MINIMUM MAXIMAL FLOW PROBLEM
– AN OPTIMIZATION OVER THE EFFICIENT SET –

MAIKO SHIGENO, ICHIRO TAKAHASHI, AND YOSHITSUGU YAMAMOTO

Abstract. The network flow theory and algorithms have been developed on the assumption that each arc flow is controllable and we freely raise and reduce it. We however consider in this paper the situation where we are not able or allowed to reduce the given arc flow. Then we may end up with a maximal flow depending on the initial flow as well as the way of augmentation. Therefore the minimum of the flow values that are attained by maximal flows will play an important role to see how inefficiently the network can be utilized. We formulate this problem as an optimization over the efficient set of a multicriteria program, propose an algorithm, prove its finite convergence, and report on some computational experiments.

1. Introduction

Considering the maximum flow problem, we usually take it for granted that each arc flow is controllable, i.e., we freely increase and decrease it as long as the conservation equations and capacity constraints are kept satisfied. However, in the situation where we are not able or allowed to reduce the given arc flow, we may fail to reach a *maximum* flow and get stuck in an undesired *maximal* flow. With such restricted controllability, we may end up with different maximal flows depending on the initial flow as well as the way of augmentation. Therefore the minimum of the flow values that are attained by maximal flows will play a prominent role in evaluating how inefficiently the network can be utilized.

Let $(V, s, t, E, \partial^+, \partial^-, c)$ denote a network of node set V with two designated nodes source s and sink t, arc set E, incidence functions ∂^+ and ∂^-, and a nonnegative capacity c_h for each arc h, where $\partial^+ h$ is the node that arc h leaves and $\partial^- h$ is the node that arc h enters. A vector $x = (\dots, x_h, \dots)$ of $|E|$-dimension is said to be a *feasible flow* if it satisfies the conservation equations and capacity constraints:

$$\sum_{\partial^+ h = i} x_h = \sum_{\partial^- h = i} x_h \text{ for all node } i \in V \setminus \{s, t\} \tag{1.1}$$

$$0 \le x_h \le c_h \text{ for all } h \in E. \tag{1.2}$$

Date: March 13, 2001, Revised April 2, 2001.

Key words and phrases. maximal flow, multicriteria program, efficient set, nonconvex optimization.

The authors thank Prof.N.v. Thoai for his comments on an earlier version of this paper. The first and the third authors are supported by the Garnt-in-Aid for Scientific Research of the Ministry of Education, Culture, Sports, Science and Technology of Japan.

Defining the $|V \setminus \{s,t\}| \times |E|$ matrix $A = [a_{ih}]_{h \in E}^{i \in V \setminus \{s,t\}}$, called the incidence matrix, by

$$a_{ih} = \begin{cases} +1 & \text{if } \partial^+ h = i \\ -1 & \text{if } \partial^- h = i \\ 0 & \text{otherwise,} \end{cases} \tag{1.3}$$

the conservation equation is simply written as $Ax = 0$. A feasible flow x is said to be a *maximal flow* if there is no feasible flow x' such that $x' \geq x$ and $x' \neq x$. The flow value, denoted by $\phi(x)$, of feasible flow x is given by

$$\phi(x) = \sum_{\partial^+ h = s} x_h - \sum_{\partial^- h = s} x_h. \tag{1.4}$$

Then the above problem of finding the minimum flow value of maximal flows, which was first raised by Shi and Yamamoto [24], is written as:

$$(mmF) \qquad \left| \begin{array}{ll} \text{minimize} & \phi(x) \\ \text{subject to} & x \text{ is a maximal flow.} \end{array} \right.$$

Note that this problem encompasses the minimum maximal matching problem, which is known to be NP-hard, e.g. [12], and is closely related to the uncontrollable flow problem raised by Iri [16]. In Figure 1 is shown an example of Iri [17] which should contrast the minumum maximal flow with the maximum flow. The number attached to each arc denotes the arc capacity. The maximum flow value grows as the arc capacity c increases, while the minimum maximal flow value does not.

The purpose of this paper is to formulate Problem (mmF) as a linear optimization problem over the efficient set of a multicriteria program and to propose an algorithm. The algorithm is mainly based on the local and global optimization techniques and exploits the integrality property of network flows.

In the next section some known results on the multicriteria program and the linear optimization over the efficient set are presented. In Section 3 local and global optimization techniques are discussed. In Section 4, combining these techniques and exploiting the network structure, we propose an algorithm for Problem (mmF) and show its finite convergence. In Section 5 are reported some computational experiments. Finally, Section 6 contains some conclusions.

2. Preliminaries on Multicriteria Program

Throughout this paper R^k denotes the set of k-dimensional real column vectors,

$$R^k_+ = \{\, x \mid x \in R^k; x \geq 0 \,\} \text{ and } R^k_{++} = \{\, x \mid x \in R^k; x > 0 \,\}.$$

R_k denotes the set of k-dimensional real row vectors, and R_{k+} and R_{k++} are defined in the similar way. We use e and 1 to denote a row vector and a column vector of ones, respectively, and e_k to denote the kth unit row vector of an appropriate dimension.

Definition 2.1. Let C be a $p \times n$ matrix and X be a polyhedral set of R^n defined as $X = \{\, x \mid x \in R^n_+; Dx = b \,\}$, where D is an $m \times n$ matrix and $b \in R^m$. Then we call the vector maximization problem

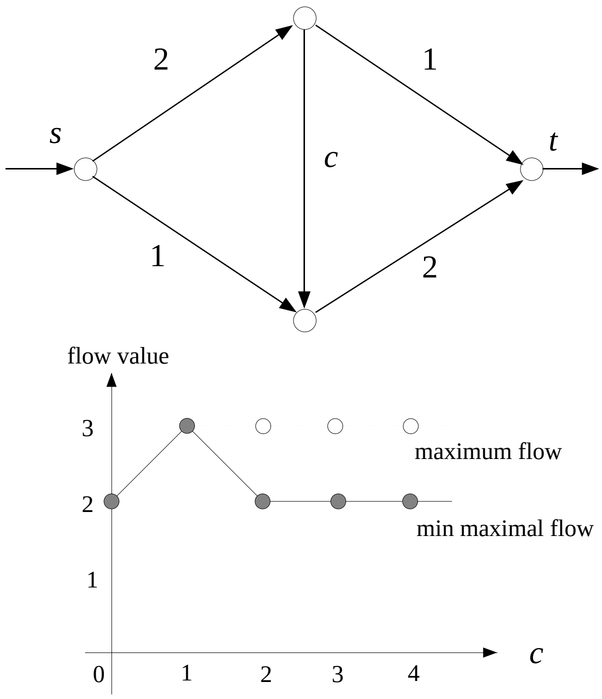

FIGURE 1. Maximum Flow vs. Minimum Maximal Flow

$$(MC) \qquad \left| \begin{array}{ll} \text{vector maximize} & Cx \\ \text{subject to} & x \in X \end{array} \right.$$

a *linear multicriteria program.* We assume that X is bounded and denote the set of its vertices (extreme points) by X_V. A point $x \in R^n$ is said to be an *efficient point* of Problem (MC) if $x \in X$ and there is no point $x' \in X$ such that

$$Cx' \geq Cx \text{ and } Cx' \neq Cx. \tag{2.1}$$

We denote the set of efficient points of (MC) by X_E.

The *linear optimization over the efficient set* is the following problem:

$$(P) \qquad \left| \begin{array}{ll} \text{minimize} & dx \\ \text{subject to} & x \in X_E, \end{array} \right.$$

where $d \in R_n$.

Let (MC) be defined for $C = I$, the indentity matrix of dimension $|E|$, and the set of feasible flows $X = \{\, x \mid x \in R^{|E|}; Ax = 0; 0 \leq x \leq c \,\}$, and let $dx = \phi(x)$. Then the minimum maximal flow problem (mmF) reduces to Problem (P). In this case Problem (MC) has the criteria as many as the arcs of the network, hence the algorithms, e.g. Benson [2, 3] and Thach, Konno and Yokota [26], that exploit the low dimensionality of p would not work efficiently. For the details of Problem (P) and the algorithms the readers should refer to An, Tao and Muu [1], Benson and Lee [5], Benson and Sayin [6], Dauer and Fosnaugh [9], Horst and Thoai [14], Muu [18], Sayin [22], Thoai [28, 29], White [31], Yamada, Tanino and Inuiguchi [32], and Yamamoto [33].

We introduce several well-known results about Problem (P), whose proofs can be found in, for example Benson [4], Sawaragi, Nakayama and Tanino [23], Steuer [25], and White [30]. We will outline some of the proofs to make this paper self-contained.

Theorem 2.2.

$$X_E = \left\{ x \;\middle|\; \begin{array}{l} x \in X;\ \textit{there is a } \lambda \in R_{p++} \textit{ such that} \\ \lambda C x \geq \lambda C x' \textit{ for all } x' \in X \end{array} \right\}. \tag{2.2}$$

Furthermore, there is an $M > 0$ such that R_{p++} above can be replaced by the $(p-1)$-dimensional simplex defined by

$$\Lambda = \{ \lambda \mid \lambda \in R_{p+}; \lambda \geq e; \lambda 1 = M \}. \tag{2.3}$$

Proof. For the proof of (2.2) see the above literatures. For the sake of further discussion we will outline the proof of the fact that Λ defined by (2.3) can replace R_{p++} in (2.2). By (2.2) X_E is the union of finitely many faces, say $F^1, \ldots, F^L$ of X such that F^ℓ is the optimum set of maximizing $\lambda_\ell C x$ over X for some $\lambda_\ell \in R_{p++}$. Let $\alpha_\ell = 1/(\min_{i=1,\ldots,p} \lambda_{\ell i})$ and $M = \max_{\ell=1\ldots,L} \alpha_\ell(\lambda_\ell 1)$, where 1 is the p-dimensional column vector of ones. Then for $\ell = 1, \ldots, L$ $(M/\lambda_\ell 1)\lambda_\ell$ lies in Λ defined by (2.3), and F^ℓ remains the optimum set of maximizing $(M/\lambda_\ell 1)\lambda_\ell C x$ over X. $\square$

As seen in the proof, the set X_E is a union of several faces of X. Furthermore we have the following theorem, for whose proof see Theorem 9.19 and Theorem 9.23 in Steuer [25], Theorem 3.31 in Sawaragi, Nakayama and Tanino [23], and Naccache [19].

Theorem 2.3. *The set X_E is a connected union of several faces of X. Any two vertices in X_E are connected by a path of efficient edges, where an efficient edge is an edge of X contained in X_E.*

This theorem implies the possibility of reaching any efficient vertex from any given efficient vertex by a series of pivot operations. This observation forms the foundation of the Adjacent Vertex Search Procedure, which will be explained in the next section.

Lemma 2.4. *Let $x = (x_B, x_N)$ be a basic feasible solution of X and let $D = [D_B, D_N]$ and $C = [C_B, C_N]$ be the partitions of D and C corresponding to the basic part x_B and the nonbasic part x_N of x, respectively. Let c^j and d^j be the columns of C_N and D_N, respectively, corresponding to a nonbasic variable x_j. The edge obtained by increasing x_j is an efficient edge if and only if $\lambda(C_N - C_B D_B^{-1} D_N) \leq 0$ and $\lambda(c^j - C_B D_B^{-1} d^j) = 0$ for some $\lambda \in \Lambda$. Furthermore the condition is equivalent to*

$$\max \{ \lambda(c^j - C_B D_B^{-1} d^j) \mid \lambda \in \Lambda; \lambda(C_N - C_B D_B^{-1} D_N) \leq 0 \} = 0. \tag{2.4}$$

Thus by solving the above linear programming we can find an efficient edge incident to the efficient vertex. We also see the following theorem about the location of an optimum solution of Problem (P).

Theorem 2.5. *There is an optimum solution of (P) in the vertex set X_V of X.*

Proof. As in the proof of Theorem 2.2, let $F^1, \ldots, F^L$ be the faces of X that constitute X_E. Then Problem (P) reduces to the family of problems

$$(P^\ell) \qquad \left|\begin{array}{ll} \text{minimize} & dx \\ \text{subject to} & x \in F^\ell, \end{array}\right.$$

whose optimum solution is located in the vertex set F^ℓ_V of F^ℓ due to the linearlity of dx. Since F^ℓ is a face of X, F^ℓ_V is contained in X_V. This completes the proof. □

Hence we have only to search in X_V for an optimum solution of (P), however the enumeration of X_V should be used only as a last resort for solving the problem.

3. Local and Global Optimization Techniques

In this section we will explain a local technique *Adjacent Vertex Search Procedure* and a global technique *Nonadjacent Vertex Search Procedure* for Problem (P).

The algorithms for the optimization over the efficient set proposed by Philip [20], Ecker and Song [10], Fülöp [11] and Bolintineanu [7] are mainly based on the technique of moving from an efficient vertex to an efficient neighbor with a smaller objective function value via an efficient edge. As shown in Theorem 2.3, the efficient set X_E is connected, and all the efficient vertices are conneced by paths of efficient edges. Thus, starting from any given efficient vertex, we could reach an optimum solution of Problem (P) by a series of pivot operations in theory. However, we cannot decrease the objective function value monotonically along the path that we trace, i.e., we will be eventually caught by a non-optimum efficient vertex none of whose efficient neighbors have a smaller objective function value. We see that the efficient vertex is a local minimum point as in the following Lemma 3.1, which can be found in Bolintineanu [7].

Lemma 3.1. *Let $x \in X_V \cap X_E$ and suppose that no efficient vertices linked to x by an efficient edge have a smaller objective function value than x. Then x is a local minimum point for (P).*

For $x, x' \in X_V$ let $[x, x']$ denote the edge connecting x and x'. For $x \in X_V \cap X_E$ let

$$N_E(x) = \{\, x' \mid x' \in X_V \cap X_E; [x, x'] \subseteq X_E \,\}, \tag{3.1}$$

i.e., the set of efficient vertices linked to x by an efficient edge. The Adjacent Vertex Search Procedure, which will be abbreviated by AVS Procedure, goes as follows.

Adjacent Vertex Search (AVS) Procedure

⟨⟨Initialization⟩⟩

Find $x^0 \in X_V \cap X_E$. If $N_E(x^0) = \emptyset$, then x^0 is an optimum solution of (P). Otherwise, set $k = 0$ and go to Step k.

⟨⟨Step k⟩⟩

⟨$k1$⟩ If $\{\, x \mid x \in N_E(x^k); dx < dx^k \,\} \neq \emptyset$, choose x^{k+1} from this set, $k = k + 1$ and go to Step k.

$\langle k2 \rangle$ Otherwise, set $v = x^k$ and stop.

Note that the procedure generates a sequence of distinct efficient vertices $x^0, x^1, \dots, x^k$ with a decreasing objective funtion value, i.e., $dx^0 > dx^1 > \cdots > dx^k$.

As was seen in Lemma 3.1, the efficient vertex v obtained by the AVS Procedure is only a local minimum solution. We need to see if there is an efficient point whose objective function value is less than that of v, and to find one if any. Let

$$H = X \cap \{\, x \mid dx = dv \,\} \tag{3.2}$$

and let H_E be the set of efficient points of vector $\max\{\, Cx \mid x \in H \,\}$. Then from the relation $H \subseteq X$ we see

$$X_E \cap H \subseteq H_E. \tag{3.3}$$

Based on this observation the algorithms in the papers mentioned at the beginning of this section enumerate the vertices of H_E to find an efficient edge $[u, u']$ of X such that $\min\{du, du'\} < dv$. Since the dimension of H is usually less than that of X by only one, the enumeration is very costly and deteriorates the efficiency of the algorithms.

Now we explain the global technique, which was originated by Phong and Tuyen [21], of determining if there is an efficient point x with $dx \le \alpha$ for a given $\alpha \in R$, where the pair of functions σ and τ_α plays a crucial role.

Definition 3.2. For $\lambda \in R_{p++}$ and $\alpha \in R$ let

$$\sigma(\lambda) = \max\{\, \lambda C x \mid x \in X \,\} \tag{3.4}$$

$$\tau_\alpha(\lambda) = \max\{\, \lambda C x \mid x \in X; dx \le \alpha \,\}. \tag{3.5}$$

Lemma 3.3. *(i) $\sigma(\cdot)$ and $\tau_\alpha(\cdot)$ are piecewise linear positively homogeneous convex functions on R_{p++}.*

(ii) For $\lambda \in R_{p++}$

$$\sigma(\lambda) = \max\{\, \lambda C v \mid v \in X_E \cap X_V \,\} \tag{3.6}$$

$$\tau_\alpha(\lambda) = \max\{\, \lambda C v \mid v \textit{ is an efficient vertex of } X \cap \{\, x \mid dx \le \alpha \,\}\,\}. \tag{3.7}$$

(iii) $\tau_\alpha(\lambda) \le \sigma(\lambda)$ for any $\lambda \in R_{p++}$.

(iv) $\tau_\alpha(\lambda)$ is a nondecreasing function in $\alpha \in R$.

Proof. All statements are readily seen from the theory of linear programming. □

Phong and Tuyen [21] showed the following theorem, whose proof will be given to make this paper self-contained.

Theorem 3.4. *$X_E \cap \{\, x \mid dx \le \alpha \,\} \ne \emptyset$ if and only if $\sigma(\lambda) = \tau_\alpha(\lambda)$ for some $\lambda \in \Lambda$.*

Proof. Suppose $\bar{x} \in X_E \cap \{\, x \mid dx \le \alpha \,\}$, then $\sigma(\bar{\lambda}) = \bar{\lambda} C \bar{x}$ for some $\bar{\lambda} \in \Lambda$. Since $d\bar{x} \le \alpha$, $\bar{\lambda} C \bar{x} \le \tau_\alpha(\bar{\lambda})$, which is less than or equal to $\sigma(\bar{\lambda})$. Therefore $\sigma(\bar{\lambda}) = \tau_\alpha(\bar{\lambda})$.

Suppose $\sigma(\bar{\lambda}) = \tau_\alpha(\bar{\lambda})$ at $\bar{\lambda} \in \Lambda$ and let $\bar{x}$ be a point that attains $\max\{\, \bar{\lambda} C x \mid x \in X; dx \le \alpha \,\} = \tau_\alpha(\bar{\lambda})$. Then, since $\sigma(\bar{\lambda}) = \tau_\alpha(\bar{\lambda})$, $\bar{x}$ maximizes $\bar{\lambda} C x$ over X, meaning $\bar{x} \in X_E$. □

Note that the point $\bar{x}$ obtained as a solution of $\max\{\,\bar{\lambda}Cx \mid x \in X; dx \leq \alpha\,\}$ is in general not a vertex of X. However, the minimal face of X that contains $\bar{x}$ lies entirely in X_E and can be easily identified. Then minimizing dx over the face would yield an efficient vertex of X satisfying $dx \leq \alpha$. In this way, by the additional computation if necessary, we always find an efficient vetex of X when there is an efficient point satisfying $dx \leq \alpha$.

In the sequel we restrict σ and τ_α on Λ. Let $\operatorname{epi}\sigma$ denote the epigraph of $\sigma : \Lambda \to R$, i.e.,

$$\begin{aligned} &(3.8) \qquad \operatorname{epi}\sigma = \{\,(\lambda, \mu) \mid \lambda \in \Lambda; \mu \in R; \mu \geq \sigma(\lambda)\,\} \\ &(3.9) \qquad\qquad = \{\,(\lambda, \mu) \mid (\lambda, \mu) \in \Lambda \times R; \mu - \lambda C v \geq 0 \text{ for all } v \in X_V \cap X_E\,\}. \end{aligned}$$

Then by the piecewise linear convexity of σ and τ_α we have

Lemma 3.5. *$\sigma(\lambda) = \tau_\alpha(\lambda)$ for some $\lambda \in \Lambda$ if and only if there is a vertex (λ, μ) of $\operatorname{epi}\sigma$ such that $\mu = \tau_\alpha(\lambda)$.*

Proof. Since the "if" part is trivial, we show the "only if" part. Note first that the recession cone of $\operatorname{epi}\sigma$ is $\{0\} \times R_+$ due to the boundedness of Λ and hence any point (λ, μ) in $\operatorname{epi}\sigma$ is a convex combination of its vertices plus a vector $(0, \theta)$ for some $\theta \geq 0$. Let (λ_ℓ, μ_ℓ) for $\ell = 1, \ldots, L$ be vertices of $\operatorname{epi}\sigma$ and suppose

$$\mu_\ell > \tau_\alpha(\lambda_\ell)$$

holds for $\ell = 1, \ldots, L$. Let λ be an arbitrary point of Λ, then $(\lambda, \sigma(\lambda)) \in \operatorname{epi}\sigma$, and hence

$$\lambda = \sum_\ell \theta_\ell \lambda_\ell \quad \text{and} \quad \sigma(\lambda) = \sum_\ell \theta_\ell \mu_\ell + \theta$$

for some $\theta \geq 0$ and $\theta_\ell \geq 0$ with $\sum_\ell \theta_\ell = 1$. Then by the convexity of τ_α and the assumption we have

$$\sigma(\lambda) \geq \sum_\ell \theta_\ell \mu_\ell > \sum_\ell \theta_\ell \tau_\alpha(\lambda_\ell) \geq \tau_\alpha(\lambda).$$

This completes the proof. $\square$

Figure 2 shows σ and τ_α on Λ. Since Λ is a bounded set of points λ satisfying $\lambda 1 = M$, their positive homogeneity is not observed in this figure.

For a nonempty subset W of $X_V \cap X_E$ let

$$(3.10) \qquad \sigma_W(\lambda) = \max\{\,\lambda C v \mid v \in W\,\}.$$

for $\lambda \in \Lambda$. Then

$$(3.11) \qquad \sigma_W(\lambda) \leq \sigma(\lambda)$$

for any $\lambda \in \Lambda$ or

$$(3.12) \qquad \operatorname{epi}\sigma \subseteq \operatorname{epi}\sigma_W,$$

i.e., $\operatorname{epi}\sigma_W$ is a polyhedral outer approximation of $\operatorname{epi}\sigma$. We readily have the following corollary from Theorem 3.4 and the piecewise linearity of $\sigma_W(\lambda)$.

Corollary 3.6. *(i) If $\tau_\alpha(\lambda) < \sigma_W(\lambda)$ for all $\lambda \in \Lambda$, then $X_E \cap \{\,x \mid dx \leq \alpha\,\} = \emptyset$.*

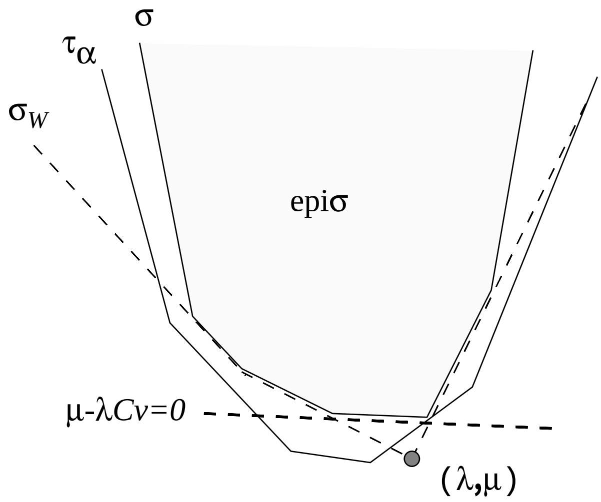

FIGURE 2. σ and τ_α

(ii) $\tau_\alpha(\lambda) \geq \sigma_W(\lambda)$ *for some* $\lambda \in \Lambda$ *if and only if there is a vertex* (λ, μ) *of epi* σ_W *such that* $\mu \leq \tau_\alpha(\lambda)$.

Proof. Since $\sigma_W(\lambda) \leq \sigma(\lambda)$ for every $\lambda \in \Lambda$, Theorem 3.4 yields (i).

The "if" part of (ii) is trivial and the "only if" part could be seen in an analogous way as in the proof of Lemma 3.5. We will however sketch the proof.

Since the recession cone of epi σ_W is $\{0\} \times R_+$, any point (λ, μ) in epi σ_W is a convex combination of its vertices plus a vector $(0, \theta)$ for some $\theta \geq 0$. Let (λ_ℓ, μ_ℓ) for $\ell = 1, \ldots, L$ be vertices of epi σ_W and suppose

$$\mu_\ell > \tau_\alpha(\lambda_\ell)$$

holds for $\ell = 1, \ldots, L$. Let λ be an arbitrary point of Λ, then $(\lambda, \sigma_W(\lambda)) \in \text{epi}\,\sigma_W$, and

$$\lambda = \sum_\ell \theta_\ell \lambda_\ell \quad \text{and} \quad \sigma_W(\lambda) = \sum_\ell \theta_\ell \mu_\ell + \theta$$

for some $\theta \geq 0$ and $\theta_\ell \geq 0$ with $\sum_\ell \theta_\ell = 1$. Then by the convexity of τ_α and the assumption we have

$$\sigma_W(\lambda) \geq \sum_\ell \theta_\ell \mu_\ell > \sum_\ell \theta_\ell \tau_\alpha(\lambda_\ell) \geq \tau_\alpha(\lambda).$$

This completes the proof. □

This corollary means that we can check whether $\tau_\alpha(\lambda) = \sigma_W(\lambda)$ at some $\lambda \in \Lambda$ by evaluating $\tau_\alpha(\lambda)$ at vertices (λ, μ) of epi σ_W. If $\tau_\alpha(\lambda) < \mu$ for every vertex (λ, μ), we conclude that $\tau_\alpha < \sigma_W$, and hence $X_E \cap \{ x \mid dx \leq \alpha \} = \emptyset$ by Corollary 3.6. Otherwise, i.e., we have found a vertex (λ, μ) with $\tau_\alpha(\lambda) \geq \mu$. Two possible cases occur. If $\sigma(\lambda) \leq \mu$, implying $\sigma(\lambda) = \mu = \tau_\alpha(\lambda)$, we see that $X_E \cap \{ x \mid dx \leq \alpha \} \neq \emptyset$ by Theorem 3.4. As shown in its proof and the statement following it, we will obtain a point of $X_V \cap X_E \cap \{ x \mid dx \leq \alpha \}$ by solving $\max\{ \lambda Cx \mid x \in X; dx \leq \alpha \}$

with additional computation if necessary. If $\sigma(\lambda) > \mu$, a vertex v of X that attains $\max\{\, \lambda Cx \mid x \in X \,\}$ is not in W. See Figure 2. Then W is augmented by this vertex v to make a better underestimation $\sigma_{W\cup\{v\}}$ of σ.

Nonadjacent Vertex Search (NVS) Procedure

$\langle\langle$Initialization$\rangle\rangle$

Set W_0 be a nonempty subset of $X_V \cap X_E$ and V_0 be the vertex set of $\operatorname{epi}\sigma_{W_0}$. Set $k = 0$ and go to Step k.

$\langle\langle$Step $k\rangle\rangle$

$\langle k1\rangle$ If $\tau_\alpha(\lambda) < \mu$ for all $(\lambda, \mu) \in V_k$, then stop. Otherwise, go to Step $k2$.

$\langle k2\rangle$ Choose $(\lambda_k, \mu_k) \in V_k$ such that $\tau_\alpha(\lambda_k) \geq \mu_k$ and evaluate $\sigma(\lambda_k)$.

$\langle k2.1\rangle$ If $\sigma(\lambda_k) \leq \mu_k$, then solve $\max\{\, \lambda_k Cx \mid x \in X; dx \leq \alpha \,\}$ obtaining $w \in X_V \cap X_E \cap \{\, x \mid dx \leq \alpha \,\}$ and stop.

$\langle k2.2\rangle$ Otherwise, solve $\max\{\, \lambda_k Cx \mid x \in X \,\}$ obtaining $v_k \in X_V \cap X_E$. Set $W_{k+1} = W_k \cup \{v_k\}$ and V_{k+1} be the vertex set of $\operatorname{epi}\sigma_{W_{k+1}}$. Set $k = k+1$ and go to Step k.

Theorem 3.7. *The above procedure terminates after a finite number of augmentations of W_k and either provides a point w of $X_V \cap X_E \cap \{\, x \mid dx \leq \alpha \,\}$ or shows that $X_E \cap \{\, x \mid dx \leq \alpha \,\}$ is empty.*

Proof. When the procedure stops at Step $k1$, we see that $\tau_\alpha < \sigma_{W_k} \leq \sigma$ and hence $X_E \cap \{\, x \mid dx \leq \alpha \,\}$ is empty.

When the procedure stops at Step $k2.1$, we have

$$\sigma(\lambda_k) \leq \mu_k \leq \tau_\alpha(\lambda_k),$$

implying $\sigma(\lambda_k) = \tau_\alpha(\lambda_k)$. Then w is an efficient vertex satisfying $dw \leq \alpha$. We show that v_k in Step $k2.2$ does not belong to W_k. Note that $(\lambda_k, \mu_k) \in V_k \subseteq \operatorname{epi}\sigma_{W_k}$ implies $\sigma_{W_k}(\lambda_k) \leq \mu_k$, and by the choice of v_k, $\lambda_k C v_k = \sigma(\lambda_k)$. Then $\lambda_k C v_k > \sigma_{W_k}(\lambda_k)$, which means that $v_k \notin W_k$. Therefore $W_0 \subset \cdots \subset W_k \subset W_{k+1}$, all of which are contained in the finite set $X_V \cap X_E$. This yields the finiteness of the procedure. □

Note that when a set of a single point, say v, is chosen as W_0, $\operatorname{epi}\sigma_{W_0}$ is simply written as

$$\operatorname{epi}\sigma_{W_0} = \{\, (\lambda, \mu) \mid \lambda \geq e; \lambda 1 = M; \mu - \lambda C v \geq 0 \,\}$$

and has p vertices, all of which are easily computed. The main technique used in the procedure is generating the vertex set of $\operatorname{epi}\sigma_{W_{k+1}}$ from that of $\operatorname{epi}\sigma_{W_k}$. Note first that $\operatorname{epi}\sigma_{W_k}$ is represented by finitely many linear inequalities each of which corresponds to a point of W_k:

$$\operatorname{epi}\sigma_{W_k} = \{\, (\lambda, \mu) \mid \lambda \geq e; \lambda 1 = M; \mu - \lambda C v \geq 0 \text{ for } v \in W_k \,\}. \tag{3.13}$$

Suppose that we have known the vertex set V_k of $\operatorname{epi}\sigma_{W_k}$, and we find a vertex v_k of X by maximizing $\lambda_k Cx$ over X in Step $k2.2$. This vertex will add an inequality $\mu - \lambda C v_k \geq 0$, which cuts off the vertex (λ_k, μ_k) of $\operatorname{epi}\sigma_{W_k}$. To generate the vertex set of $\operatorname{epi}\sigma_{W_{k+1}}$ we have only to generate the vertex set of

$(\operatorname{epi}\sigma_{W_k}) \cap \{ (\lambda, \mu) \mid \mu - \lambda C v_k = 0 \}$. There have been proposed a number of algorithms for this purpose, e.g., Horst, de Vries and Thoai [13], Chen, Hansen and Jaumard [8], and Thieu, Tam and Ban [27]. See also Section 4.2, Chapter II of Horst and Tuy [15].

4. Minimum Maximal Flow Problem

The minimum maximal flow problem (mmF) introduced in Section 1 is a linear optimization problem over the efficient set of (MC) with an $|E| \times |E|$ identity matrix as C and the set of feasible flows as X, i.e., $X = \{ x \mid x \in R^{|E|}; Ax = 0; 0 \le x \le c \}$. A maximal flow of (mmF) corresponds to an efficient point of (MC). We refer to a maximal flow that is a vetex of X as an *extreme maximal flow*. We assume hereafter that the capacity c_h is a nonnegative integer for every edge $h \in E$. By the network structure and the integrality of the capacities, we see that the objective function takes an integral value at each extreme maximal flow as well as an optimum solution of (mmF). Then we see

Lemma 4.1. *The AVS Procedure, when applied to Problem* (mmF)*, generates a sequence of extreme maximal flows with a decreasing integral objective function value.*

Let v be the extreme maximal flow obtained by the AVS Procedure. Then dv is an integer and there is a maximal flow x with $dx \le dv - 1$ if and only if v is not an optimum solution. Therefore the NVS Procedure with $\alpha = dv - 1$ determines if v is optimum, and if not, it finds an extreme maximal flow with an objective function value not greater than $dv - 1$.

Algorithm for (mmF)

⟨⟨Initialization⟩⟩

Find an extreme maximal flow w^0. If $N_E(w^0)$ is empty, stop with w^0 as an optimum solution. Otherwise, set $\nu = 1$ and go to Iteration ν.

⟨⟨Iteration ν⟩⟩

⟨ν1⟩ Apply the AVS Procedure to Problem (mmF) starting with $w^{\nu-1}$, and let v^ν be the extreme maximal flow obtained. Set $\alpha_\nu = dv^\nu - 1$ and go to Step ν2.

⟨ν2⟩ Apply the NVS Procedure for α_ν. If $X_E \cap \{ x \mid dx \le \alpha_\nu \}$ is empty, stop with v^ν as an optimum solution.

⟨ν3⟩ Otherwise, set w^ν be the extreme maximal flow found by the procedure such that $dw^\nu \le \alpha_\nu$, set $\nu = \nu + 1$ and go to Iteration ν.

Suppose that we have seen $\tau_{\alpha_\nu}(\lambda) < \mu$ at a vertex (λ, μ) of $\operatorname{epi}\sigma_{W_k}$. Since

$$\tau_{\alpha_{\nu+1}}(\lambda) \le \tau_{\alpha_\nu}(\lambda)$$

from (iv) of Lemma 3.3, this vertex can and should be eliminated from further consideration.

Theorem 4.2. *The above algorithm terminates within* dw^0 *of iterations.*

Proof. Clearly

$$dw^0 \ge dv^1 > \cdots \ge dv^\nu > dw^\nu \ge dv^{\nu+1} > \cdots , \tag{4.1}$$

that implies together with the integrality of the objective function value that

$$0 \le dw^{\nu} \le dw^{0} - \nu. \tag{4.2}$$

Therefore the algorithm iterates at most dw^0 times. □

As stated in Theorem 2.2, the set Λ could replace R_{p++} if a sufficiently large M is chosen. We will show that $|E|^2$ suffices as M. Now let $\bar{x} \in R^{|E|}$ be a given maximal flow and let $F = \{\, h \mid h \in E; \bar{x}_h = c_h \,\}$ and $\overline{F} = E \setminus F$. Note that $F \neq \emptyset$. We refer to a directed path from node i to node j as an $i - j$ path.

Lemma 4.3. *Let G be the graph of node set V and arc set $\overline{F}$.*

(i) G is acyclic and does not contain an $s - t$ path or a $t - s$ path.
(ii) For each node $i \in V \setminus \{s, t\}$ at least one of the following two cases occurs:
case 1: G has neither an $s - i$ path nor a $t - i$ path.
case 2: G has neither an $i - s$ path nor an $i - t$ path.

Proof. The assertion (i) is clear from the fact that $\bar{x}$ is a maximal flow. Let i be an arbitrary node and suppose that case 1 of (i) does not occur, i.e., there is either an $s - i$ path or a $t - i$ path. If there is an $s - i$ path, we have by (i) that there is neither an $i - s$ path nor an $i - t$ path, and if there is a $t - i$ path, we see that there is neither an $i - s$ path nor an $i - t$ path. These correspond to case 2. □

Now let a_ℓ denote the row of the incidence matrix A of the network defined by (1.3) corresponding to node $\ell \in V \setminus \{s, t\}$. Suppose we are given a nonempty subset U of $V \setminus \{s, t\}$ and let

$$\Delta_E^{+}(U) = \{\, h \mid h \in E; \partial^{+} h \in U; \partial^{-} h \in V \setminus U \,\} \tag{4.3}$$

$$\Delta_E^{-}(U) = \{\, h \mid h \in E; \partial^{-} h \in U; \partial^{+} h \in V \setminus U \,\}. \tag{4.4}$$

Then it will be readily seen from the definition of the incidence matrix that

$$\sum_{\ell \in U} a_\ell = \sum_{k \in \Delta_E^{+}(U)} e_k + \sum_{k \in \Delta_E^{-}(U)} (-e_k). \tag{4.5}$$

Lemma 4.4. *For each $h \in \overline{F}$ it holds that*

$$e_h = \alpha_h \sum_{\ell \in V_h} a_\ell + \sum_{k \in F} \beta_{hk} e_k - \sum_{k \in E \setminus \{h\}} \gamma_{hk} e_k \tag{4.6}$$

for some $\alpha_h \in \{-1, 1\}$, $V_h \subseteq V \setminus \{s, t\}$, $\beta_{hk} \in \{0, 1\}$ and $\gamma_{hk} \in \{0, 1\}$.

Proof. Let $i = \partial^{+} h$ and $j = \partial^{-} h$ and we consider the following two cases.
case 1: node i satisfies the condition of case 1 of Lemma 4.3.
Let

$$V_h^{+} = \{\, \ell \mid \ell \in V;\ \text{there is an } \ell - i \text{ path of } G \,\}. \tag{4.7}$$

Then we see from Lemma 4.3 that $s, t, j \notin V_h^{+}$ and that no arcs of $\overline{F}$ come into V_h^{+} from its complement $\overline{V_h^{+}} = V \setminus V_h^{+}$. Therefore the cut $(V_h^{+}, \overline{V_h^{+}})$ consists of the three sets of arcs: $\Delta_{\overline{F}}^{+}(V_h^{+})$, $\Delta_F^{+}(V_h^{+})$ and $\Delta_F^{-}(V_h^{+})$. By (4.5) we obtain

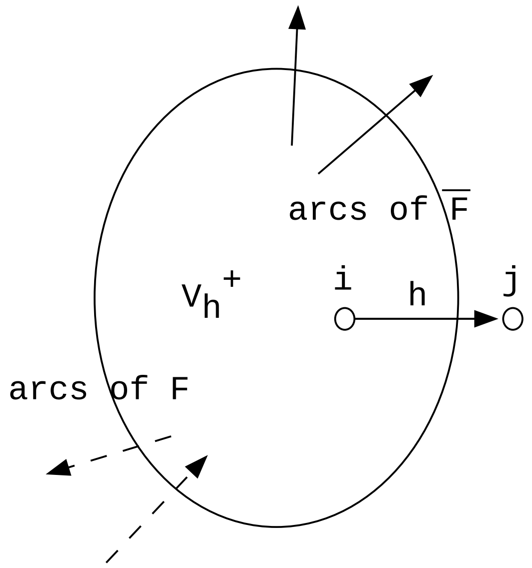

FIGURE 3. V_h^+ and arcs

$$\sum_{\ell \in V_h^+} a_\ell = \sum_{k \in \Delta^+_{\overline{F}}(V_h^+)} e_k + \sum_{k \in \Delta^+_F(V_h^+)} e_k + \sum_{k \in \Delta^-_F(V_h^+)} (-e_k), \tag{4.8}$$

which is rewritten as, since $h \in \Delta^+_{\overline{F}}(V_h^+)$,

$$\sum_{\ell \in V_h^+} a_\ell = e_h + \sum_{k \in \Delta^+_{\overline{F}}(V_h^+) \setminus \{h\}} e_k + \sum_{k \in \Delta^+_F(V_h^+)} e_k + \sum_{k \in \Delta^-_F(V_h^+)} (-e_k), \tag{4.9}$$

Thus we obtain

$$e_h = \sum_{\ell \in V_h^+} a_\ell + \sum_{k \in \Delta^-_F(V_h^+)} e_k - (\sum_{k \in \Delta^+_F(V_h^+)} e_k + \sum_{k \in \Delta^+_{\overline{F}}(V_h^+) \setminus \{h\}} e_k). \tag{4.10}$$

case 2: node i satisfies the condition of case 2 of Lemma 4.3.
Since node i satisfies the conditon of case 2 and arc $h = (i, j)$ is in $\overline{F}$, node j also satisfies the condition. Let $V_h^- = \{\, \ell \mid \ell \in V;$ there is a $\underline{j - \ell}$ path of $G\,\}$. Then we see $s, \underline{t}, i \notin V_h^-$ and that no arcs of $\overline{F}$ go from V_h^- into $\overline{V_h^-} = V \setminus V_h^-$, and the cut $(V_h^-, \overline{V_h^-})$ consists of $\Delta^-_{\overline{F}}(V_h^-)$, $\Delta^-_F(V_h^-)$ and $\Delta^+_F(V_h^-)$. Therefore

$$\sum_{\ell \in V_h^-} a_\ell = \sum_{k \in \Delta^-_{\overline{F}}(V_h^-)} (-e_k) + \sum_{k \in \Delta^-_F(V_h^-)} (-e_k) + \sum_{k \in \Delta^+_F(V_h^-)} e_k \tag{4.11}$$

$$= -e_h + \sum_{k \in \Delta^-_{\overline{F}}(V_h^-) \setminus \{h\}} (-e_k) + \sum_{k \in \Delta^-_F(V_h^-)} (-e_k) + \sum_{k \in \Delta^+_F(V_h^-)} e_k. \tag{4.12}$$

Hence

$$e_h = (-\sum_{\ell \in V_h^-} a_\ell) + \sum_{k \in \Delta_F^+(V_h^-)} e_k - (\sum_{k \in \Delta_F^-(V_h^-)} e_k + \sum_{k \in \Delta_{\overline{F}}^-(V_h^-)\setminus\{h\}} e_k). \tag{4.13}$$

This completes the proof. □

Theorem 4.5. *For a given maximal flow $\bar{x}$ there is an integral vector $\lambda \in R_{|E|}$ such that $1 \le \lambda_h \le |E|$ for each $h \in E$ and $\bar{x}$ maximizes λx over the set of feasible flows.*

Proof. By Lemma 4.4 we see for each $h \in \overline{F}$

$$e_h + \sum_{k \in E\setminus\{h\}} \gamma_{hk} e_k = \alpha_h \sum_{\ell \in V_h} a_\ell + \sum_{k \in F} \beta_{hk} e_k \tag{4.14}$$

for some $\alpha_h \in \{-1, 1\}$, $V_h \subseteq V \setminus \{s,t\}$, $\beta_{hk} \in \{0,1\}$ and $\gamma_{hk} \in \{0,1\}$. Adding these equations over $h \in \overline{F}$ and the identities $e_h = e_h$ for $h \in F$, we obtain

$$\sum_{k \in E} \lambda_k e_k = \sum_{\ell \in V\setminus\{s,t\}} \delta_\ell a_\ell + \sum_{k \in F} \zeta_k e_k, \tag{4.15}$$

where $\lambda_k = 1 + \sum_{h \in E\setminus\{k\}} \gamma_{hk}$ for $k \in E$, $\zeta_k = \sum_{h \in F} \beta_{hk}$ for $k \in F$, and δ_ℓ is appropriately defined for $\ell \in V \setminus \{s,t\}$. Note that

$$1 \le \lambda_k \le 1 + (|E| - 1) = |E| \tag{4.16}$$

for $k \in E$ and $\zeta_k \ge 0$ for $k \in F$. Let $\lambda = \sum_{k \in E} \lambda_k e_k$. Then for any feasible flow x it holds that

$$\lambda\bar{x} = \sum_{k \in E} \lambda_k e_k \bar{x} = \sum_{\ell \in V\setminus\{s,t\}} \delta_\ell a_\ell \bar{x} + \sum_{k \in F} \zeta_k e_k \bar{x} \tag{4.17}$$

$$= \sum_{k \in F} \zeta_k \bar{x}_k = \sum_{k \in F} \zeta_k c_k \tag{4.18}$$

$$\ge \sum_{k \in F} \zeta_k x_k = \sum_{\ell \in V\setminus\{s,t\}} \delta_\ell a_\ell x + \sum_{k \in F} \zeta_k e_k x = \lambda x, \tag{4.19}$$

meaning that the maximal flow $\bar{x}$ maximizes λx over the set of feasible flows. □

Corollary 4.6. $|E|^2$ *suffices for M defining Λ of* (2.3).

Proof. Let $\bar{x}$ be a maximal flow. By Theorem 4.5 it maximizes λx over the feasible flows for some $\lambda \in R_{|E|}$ such that $1 \le \lambda_h \le |E|$ for each $h \in E$. Let $\bar{\lambda} = (|E|^2 / \sum_{h \in E} \lambda_h)\lambda$. Then since $|E|^2 \ge \sum_{h \in E} \lambda_h$, $\bar{\lambda}$ lies in Λ defined for $M = |E|^2$ and $\bar{x}$ maximizes $\bar{\lambda} x$ over the feasible flows. □

5. Computational Experiment

Since problem (P) becomes easier to solve as the network becomes sparser, we fixed the number of nodes to $|V| = 16$ and varied the number of arcs $|E|$ from 20 to 76 in generating the problem instances. We generated ten instances for each number of arcs by randomly choosing arcs from $V \times V$ of possible locations, and also randomly choosing each arc capacity c_h from $\{1, 2, \ldots, 10\}$. The program was coded in Turbo Pascal and run on DELL Dimension XPS B600r. We employed the

Table 1. Running Time and Percentage of NVS Procedure

Number of arcs	Mean	Max.	Min.	NVS(%)
20	1.07	2.52	0.33	85
24	4.07	8.95	1.26	89
28	6.19	11.14	0.99	94
32	12.89	18.02	6.32	96
36	41.29	87.33	9.22	98
40	42.64	130.00	11.59	97
44	91.16	224.37	17.54	96
48	113.36	393.76	49.87	98
52	166.84	303.68	69.32	97
56	172.63	385.19	78.88	98
60	195.84	357.73	116.55	98
64	344.29	742.43	128.31	97
68	407.18	898.03	216.74	97
72	504.10	1876.04	233.76	97
76	623.54	2430.12	240.30	97

method proposed by Horst, de Vries and Thoai [13] to generate the vertex set of $\operatorname{epi}\sigma_{W^k}$.

Each row of Table 1 shows the mean, maximum and minimum of the running time in second, and the percentage of the time spent by the NVS Procedure in the total of the running time. We observe a high percentage of the time spent by the NVS Procedure, however, only one application of the AVS Procedure, followed by the NVS Procedure, provided global optimum solutions in most of the instances we solved, in fact 145 instances out of 150. The remaining five instances required the application of AVS and NVS Procedures only two times each. Note that at least one application of NVS Procedure is always needed to check the optimality of the current solution. This result together with the approximate polynomial in Figure 4

$$\text{Mean of the running time} \approx 0.023(|E| - 16)^{2.44}$$

expressing the mean running time in terms of the number of arcs should lead to the conclusion that the algorithm is quite efficient.

6. Conclusions

Combining the Adjacent Vertex Search Procedure and the Nonadjacent Vertex Search Procedure, we have proposed an algorithm for solving the minimum maximal flow problem. Owing to the network structure as well as the integrality of capacities, the algorithm yields a globally optimum solution within a finite number of iterations. However, we did not fully utilize the favorable properties of the network structure. In fact, no network algorithms are employed in AVS as well as NVS Procedures. Research on the application of efficient network algorithms should be carried out.

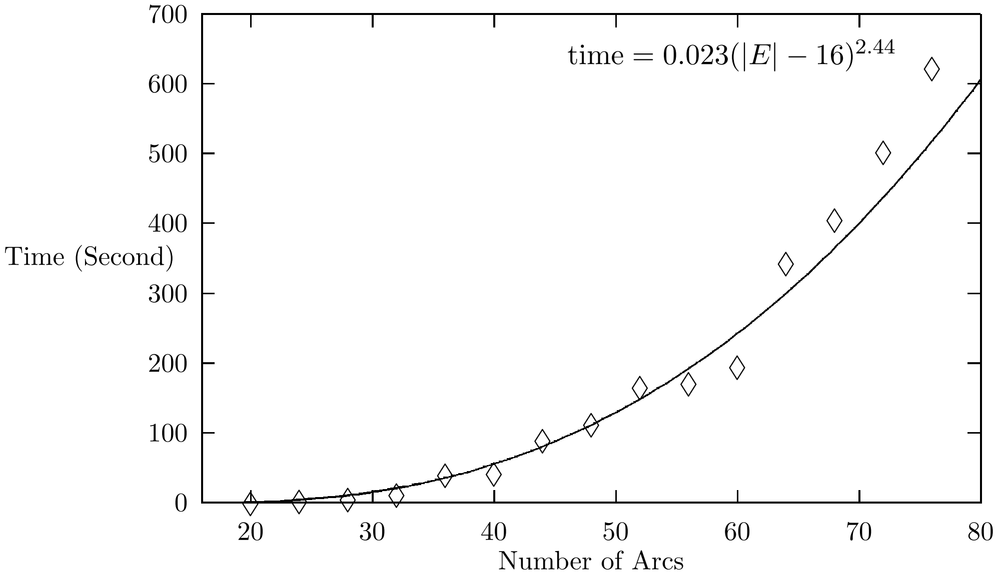

FIGURE 4. Mean of Running Time and Approximate Polynomial

REFERENCES

[1] L.T.H. An, P.D. Tao and L.D. Muu, "Numerical solution for optimization over the efficient set by d.c. optimization algorithms," *Operations Research Letters* **19** (1996) 117–128.

[2] H.P. Benson, "An all-linear programming relaxation algorithm for optimizing over the efficient set," *Journal of Global Optimization* **1** (1991) 83–104.

[3] H.P. Benson, "A finite nonadjacent extreme-point search algorithm for optimization over the efficient set," *Journal of Optimization Theory and Applications* **73** (1992) 47–64.

[4] H.P. Benson, "A geometric analysis of the efficient outcome set in multiple objective convex program with linear criteria functions," *Journal of Global Optimization* **6** (1995) 213–251.

[5] H.P. Benson and D. Lee, "Outcome-based algorithm for optimizing over the efficient set of a bicriteria linear programming," *Journal of Optimization Theory and Applications* **88** (1996) 77–105.

[6] H.P. Benson and S. Sayin, "Optimization over the efficient set: four special case," *Journal of Optimization Theory and Applications* **80** (1994) 3–18.

[7] S. Bolintineanu, "Minimization of a quasi-concave function over an efficient set," *Mathematical Programming* **61** (1993) 89–110.

[8] P.C. Chen, P. Hansen and B. Jaumard, "On-line and off-line vertex enumeration by adjacency lists," *Operations Research Letters* **10** (1991) 403–409.

[9] J.P. Dauer and T.A. Fosnaugh, "Optimization over the efficient set," *Journal of Global Optimization* **7** (1995) 261–277.

[10] J.G. Ecker and J.H. Song, "Optimizing a linear function over an efficient set," *Journal of Optimization Theory and Applications* **83** (1994) 541–563.

[11] J. Fülöp, "A cutting plane algorithm for linear optimization over the efficient set," in: S.Komlösi, T.Rapcsàk and S.Shaible eds., *Generalized Convexity*, Lecture Notes in Economics and Mathematical Systems 405, (Springer-Verlag, Berlin, 1994) pp.374–385.

[12] M.R. Garay and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, (Freeman, San Francisco, 1979).

[13] R. Horst, J. de Vries and N.V. Thoai, "On finding new vertices and redundant constraints in cutting plane algorithms for global optimization," *Operations Research Letters* **7** (1988) 85–90.

[14] R. Horst and N.V. Thoai, "Maximizing a concave function over the efficient or weakly-efficient set," *European Journal of Operational Research* **117** (1999) 239–252.
[15] R. Horst and H. Tuy, *Global Optimization: Deterministic Approach*, (Springer, Berlin, 1996).
[16] M. Iri, "An essay in the theory of uncontrollable flows and congestion", Technical Report, Department of Information and System Engineering, Faculty of Science and Engineering, Chuo University, TRISE 94-03 (1994).
[17] M. Iri, "Network flow -theory and applications with practical impact," J.Doležal and J.Fidler eds., *System Modelling and Optimization*, (Chapman & Hall, London, 1996) pp.24-36.
[18] L.D. Muu, "A convex-concave programming method for optimizing over the efficient set," *Acta Mathematica Vietnamica* **25** 1, 67–85.
[19] P.H. Naccache, "Connectedness of the set of nondominated outcomes in multicriteria optimization," *Journal of Optimization Theory and Applications* **25** (1978) 459–467.
[20] J. Philip, "Algorithms for the vector maximization problem," *Mathematical Programming* **2** (1972) 207–229.
[21] T.Q. Phong and J.Q. Tuyen, "Bisection search algorithm for optimizing over the efficient set," to appear in *Vietman Journal of Mathematics*.
[22] S. Sayin, "Optimizing over the efficient set using a top-down search of faces," *Operations Research* **48** (2000) 65–72.
[23] Y. Sawaragi, H. Nakayama and T. Tanino, *Theory of Multiobjective Optimization* (Academic Press, Orland, 1985).
[24] J.M. Shi and Y. Yamamoto, "A global optimization method for minimum maximal flow problem," *Acta Mathematica Vietnamica* **22** (1997) 271-287.
[25] R.E. Steuer, *Multiple Criteria Optimization: Theory, Computation and Application* (Wiley, New York, 1985).
[26] P.T. Thach, H. Konno and D. Yokota, "Dual approach to nminimization on the set of pareto-optimal solutions," *Journal of Optimization Theory and Applications* **88** (1996) 689–707.
[27] T.V. Thieu, B.T. Tam and V.T. Ban, "An outer-approximation method for globally minimizing a concave function over a compact convex set," *Acta Mathematica Vietnamica* **8** (1983) 21–40.
[28] N.V. Thoai, "A class of optimization problems over the efficient set of a multiple criteria nonlinear programming problem," *European Journal of Operational Research* **122** (2000) 58–68.
[29] N.V. Thoai, "Conical algorithm in global optimization for optimizing over efficient sets," *Journal of Global Optimization* **18** (2000) 321–336.
[30] D.J. White, *Optimality and Efficiency*, (John Wiley & Sons, Chichester, 1982).
[31] D.J. White, "The maximization of a function over the efficient set via a penalty function approach," *European Journal of Operational Research* **94** (1996) 143–153.
[32] S. Yamada, T. Tanino and M. Inuiguchi, "An inner approximation method for optimization over the weakly efficient set," *Journal of Global Optimization* **16** (2000) 197–217.
[33] Y. Yamamoto, "Optimization over the efficient set: Overview," to appear in *Journal of Global Optimization* .

Institute of Policy and Planning Sciences, Univeristy of Tsukuba, Tsukuba, Ibaraki 305-8573, Japan
E-mail address: `yamamoto@shako.sk.tsukuba.ac.jp; maiko@shako.sk.tsukuba.ac.jp`