No.462

A Recursive Algorithm for Finding the Minimum Norm Point in a Polytope and a Pair of Closest Points in two Polytopes

K. Sekitani and Y. Yamamoto

June 1991

Institute of Socio-Economic Planning, University of Tsukuba
Tsukuba, Ibaraki 305, Japan
phone: 0298-53-5001/-5585, fax.: 0298-53-5070/-55-3849
e-mail: yamamoto@shako.sk.tsukuba.ac.jp%cunyvm.cuny.edu

A Recursive Algorithm for Finding the Minimum Norm Point in a Polytope and a Pair of Closest Points in two Polytopes [1]

K. Sekitani and Y. Yamamoto [2]

June 1991

Abstract For a given pair of finite point sets P and Q in some Euclidean space we consider two problems: Problem 1 of finding the minimum Euclidean norm point in the convex hull of P and Problem 2 of finding a minimum Euclidean distance pair of points in the convex hulls of P and Q. We propose a finite recursive algorithm for these problems. The algorithm is not based on the simplicial decomposition of convex sets and does not require to solve systems of linear equations.

Keywords minimum norm point, minimum distance pair of points, recursive algorithm, convex quadratic program

1 Introduction

Let P and Q be a given pair of sets of finite points of R^n. Let us denote the convex hull of P by $C(P)$. We consider the following two problems:

Problem 1 : Find the point of $C(P)$ which has the minimum Euclidean norm,
Problem 2 : Find a pair of points of $C(P)$ and $C(Q)$ which has the minimum Euclidean distance.

Note that the point of $C(P)$ which has the minimum Euclidean norm is unique. We denote it by $Nr(C(P))$. We denote the set of pairs of points of $C(P)$ and $C(Q)$ which have the minimum Euclidean distance by $Nr(C(P), C(Q))$. These problems have been considered by Wolfe [4]. He provided several basic results and also proposed an algorithm based on the simplicial decomposition of the convex hull

[1] Part of this research was done when the second author was supported by the Alexander von Humboldt-Foundation, Germany. This research is also partly supported by the Grant-in-Aid for Scientific Research of the Ministry of Education, Science and Culture.

[2] Institute of Socio-Economic Planning, University of Tsukuba, Tsukuba, Ibaraki 305, Japan
phone: 0298-53-5001/-5585, fax.: 0298-53-5070/-55-3849
e-mail: yamamoto@shako.sk.tsukuba.ac.jp%cunyvm.cuny.edu

$C(P)$. Fujishige and Zhan [3] proposed a dual algorithm for Problem 1 which is based on its dual formulation as well as the simplicial decomposition (see Wolfe [4] and also Freund [2]).

The following basic optimality condition is proved by Wolfe [4, Theorem 2.1].

Theorem 1.1 *$\hat{x} \in C(P)$ is $Nr(C(P))$ if and only if*

$$\| \hat{x} \|^2 \leq \hat{x}^t p \text{ for every } p \in P.$$

Concerning the relation between Problem 1 and 2 Canon and Cullum [1] showed that Problem 2 is reduced to Problem 1 with the set P replaced by $P - Q = \{ p - q \mid p \in P,\ q \in Q \}$. Hence we see the following optimality condition for Problem 2.

Theorem 1.2 *$\hat{x}^P \in C(P)$ and $\hat{x}^Q \in C(Q)$ are a minimum Euclidean distance pair if and only if*

$$\| \hat{x}^P - \hat{x}^Q \|^2 \leq (\hat{x}^P - \hat{x}^Q)^t (p - q) \text{ for every } p \in P \text{ and every } q \in Q.$$

In Section 2 we propose a recursive algorithm for Problem 1 and show that it provides $Nr(C(P))$ after a finite number of iterations. In Section 3 we show some norm monotonicity property of iterates and how the accuracy of solutions to be obtained is affected by the computational error. In Section 4 an algorithm for Problem 2 is proposed based on the equivalence of $Nr(C(P), C(Q))$ to $Nr(C(P - Q))$.

2 Algorithm $\mathcal{N}_1$ for Finding $Nr(C(P))$

In this section we consider Problem 1 of finding $Nr(C(P))$ for a given finite point set P. The algorithm first chooses a point x_0 from the convex hull $C(P)$. A point of P with the minimum norm of all points of P is recommended. In the k^{th} iteration with x_{k-1} as the current point it generates a proper subset P_k of P being the set of points minimizing the linear function $x_{k-1}^t p$ over all $p \in P$. Then it calls itself with P_k as the input data and moves from x_{k-1} to the point y_k obtained as the output of this recursive call.

Algorithm $\mathcal{N}_1(P)$

Step 0 : Choose a point x_0 from $C(P)$ and $k := 1$.

Step 1 : $\alpha_k := \min\{ x_{k-1}^t p \mid p \in P \}$.
If $\| x_{k-1} \|^2 \leq \alpha_k$, then $\hat{x} := x_{k-1}$ and stop.

Step 2 : $P_k := \{ p \mid p \in P \text{ and } x_{k-1}^t p = \alpha_k \}$.
Call $\mathcal{N}_1(P_k)$ and $y_k := Nr(C(P_k))$.

Step 3 : $\beta_k := \min\{ y_k^t p \mid p \in P \setminus P_k \}$.
If $\| y_k \|^2 \leq \beta_k$, then $\hat{x} := y_k$ and stop.

Step 4 : $\lambda_k := \max\{ \lambda \mid \{(1-\lambda)x_{k-1} + \lambda y_k\}^t y_k \leq \{(1-\lambda)x_{k-1} + \lambda y_k\}^t p$ for every $p \in P \setminus P_k \}$.
$x_k := (1-\lambda_k)x_{k-1} + \lambda_k y_k$, $k := k+1$ and go to Step 1.

Lemma 2.1 $0 < \lambda_k < 1$ *for* $k = 1, 2, \ldots$.

Proof: Clearly λ_k is determined by

$$\lambda_k = \min\{ \frac{x_{k-1}^t (p - y_k)}{(y_k - x_{k-1})^t (y_k - p)} \mid p \in P \setminus P_k \text{ and } (y_k - x_{k-1})^t (y_k - p) > 0 \}.$$

Since $y_k \in C(P_k)$, we see that $x_{k-1}^t(p - y_k) = x_{k-1}^t p - x_{k-1}^t y_k > 0$ for every $p \in P \setminus P_k$. Therefore $\lambda_k > 0$. Since the algorithm did not stop at Step 3, there is a point $p' \in P \setminus P_k$ such that $\| y_k \|^2 > y_k^t p'$. This means that $\lambda_k < 1$. □

By the above lemma we see that each point x_{k-1} as well as y_k is contained in $C(P)$. Then by applying Theorem 1.1 we will see that the point $\hat{x}$ obtained either in Step 1 or in Step 3 is $Nr(C(P))$.

Lemma 2.2 $\hat{x} = Nr(C(P))$.

Proof: Note that the initial point x_0 is chosen from $C(P)$, $y_{k-1} = Nr(C(P_{k-1})) \in C(P_{k-1}) \subseteq C(P)$ and $0 < \lambda_{k-1} < 1$. Then we see $x_{k-1} \in C(P)$ by induction. Therefore x_{k-1} of Step 1 is $Nr(C(P))$ by Theorem 1.1. To prove that y_k of Step 3 is $Nr(C(P))$ we have only to point out that $\| y_k \|^2 \leq y_k^t p$ holds for every $p \in P_k$ because $y_k = Nr(C(P_k))$. □

We will see that the set P_k generated in Step 2 is a fringe of the original set P in the sense that $C(P_k)$ is a proper face of $C(P)$.

Lemma 2.3 *If* $\| x_{k-1} \|^2 > \alpha_k$, *then* P_k *is a proper subset of* P *and* $C(P_k)$ *is a proper face of* $C(P)$.

Proof: Since $\| x_{k-1} \|^2 > \alpha_k = \min\{ x_{k-1}^t p \mid p \in P \}$, there exists a point of P which does not lie on the affine hull of P_k. Then $\dim C(P_k) < \dim C(P)$. This proves the lemma. □

Lemma 2.4 *There exists a point $\bar{p} \in P_{k+1} \setminus P_k$ such that $y_k^t \bar{p} < \| y_k \|^2$.*

Proof: Let $\bar{p}$ be a point of $P \setminus P_k$ such that $(y_k - x_{k-1})^t (y_k - \bar{p}) > 0$ and

$$\lambda_k = \frac{x_{k-1}^t (\bar{p} - y_k)}{(y_k - x_{k-1})^t (y_k - \bar{p})}. \tag{2.1}$$

Then $\bar{p} \in P_{k+1}$. In fact for a point $p \in P \setminus P_k$ we have

$$\begin{aligned} \{(1-\lambda_k)x_{k-1} + \lambda_k y_k\}^t \bar{p} &= \{(1-\lambda_k)x_{k-1} + \lambda_k y_k\}^t y_k \\ &\leq \{(1-\lambda_k)x_{k-1} + \lambda_k y_k\}^t p \end{aligned}$$

by the definition of λ_k. For a point $p \in P_k$ we also see from the definition of y_k

$$\begin{aligned} \{(1-\lambda_k)x_{k-1} + \lambda_k y_k\}^t \bar{p} &= \{(1-\lambda_k)x_{k-1} + \lambda_k y_k\}^t y_k \\ &= (1-\lambda_k)x_{k-1}^t y_k + \lambda_k \| y_k \|^2 \\ &= (1-\lambda_k)x_{k-1}^t p + \lambda_k \| y_k \|^2 \\ &\leq (1-\lambda_k)x_{k-1}^t p + \lambda_k y_k^t p \\ &= \{(1-\lambda_k)x_{k-1} + \lambda_k y_k\}^t p. \end{aligned}$$

Since $\lambda_k < 1$ as shown in Lemma 2.1, we see from (2.1) that $x_{k-1}^t(\bar{p} - y_k) < (y_k - x_{k-1})^t (y_k - \bar{p})$, which is equivalent to $y_k^t \bar{p} < \| y_k \|^2$. □

Lemma 2.5 $y_k \in C(P_{k+1})$ *for* $k = 1, 2, \ldots$.

Proof: Since $y_k = Nr(C(P_k)) \in C(P_k) \subseteq C(P)$ and we have seen

$$\{(1-\lambda_k)x_{k-1} + \lambda_k y_k\}^t y_k = \min\{\, \{(1-\lambda_k)x_{k-1} + \lambda_k y_k\}^t p \mid p \in P \,\} \tag{2.2}$$

in the proof of Lemma 2.4, the lemma follows from the definition of P_{k+1}. □

The following lemma is the key to finite convergence of the algorithm.

Lemma 2.6 $\| y_{k+1} \| < \| y_k \|$ *for* $k = 1, 2, \ldots$.

Proof: For the point $\bar{p}$ of Lemma 2.4, we have $y_k^t(y_k - \bar{p}) / \| y_k - \bar{p} \|^2 > 0$. Choose a λ such that

$$0 < \lambda < \min\{ \frac{2y_k^t(y_k - \bar{p})}{\| y_k - \bar{p} \|^2}, 1 \},$$

and consider the point $z = (1-\lambda)y_k + \lambda \bar{p}$. Then $z \in C(P_{k+1})$ by Lemma 2.4 and Lemma 2.5. Furthermore $\| z \|^2 = \| y_k \|^2 + \lambda\{2y_k^t(\bar{p} - y_k) + \lambda \| \bar{p} - y_k \|^2\} < \| y_k \|^2$. Therefore we obtain $\| y_{k+1} \| = \| Nr(C(P_{k+1})) \| \leq \| z \| < \| y_k \|$. □

Lemma 2.7 *When P consists of a single point, the algorithm $\mathcal{N}_1$ terminates within a finite number of iterations.*

Theorem 2.8 *When P consists of finitely many points, the algorithm $\mathcal{N}_1$ provides the minimum norm point $Nr(C(P))$ within a finite number of iterations.*

Proof: To prove the theorem by induction we assume that $\mathcal{N}_1(P')$ is finite whenever P' has fewer points than P has. Then by Lemma 2.3 we see that each step of the algorithm $\mathcal{N}_1(P)$ is finite. Since $\| Nr(C(P_{k+1})) \| = \| y_{k+1} \| < \| y_k \| = \| Nr(C(P_k)) \|$, no P_k is generated more than once. Thus together with Lemma 2.7 we see that $\mathcal{N}_1(P)$ terminates within a finite number of iterations. □

3 Norm Monotonicity and Error Analysis

The monotonicity property of norm $\| y_k \|$ was crucial in proving the finite termination of the algorithm $\mathcal{N}_1$. We will see the same monotonicity for another iterates x_k. We denote the affine combination of x_{k-1} and y_k with coefficient λ by $z(\lambda)$, i.e.,

$$z(\lambda) = (1-\lambda)x_{k-1} + \lambda y_k.$$

Theorem 3.1 $\| x_k \| < \| x_{k-1} \|$ *for* $k = 1, 2, \ldots$.

Proof: Since $y_k \in C(P_k)$ and $\| x_{k-1} \|^2 > \alpha_k = x_{k-1}^t p$ for every $p \in P_k$, $x_{k-1}^t(x_{k-1} - y_k) > 0$ and $x_{k-1} \neq y_k$. We consider the point $z(\lambda)$ for $\lambda = x_{k-1}^t(x_{k-1} - y_k)/\| x_{k-1} - y_k \|^2$. Note that $\lambda > 0$ and

$$\begin{aligned} \| z(\lambda) \|^2 &= \| y_k - x_{k-1} \|^2 \{\lambda - \frac{x_{k-1}^t(x_{k-1} - y_k)}{\| x_{k-1} - y_k \|^2}\}^2 + \| x_{k-1} \|^2 - \frac{\{x_{k-1}^t(x_{k-1} - y_k)\}^2}{\| x_{k-1} - y_k \|^2} \\ &= \| x_{k-1} \|^2 - \lambda^2 \| x_{k-1} - y_k \|^2 \\ &< \| x_{k-1} \|^2. \end{aligned}$$

The equality (2.2) implies from $x_{k-1} \in C(P)$ that

$$\{(1-\lambda_k)x_{k-1} + \lambda_k y_k\}^t y_k \leq \{(1-\lambda_k)x_{k-1} + \lambda_k y_k\}^t x_{k-1}.$$

Therefore we see from Lemma 2.1 that $0 < \lambda_k \leq x_{k-1}^t(x_{k-1} - y_k)/\| x_{k-1} - y_k \|^2 = \lambda$ and $x_k = (1-\mu)x_{k-1} + \mu z(\lambda)$ for some $0 < \mu \leq 1$. Hence $\| x_k \| = \| (1-\mu)x_{k-1} + \mu z(\lambda) \| < \| x_{k-1} \|$. □

Computational errors in evaluating α_k, β_k and norms $\| x_{k-1} \|$and $\| y_k \|$ are unavoidable and some tolerance should be introduced to the termination criteria. We should know how the solution to be obtained is affected by introducing the tolerance. In Lemma 3.4 and Lemma 3.5 we will show the error of solutions to be obtained when the termination criterion, e.g., $\| x_{k-1} \|^2 \leq \alpha_k$, is relaxed to $\| x_{k-1} \|^2 \leq \alpha_k + \delta_1$. The following theorem is proved by Wolfe [4, Theorem 2.2].

Theorem 3.2 *If $0 \neq x \in C(P)$ and $\alpha = \min\{x^t p \mid p \in P\}$, then*

$$\frac{\alpha}{\| x \|} \leq \| Nr(C(P)) \| \leq \| x \|.$$

Lemma 3.3 *If $x \in C(P)$ and $\alpha = \min\{x^t p \mid p \in P\}$, then*

$$\| x - Nr(C(P)) \|^2 \leq \| x \|^2 - \alpha.$$

Proof: We see from Theorem 1.1 and the definition of α that

$$\begin{aligned} \| x - Nr(C(P)) \|^2 &= \| x \|^2 - 2x^t Nr(C(P)) + \| Nr(C(P)) \|^2 \\ &\leq \| x \|^2 - 2x^t Nr(C(P)) + Nr(C(P))^t x \\ &= \| x \|^2 - x^t Nr(C(P)) \\ &\leq \| x \|^2 - \alpha. \end{aligned}$$

□

The next result immediately follows from Lemma 3.3.

Lemma 3.4 *If $\alpha_k \leq \| x_{k-1} \|^2 \leq \alpha_k + \delta_1$ in Step 1, then*

$$\| x_{k-1} - Nr(C(P)) \|^2 \leq \| x_{k-1} \|^2 - \alpha_k \leq \delta_1.$$

If $\beta_k \leq \| y_k \|^2 \leq \beta_k + \delta_2$ in Step 3, then

$$\| y_k - Nr(C(P)) \|^2 \leq \| y_k \|^2 - \beta_k \leq \delta_2.$$

Lemma 3.5 *Assume $x_{k-1} \neq 0$ and $y_k \neq 0$. If $\alpha_k \leq \| x_{k-1} \|^2 \leq \alpha_k + \delta_1$, then*

$$0 \leq \| x_{k-1} \| - \| Nr(C(P)) \| \leq \min\{ \sqrt{\delta_1}, \sqrt{\alpha_k + \delta_1}, \frac{\delta_1}{\| x_{k-1} \|} \}.$$

Moreover if $\alpha_k > 0$ and $Nr(C(P)) \neq 0$, then

$$0 \leq \frac{\| x_{k-1} \| - \| Nr(C(P)) \|}{\| Nr(C(P)) \|} \leq \min\{ \frac{\| x_{k-1} \|}{\alpha_k} \sqrt{\delta_1}, \frac{\delta_1}{\alpha_k} \}.$$

If $\beta_k \leq \| y_k \|^2 \leq \beta_k + \delta_2$, then

$$0 \leq \| y_k \| - \| Nr(C(P)) \| \leq \min\{ \sqrt{\delta_2}, \sqrt{\beta_k + \delta_2}, \frac{\delta_2}{\| y_k \|} \}.$$

Moreover if $\beta_k > 0$ and $Nr(C(P)) \neq 0$, then

$$0 \leq \frac{\| y_k \| - \| Nr(C(P)) \|}{\| Nr(C(P)) \|} \leq \min\{ \frac{\| y_k \|}{\beta_k} \sqrt{\delta_2}, \frac{\delta_2}{\beta_k} \}.$$

Proof: From lemma 3.4 it holds that $\| x_{k-1} \| - \| Nr(C(P)) \| \leq \| x_{k-1} - Nr(C(P)) \| \leq \sqrt{\delta_1}$. We see from the assumption $\| x_{k-1} \|^2 \leq \alpha_k + \delta_1$ that

$$\| x_{k-1} \| - \| Nr(C(P)) \| \leq \sqrt{\alpha_k + \delta_1} - \| Nr(C(P)) \| \leq \sqrt{\alpha_k + \delta_1}.$$

Theorem 3.2 implies that

$$0 \leq \| x_{k-1} \| - \| Nr(C(P)) \| \leq \| x_{k-1} \| - \frac{\alpha_k}{\| x_{k-1} \|} = \frac{\| x_{k-1} \|^2 - \alpha_k}{\| x_{k-1} \|} \leq \frac{\delta_1}{\| x_{k-1} \|}.$$

Hence we obtain $0 \leq \| x_{k-1} \| - \| Nr(C(P)) \| \leq \min\{ \sqrt{\delta_1}, \sqrt{\alpha_k + \delta_1}, \frac{\delta_1}{\| x_{k-1} \|} \}$. Since $\alpha_k > 0$, we see that $\sqrt{\delta_1} < \sqrt{\alpha_k + \delta_1}$ and

$$\begin{aligned} 0 \leq \frac{\| x_{k-1} \| - \| Nr(C(P)) \|}{\| Nr(C(P)) \|} &\leq \min\{ \sqrt{\delta_1}, \frac{\delta_1}{\| x_{k-1} \|} \} / \| Nr(C(P)) \| \\ &\leq \frac{\| x_{k-1} \|}{\alpha_k} \min\{ \sqrt{\delta_1}, \frac{\delta_1}{\| x_{k-1} \|} \} \\ &= \min\{ \frac{\| x_{k-1} \|}{\alpha_k} \sqrt{\delta_1}, \frac{\delta_1}{\alpha_k} \}. \end{aligned}$$

The required inequalities with respect to y_k can be seen in the same way as x_{k-1}.□

The step size λ_k in Step 4 could be another measure to evaluate the error of solutions. Furthermore we will see that if the successive iterates x_{k-1} and x_k are close to each other, x_k is a good approximation of $Nr(C(P))$. Given a coefficient λ, we denote the set of the affine combinations of x_{k-1} and points $p \in P$ by

$$P(x_{k-1}, \lambda) = \{ p' \mid p' = (1 - \lambda) x_{k-1} + \lambda p, \ p \in P \}.$$

Lemma 3.6 *If $0 \leq \lambda \leq 1$ and $z(\lambda)^t y_k \leq z(\lambda)^t p$ for every $p \in P$, then*

$$z(\lambda) = Nr\left(C\left(P\left(x_{k-1}, \lambda\right)\right)\right).$$

Proof: Let p be an arbitrary point of P. Then

$$z(\lambda)^t \{(1 - \lambda) x_{k-1} + \lambda p\} - z(\lambda)^t \{(1 - \lambda) x_{k-1} + \lambda y_k\} = \lambda z(\lambda)^t (p - y_k) \geq 0,$$

by the assumption. This means that $\| z(\lambda) \|^2 \leq z(\lambda)^t p'$ for every $p' \in P(x_{k-1}, \lambda)$. Since $y_k \in C(P)$ and $0 \leq \lambda \leq 1$, $z(\lambda) = (1 - \lambda) x_{k-1} + \lambda y_k \in C\left(P\left(x_{k-1}, \lambda\right)\right)$. Hence it follows from Theorem 1.1 that $z(\lambda) = Nr\left(C\left(P\left(x_{k-1}, \lambda\right)\right)\right)$. □

Lemma 3.7 *Assume $0 \leq \lambda \leq 1$ and $z(\lambda)^t y_k \leq z(\lambda)^t p$ for every $p \in P$. Then*

$$z(\lambda)^t x \geq \| z(\lambda) \|^2 + (1-\lambda) z(\lambda)^t (x - x_{k-1})$$

for any point $x \in C(P)$.

Proof: Let x be an arbitrary point of $C(P)$ and let $x(\lambda) = (1-\lambda)x_{k-1} + \lambda x$. Then $x(\lambda) \in C(P(x_{k-1}, \lambda))$. We see from Lemma 3.6 that $\| z(\lambda) \|^2 \leq z(\lambda)^t x(\lambda)$ and hence

$$\begin{aligned} z(\lambda)^t x &= z(\lambda)^t \{x - x(\lambda) + x(\lambda)\} \\ &= z(\lambda)^t \{x - x(\lambda)\} + z(\lambda)^t x(\lambda) \\ &\geq (1-\lambda) z(\lambda)^t (x - x_{k-1}) + \| z(\lambda) \|^2. \end{aligned}$$

□

Theorem 3.8 $\| x_k - Nr(C(P)) \|^2 \leq 2(1-\lambda_k)(x_{k-1}^t x_k - \alpha_{k+1})$.

Proof: Since $x_k = z(\lambda_k)$, $x_k^t y_k \leq x_k^t p$ for every $p \in P$ by the definition of λ_k in Step 4. Take $Nr(C(P))$ as x of Lemma 3.7 and we see by the definitions of $Nr(C(P))$ and α_{k+1} that

$$\begin{aligned} \| x_k - Nr(C(P)) \|^2 &= \| x_k \|^2 - 2x_k^t Nr(C(P)) + \| Nr(C(P)) \|^2 \\ &\leq \| x_k \|^2 - 2\{\| x_k \|^2 + (1-\lambda_k) x_k^t (Nr(C(P)) - x_{k-1})\} \\ &\quad + \| Nr(C(P)) \|^2 \\ &\leq 2\| x_k \|^2 - 2\{\| x_k \|^2 + (1-\lambda_k) x_k^t (Nr(C(P)) - x_{k-1})\} \\ &= 2(1-\lambda_k) x_k^t (x_{k-1} - Nr(C(P))) \\ &\leq 2(1-\lambda_k)(x_k^t x_{k-1} - \alpha_{k+1}). \end{aligned}$$

□

4 Algorithm $\mathcal{N}_2$ for Finding $Nr(C(P), C(Q))$

We consider Problem 2 of finding a pair of $Nr(C(P), C(Q))$. As pointed out in [1] it is equivalent to Problem 1 for $P - Q$. Note that

$$\min\{ x^t r \mid r \in P - Q \} = \min\{ x^t p \mid p \in P \} - \max\{ x^t q \mid q \in Q \},$$

and each step below will be seen to be equivalent to each step of the algorithm $\mathcal{N}_1$.

Algorithm $\mathcal{N}_2(P,Q)$

Step 0 : Choose a point x_0^P from $C(P)$ and a point x_0^Q from $C(Q)$.
$x_0 := x_0^P - x_0^Q$, $k := 1$.

Step 1 : $\alpha_k^P := \min\{\, x_{k-1}^t p \mid p \in P\,\}$ and $\alpha_k^Q := \max\{\, x_{k-1}^t q \mid q \in Q\,\}$.
$\alpha_k := \alpha_k^P - \alpha_k^Q$.
If $\| x_{k-1} \|^2 \leq \alpha_k$, then $\hat{x}^P := x_{k-1}^P$, $\hat{x}^Q := x_{k-1}^Q$ and stop.

Step 2 : $P_k := \{\, p \mid p \in P$ and $x_{k-1}^t p = \alpha_k^P\,\}$ and $Q_k := \{\, q \mid q \in Q$ and $x_{k-1}^t q = \alpha_k^Q\,\}$.
Call $\mathcal{N}_2(P_k, Q_k)$ and let (y_k^P, y_k^Q) be a pair of $Nr(C(P_k), C(Q_k))$.
$y_k := y_k^P - y_k^Q$.

Step 3 : $\beta_k^P := \min\{\, y_k^t p \mid p \in P\,\}$ and $\beta_k^Q := \max\{\, y_k^t q \mid q \in Q\,\}$.
$\beta_k := \beta_k^P - \beta_k^Q$.
If $\| y_k \|^2 \leq \beta_k$, then $\hat{x}^P := y_k^P$, $\hat{x}^Q := y_k^Q$ and stop.

Step 4 : $\gamma_k^P(\lambda) := \max\{\, \{(1-\lambda)x_{k-1} + \lambda y_k\}^t (y_k^P - p) \mid p \in P\,\}$ and
$\gamma_k^Q(\lambda) := \min\{\, \{(1-\lambda)x_{k-1} + \lambda y_k\}^t (y_k^Q - q) \mid q \in Q\,\}$.
$\lambda_k := \max\{\, \lambda \mid \gamma_k^P(\lambda) \leq \gamma_k^Q(\lambda)\,\}$.
$x_k^P := (1-\lambda_k)x_{k-1}^P + \lambda_k y_k^P$ and $x_k^Q := (1-\lambda_k)x_{k-1}^Q + \lambda_k y_k^Q$.
$x_k := x_k^P - x_k^Q$, $k := k+1$ and go to Step 1.

Let r be a point of $(P-Q) \setminus (P_k - Q_k)$. Then $r = p - q$ for some $p \in P$ and $q \in Q$ such that either $p \notin P_k$ or $q \notin Q_k$. Therefore we cannot simplify Step 3 to $\beta_k^P := \min\{\, y_k^t p \mid p \in P \setminus P_k\,\}$ and $\beta_k^Q := \max\{\, y_k^t q \mid q \in Q \setminus Q_k\,\}$. The value λ_k determined in Step 4 is easily seen to be equal to $\max\{\, \lambda \mid \{(1-\lambda)x_{k-1} + \lambda y_k\}^t y_k \leq \{(1-\lambda)x_{k-1} + \lambda y_k\}^t (p-q)$ for every $p \in P$ and every $q \in Q\,\}$.

In exactly the same way as in Section 2 we obtain the following lemmas and finite convergence of $\mathcal{N}_2$. We omit the proof.

Lemma 4.1 $0 < \lambda_k < 1$ *for* $k = 1, 2, \ldots$.

Lemma 4.2 $(\hat{x}^P, \hat{x}^Q) \in Nr(C(P), C(Q))$.

Lemma 4.3 *If* $\| x_{k-1} \|^2 > \alpha_k$, *then either* P_k *is a proper subset of* P *and* $C(P_k)$ *is a proper face of* $C(P)$ *or* Q_k *is a proper subset of* Q *and* $C(Q_k)$ *is a proper face of* $C(Q)$.

Lemma 4.4 *There exists a pair of points $(\bar{p}, \bar{q})$ such that*

(i) $\bar{p} \notin P_k$ *or* $\bar{q} \notin Q_k$,

(ii) $\bar{p} \in P_{k+1}$ *and* $\bar{q} \in Q_{k+1}$,

(iii) $y_k^t(\bar{p} - \bar{q}) < \| y_k \|^2$.

Lemma 4.5 $y_k \in C(P_{k+1} - Q_{k+1})$ *for* $k = 1, 2, \ldots$.

Lemma 4.6 $\| y_{k+1} \| < \| y_k \|$ *for* $k = 1, 2, \ldots$.

Lemma 4.7 *When either P or Q consists of a single point, the algorithm $\mathcal{N}_2$ reduces to the algorithm $\mathcal{N}_1$ and hence terminates within a finite number of iterations.*

Theorem 4.8 *When both P and Q consist of finitely many points, the algorithm $\mathcal{N}_2$ provides a pair of points of $Nr(C(P), C(Q))$.*

5 Concluding Remark

When we are given a polytope $X = \{ x \mid Ax \geq b,\ x \geq 0 \}$ instead of the finite set P, we could apply the algorithm $\mathcal{N}_1$ to find the minimum Euclidean norm point in X. Determining α_k and β_k is reduced to a linear program on X. The step size λ_k in Step 4 is given by

$$\max\{ \lambda \mid \{(1-\lambda)x_{k-1} + \lambda y_k\}^t y_k \leq \min\{\{(1-\lambda)x_{k-1} + \lambda y_k\}^t x \mid x \in X\} \}.$$

By the duality theorem of linear program it is equivalent to

$$\max\{ \lambda \mid \{(1-\lambda)x_{k-1} + \lambda y_k\}^t y_k \leq \max\{b^t z \mid A^t z \leq (1-\lambda)x_{k-1} + \lambda y_k,\ z \geq 0\} \},$$

and hence λ_k is obtained by solving the linear program

$$\begin{array}{ll} \text{max.} & \lambda \\ \text{s.t.} & \{(1-\lambda)x_{k-1} + \lambda y_k\}^t y_k \leq b^t z \\ & A^t z \leq (1-\lambda)x_{k-1} + \lambda y_k \\ & z \geq 0. \end{array}$$

The set P_k in Step 2 must be replaced by the face of optimal solutions of

$$\begin{array}{ll} \text{min.} & x_{k-1}^t x \\ \text{s.t.} & Ax \geq b,\ x \geq 0. \end{array}$$

Though the face is identified theoretically by $\{ x \mid Ax \geq b,\ x \geq 0,\ x_{k-1}^t x = \alpha_k \}$, we should note that it is very vulnerable to the numerical error of α_k and it has one more constraint than X.

References

[1] M.D. Canon and C.D. Cullum, "The determination of optimum separating hyperplanes I. A finite step procedure" RC 2023, IBM Watson Research Center, Yorktown Heights, NY (February 1968).

[2] R.M. Freund, "Dual gauge programs, with applications to quadratic programming and the minimum-norm problem", *Mathematical Programming* 38 (1987) 47-67.

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

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