample average behavior for Type 2.

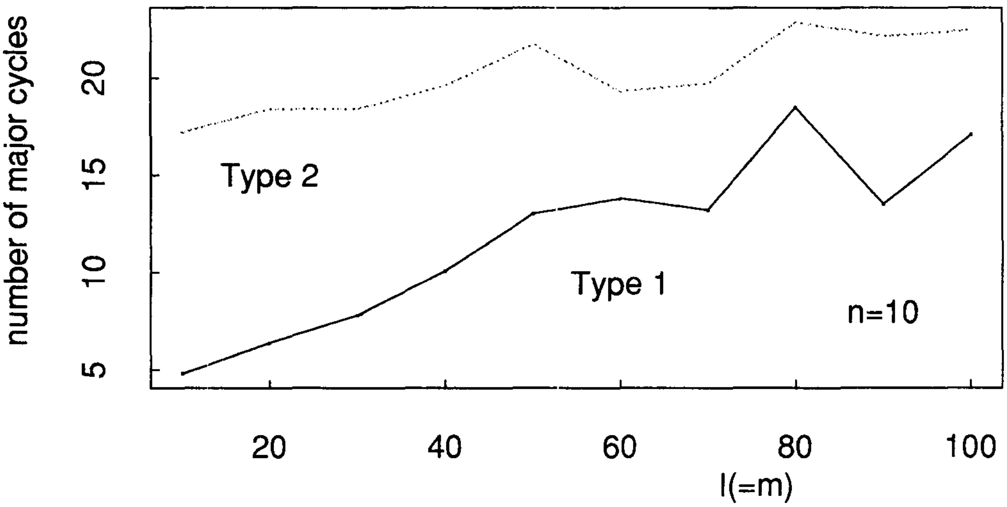

Figure 4.5. The number of major cycles versus
the number of points for Types 1 and 2 (ten-sample average).

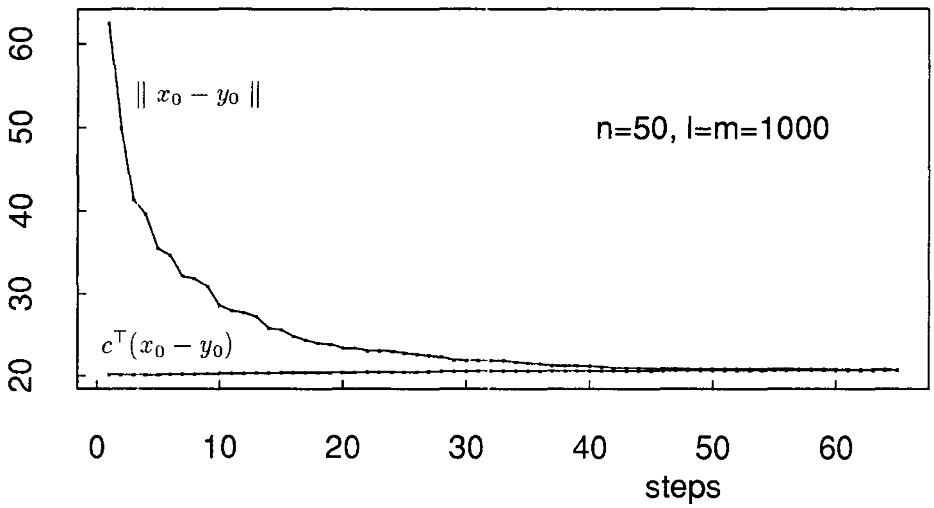

Figure 4.6. A sample behavior for Type 1.

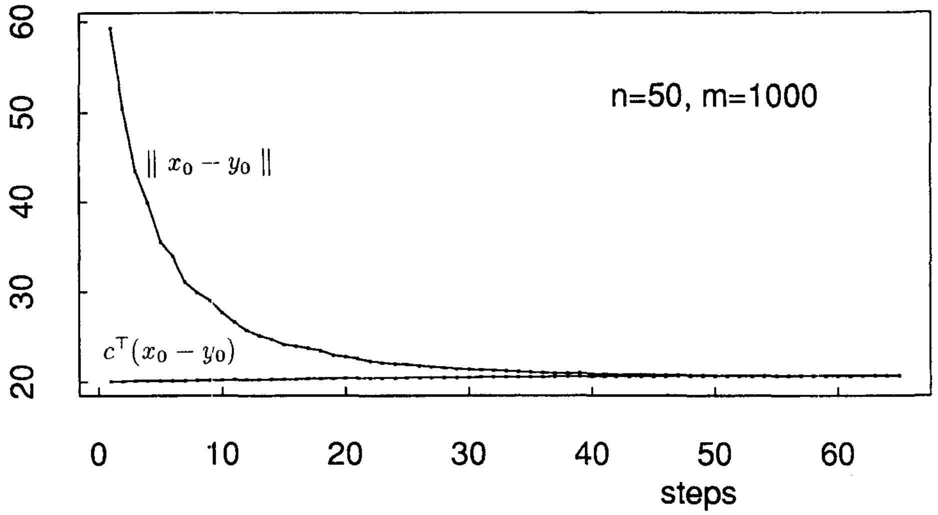

Figure 4.7. A ten-sample average behavior for Type 1.

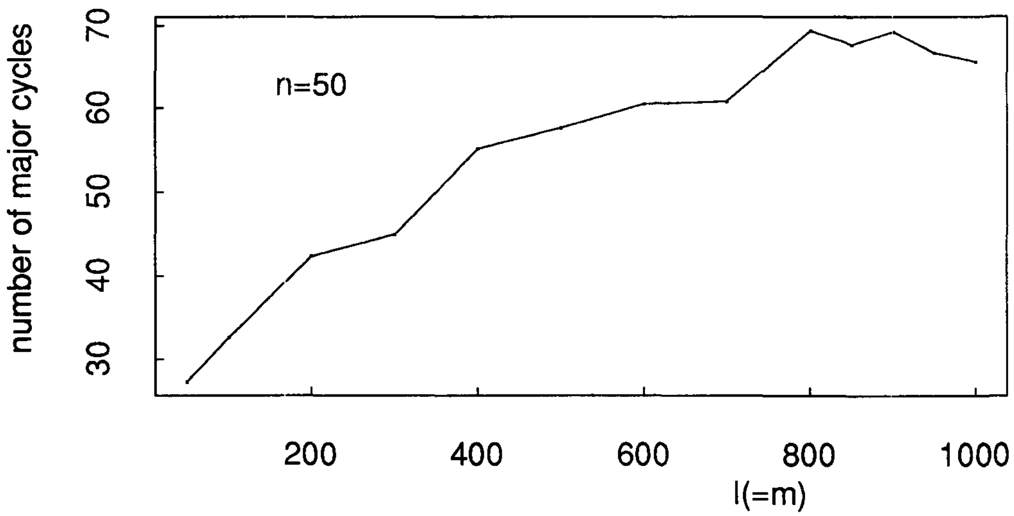

Figure 4.8.The number of the major cycles versus the number of the points for Type 1 (a ten-sample average).

References

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

[2] Fujishige, S.: *Submodular Functions and Optimization* (Annals of Discrete mathematics, Vol.47, North-Holland, Amsterdam, 1991).

[3] Fujishige, S. and Zhan, P.: A dual algorithm for finding the minimum-norm point in a polytope. *Journal of the Operations Research Society of Japan* Vol. 33 (1990), 188 ~ 195.

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

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

[6] Sekitani, K. and Yamamoto, Y.: A recursive algorithm for finding the minimum norm point in a polytope and a pair of closest points in two polytopes, *Mathematical Programming* (to appear).

[7] Stoer, J. and Witzgall, C.: *Convexity and Optimization in Finite Dimensions I* (Spring-Verlag, Berlin, 1970).

[8] Takehara, H.: An interior point algorithm for finding the minimum-norm point in a polytope. *MTEC Working Paper,* No. T911 (June 1991).

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

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