ACTA MATHEMATICA VIETNAMICA
Volume 22, Number 1, 1997, pp. 271–287

A GLOBAL OPTIMIZATION METHOD FOR MINIMUM MAXIMAL FLOW PROBLEM

J. SHI AND Y. YAMAMOTO

Dedicated to Hoang Tuy on the occasion of his seventieth birthday

ABSTRACT. In this paper, we present two approaches for solving an $\mathcal{NP}$-hard problem: minimum maximal flow problem, i.e., $\min\{\|\xi\| \;\big|\xi$ is a maximal flow$\big\}$. We introduce lower bounds on flow, and cast the problem into a minimization of a concave function over a convex set. We solve the problem by a global optimization method. As an application, we consider the minimum maximal matching problem. Some numerical experiments are also reported.

1. INTRODUCTION

Throughout this paper, $G(V, E, \partial^+, \partial^-)$ is a *directed graph* consisting of *node set* $V = \{1, 2, \dots, n\}$, *edge set* $E \subseteq V \times V$ with $|E| = m$, and two *incidence functions* $\partial^+ : E \to V$ and $\partial^- : E \to V$. *Path, cycle,* etc., and their directed versions are defined as usual. A *network* $N(G, c)$ (abbreviated by N) is a graph G with a nonnegative valued *capacity function* $c : E \to R_+$ on the edge set E, where R_+ denotes the set of nonnegative real numbers. $c(e)$ is called the *capacity* of edge $e \in E$. A function $\xi : E \to R$ is called a *flow* in the network N if it satisfies the following conservation law:

$$\forall v \in V : \sum_{\partial^+ e = v} \xi(e) = \sum_{\partial^- e = v} \xi(e). \tag{1.1}$$

A flow ξ is said to be *feasible* if $0 \le \xi \le c$. A *two-terminal network* $N(G, s, t, c)$ is a network with two special nodes *source* s and *sink* t. An *s-t flow* ξ is a flow satisfying (1.1) for all nodes of $V \setminus \{s, t\}$. The *flow value* of an *s-t* flow ξ in defined by

Received December 2, 1996
Key words. Maximal flow, cutting plane, global optimization, bipartite graph, matching

$$\|\xi\| = \sum_{\partial^+ e = s} \xi(e) - \sum_{\partial^- e = s} \xi(e),$$

or

$$\|\xi\| = \sum_{\partial^- e = t} \xi(e) - \sum_{\partial^+ e = t} \xi(e).$$

A feasible s-t flow ξ is said to be *maximal* on network N if there does not exists a feasible s-t flow ξ' of N such that

$$\xi' \geq \xi, \quad \xi' \neq \xi.$$

We consider the problem P of finding the minimum value γ^* of maximal s-t flow for network N. That is

$$P: \quad \gamma^* = \min\{\|\xi\| \;\big|\, \xi \text{ is a maximal } s\text{-}t \text{ flow of } N\}.$$

For the sake of notational simplicity, we hereafter abbreviate s-t flow by flow.

In this paper, we show that Problem P is $\mathcal{NP}$-hard and present a global optimization method for the problem.

2. Lower bound method for the problem

To solve Problem P, we introduce a lower bound for flow ξ and consider the relationship between the corresponding maximum flow problem with the lower bound and the minimum maximal flow problem.

Let ℓ be a lower bound for flow ξ with $0 \leq \ell \leq c$. We abbreviable $N(G, s, t, c)$ with lower bound ℓ by $N(\ell)$. We say that a flow ξ of N is a *feasible flow* of $N(\ell)$ if $\ell \leq \xi \leq c$. A feasible flow ξ of $N(\ell)$ is said to be a *maximum flow* of $N(\ell)$ if

$$\|\xi\| = \max\{\|\xi'\| \;\big|\, \xi' \text{ is a feasible flow of } N(\ell)\}.$$

Then we have

Lemma 2.1. *If ξ is a maximal flow in N, then there exists a lower bound ℓ such that ξ is a maximum flow of $N(\ell)$.*

Proof. Let $\ell(e) = \xi(e)$ for every $e \in E$. Suppose that ξ is not a maximum flow of $N(\ell)$. Then there exists an augmenting path such that

1. $\xi(e) < c(e)$ for all forward edges e in the augmenting path,
2. $\xi(e) > \ell(e)$ for all backward edges e in the augmenting path.

Since $\ell(e) = \xi(e)$ for every $e \in E$, the augmenting path does not include any backward edge. Increasing the flow along the path, which consists of forward edges alone, we see that ξ is not a maximal flow of N. □

We assume in this paper the following assumption.

Assumption 2.2. Graph G is connected and acyclic.

Lemma 2.3. *Under Assumption 2.2, if ξ is a maximum flow of $N(\ell)$ for some ℓ, then ξ is a maximal flow of N.*

Proof. Let ξ be a maximum flow of $N(\ell)$ for some ℓ and suppose that ξ is not a maximal flow of N. Then there exists a feasible flow η N such that

$$\eta \geq \xi, \quad \eta \neq \xi.$$

Therefore $\eta(e_i) > \xi(e_i)$ holds for some edge $e_i \in E$. This indicates that there exist two edges e_{i-1} and e_{i+1} such that

$$\partial^- e_{i-1} = \partial^+ e_i,\ \eta(e_{i-1}) > \xi(e_{i-1}), \text{ and } \partial^+ e_{i+1} = \partial^- e_i,\ \eta(e_{i+1}) > \xi(e_{i+1}).$$

Repeating this procedure with e_i replaced by e_{i+1} and e_{i-1}, we will reach s and t after finitely many steps since G is connected and acyclic. Therefore we obtain a forward directed path from s to t such that

$$\xi(e_i) < \eta(e_i) \leq c(e_i) \text{ for } e_i \text{ in the forward directed path.}$$

It indicates that ξ is not a maximum flow of $N(\ell)$. □

Let

$$L = \{\ell | 0 \leq \ell \leq c, \text{ there exists a feasible flow of } N(\ell)\}, \tag{2.1}$$

and for $\ell \in L$ consider the problem

$$P(\ell): \quad \max\{\|\xi\| \ |\xi \text{ is a feasible flow of } N(\ell)\}.$$

We denote by $\gamma(\ell)$ the optimal value of Problem $P(\ell)$, i.e.,

$$\gamma(\ell) = \max\{\|\xi\| \ |\xi \text{ is a feasible flow of } N(\ell)\}.$$

When $N(\ell)$ admits no feasible flow, we assume $\gamma(\ell) = -\infty$.

Theorem 2.4. *The optimum value* γ^* *of* P *is equal to* $\min\{\gamma(\ell)|\ell \in L\}$, *i.e.,* $\gamma^* = \min\{\gamma(\ell)|\ell \in L\}$.

Proof. From Lemma 2.1, we have

$$\gamma^* \geq \min\{\gamma(\ell)|\ell \in L\}.$$

Lemma 2.3 claims that the converse inequality holds. □

By Theorem 2.4, Problem P is cast into the minimization of $\gamma(\ell)$ over L.

Lemma 2.5. $\gamma(\cdot)$ *is a piecewise linear concave function on* L.

Proof. If follows from the duality theorem of linear programming and the property of parametric linear programming (see, e,g., [2]). □

Lemma 2.6. *If* $0 \leq \ell^2 \leq \ell^1 \in L$, *then* $\ell^2 \in L$ *and* $\gamma(\ell^2) \geq \gamma(\ell^1)$.

Proof. Directly follows from the definitions of L and $\gamma(\ell)$. □

An element $\ell \in L$ is said to be a *maximal element* of L if there does not exist $\ell' \in L$ such that $\ell' \geq \ell$ and $\ell' \neq \ell$. Denote by $L_{\max}$ the set of maximal elements of L, that is,

$$L_{\max} = \{\ell \in L| \text{ there is no } \ell' \in L \text{ such that } \ell' \geq \ell,\ \ell' \neq \ell\}.$$

Denote by Ξ the set of feasible flow of N, that is

$$\Xi = \{\xi|\xi \text{ is a feasible flow of } N\}.$$

Also, we denote $\Xi_{\max}$ the set of maximal elements of Ξ. By Lemma 2.6 we have

Corollary 2.7. $\gamma(\cdot)$ *attains the minimum on* $L_{\max}$.

Let $R^m_- = \{\ell \in R^m|\ell \leq 0\}$ and $R^m_+ = \{\ell \in R^m|\ell \geq 0\}$. Then

Lemma 2.8. (i) $\Xi \subseteq L$, (ii) $L = \left(\Xi + R^m_-\right) \cap R^m_+$, (iii) $\Xi_{\max} = L_{\max}$.

Proof. (i) Trivial.

(ii) "$\subseteq$" Suppose $\ell \in L$. By the definition of L, there exists $\xi \in \Xi$ such that $0 \leq \ell \leq \xi$. Therefore $\ell = \xi + \xi_-$ for some $\xi_- \in R^m_-$. It means $\ell \in \left(\Xi + R^m_-\right) \cap R^m_+$.

"$\supseteq$" Suppose $\ell \in \left(\Xi + R^m_-\right) \cap R^m_+$. Then $\ell \geq 0$ can be written as $\ell = \xi + \xi_-$ for some $\xi \in \Xi$ and $\xi_- \in R^m_-$. Therefore $0 \leq \ell \leq \xi \in \Xi$. It implies that $\ell \in L$ by the definition of L.

(iii) "$\subseteq$" Let $\xi_{\max} \in \Xi_{\max}$ and suppose that $\xi_{\max} \notin L_{\max}$, i.e., there exists an $\ell \in L$ such that $\xi_{\max} \leq \ell$ and $\xi_{\max} \neq \ell$. Since $\ell \in L$, we see that $\ell \leq \xi$ for some $\xi \in \Xi$. It follows that $\xi_{\max} \leq \xi$ and $\xi_{\max} \neq \xi$. This is a contradiction.

"$\supseteq$" Let $\ell_{\max} \in L_{\max}$. Then we see that $\ell_{\max} \in \Xi$ from (ii). Suppose that $\ell_{\max} \notin \Xi_{\max}$, then there is a flow $\xi \in \Xi$ such that $\ell_{\max} \leq \xi$ and $L_{\max} \neq \xi$. Since $\Xi \subseteq L$, we have $\xi \in L$. This contradicts that $\ell_{\max} \in L_{\max}$. The desired result follows this relation and (i). □

By Corollary 2.7 and Lemma 2.8, we obtain

Theorem 2.9.

$$\begin{aligned}
&\min\{\gamma(\ell)|\ell \in L\} = \min\{\gamma(\xi)|\xi \in \Xi\} \\
&= \min\{\gamma(\xi)|\xi \in \Xi_{\max}\} = \min\{\gamma(\ell)|\ell \in L_{\max}\}.
\end{aligned}$$

Proof.

$$\begin{aligned}
\min\{\gamma(\ell)|\ell \in L\} &\leq \min\{\gamma(\xi)|\xi \in \Xi\} && (\text{by } \Xi \subseteq L) \\
&\leq \min\{\gamma(\xi)|\xi \in \Xi_{\max}\} && (\text{by } \Xi_{\max} \subseteq \Xi) \\
&= \min\{\gamma(\ell)|\ell \in L_{\max}\} && (\text{by } \Xi_{\max} = L_{\max}) \\
&= \min\{\gamma(\ell)|\ell \in L\}. && (\text{by Corollary 2.7})
\end{aligned}$$

□

Theorem 2.9 provides four possible domains over which we could consider Problem P.

2.1. Cutting plane algorithm over Ξ

Now we consider the minimization of $\gamma(\xi)$ over Ξ, i.e.,

$$Q: \quad \gamma^* = \min\{\gamma(\xi)|\xi \in \Xi\}.$$

Let A be the incidence matrix of graph G and let B be the matrix A with the rows corresponding to s and t deleted, that is,

$$A = \begin{pmatrix} \cdots \\ B \\ \cdots \end{pmatrix} . \begin{matrix} \leftarrow \text{ node } s \\ \leftarrow \text{ node } t \end{matrix}$$

Then Problem Q can be rewritten as

$$Q \quad \left| \begin{array}{l} \min \gamma(\xi) \\ \text{s.t. } B\xi = 0, \\ \qquad 0 \leq \xi \leq c. \end{array} \right.$$

Let us denote the rows of B by $b^1, \ldots, b^{n-2}$. We use the convention that $\gamma(\ell) = -\infty$ when $\ell \notin L$. Based on the above discussion we propose Algorithm I for Problem Q using the cutting plane method [7, 9]. Here we denote the set of vertices of a polyhedral set S by $W(S)$.

The outline of Algorithm I is as follows: At start, the algorithm constructs a hypercube $C_0 = \{\xi | 0 \leq \xi \leq c\}$, which contains Ξ. Without loss of generality, we can assume that $c \notin \Xi$ since otherwise c would be the optimal solution of P. Therefore c does not meet the conservation law at some node and we can choose $b \in \{b^1, \ldots, b^{n-2}\}$ such that $cb \neq 0$. Add the cutting plane $\{\xi | b\xi = 0\}$ to C_0 and make a smaller polytope C_1. In general step, we have a polytope C_{k-1} and its vertex set $W(C_{k-1})$. We evaluate $\gamma(\cdot)$ at the vertices and choose one with the smallest value. If it belongs to Ξ, we have solved the problem. Otherwise we add an equality constraint, say $b\xi = 0$, of $B\xi = 0$ which is violated by the chosen vertex, and set $C_k := C_{k-1} \cap \{\xi | b\xi = 0\}$. The efficient algorithm proposed by Horst et al. [6] could apply to generating the vertices $W(C_k)$ from $W(C_{k-1})$.

We state Algorithm I formally as follows:

Algorithm I

Step 0: Let $B_0 := \{b^1, \ldots, b^{n-2}\}$; $C_0 := \{\xi | 0 \leq \xi \leq c\}$;
Choose $b \in B_0$ such that $bc \neq 0$; $k := 1$;

Step 1: $B_k := B_{k-1} \setminus \{b\}$; $C_k := C_{k-1} \cap \{\xi | b\xi = 0\}$;
Find $W(C_k)$;

Step 2: $\gamma_k := \min\{\gamma(w) | w \in W(C_k)\}$; $\Gamma_k := \{w \in W(C_k) | \gamma(w) = \gamma_k\}$;
if $\Gamma_k \cap \Xi \neq \emptyset$
then $\overline{\gamma} := \gamma_k$; Let $\overline{\xi}$ be an arbitrary element of $\Gamma_k \cap \Xi$; Stop
else Choose $w \in \Gamma_k$ and $b \in B_k$ such that $bw \neq 0$;
$k := k + 1$; go to Step 1.

Theorem 2.10. *Algorithm I finds an optimal solution of Problem P within finitely many iterations.*

Proof. Note that polytope C_k contains Ξ and $\gamma(\cdot)$ is a concave function. Therefore in general

$$\begin{aligned}\gamma_k &= \min\{\gamma(w)|w \in W(C_k)\}\\ &= \min\{\gamma(w)|w \in C_k\} \leq \min\{\gamma(\xi)|\xi \in \Xi\} = \gamma^*.\end{aligned}$$

When Algorithm I terminates at Step 2, $\Gamma_k \cap \Xi \neq \emptyset$, and for any $w \in \Gamma_k \cap \Xi$, we have

$$\gamma_k = \gamma(w) \geq \min\{\gamma(\xi)|\xi \in \Xi\} = \gamma^*.$$

Therefore $\overline{\gamma}$ is the optimal value and $\overline{\xi}$ is an optimal solution of Problem Q. Since Ξ has only $n-2$ equality constraints, after at most $n-2$ iterations C_k attains Ξ and the algorithm terminates.

2.2. Outer approximation algorithm over L

Let $N'(\ell)$ be the network $N(\ell)$ with a directed edge (t, s) added such that $\ell(t, s) = 0$ and $c(t, s) = +\infty$. For a cut $(X, \overline{X})$ of $N'(\ell)$ we denote

$$c(X, \overline{X}) = \sum_{e \in (X, \overline{X})} c(e) \quad \text{and} \quad \ell(X, \overline{X}) = \sum_{e \in (\overline{X}, X)} \ell(e).$$

The following lemma provides a well-known necessary and sufficient condition for ℓ to be in L.

Lemma 2.11 (Theorem 6.11 in [1]). *A nonnegative ℓ is in L if and only if*

$$c(X, \overline{X}) \geq \ell(\overline{X}, X)$$

holds for every cut $(X, \overline{X})$ of $N'(\ell)$. That is

$$L = \{\ell | 0 \leq \ell \leq c,\ c(X, \overline{X}) \geq \ell(\overline{X}, X) \text{ for every cut } (X, \overline{X}) \text{ of } N'(\ell)\}.$$

Based on the above discussion, we can design Algorithm II to solve Problem P. Algorithm II starts with a hypercube $L_0 = \{\ell | 0 \leq \ell \leq c\}$ containing L. For $\ell_k \in L_k \setminus L$ $(k = 0, 1, \ldots)$, by Lemma 2.11, we find a cut $(X, \overline{X})$ of $N'(\ell)$ satisfying

$$c(X, \overline{X}) < \ell(\overline{X}, X). \tag{2.2}$$

Adding the cutting plane $\{\ell | c(X, \overline{X}) - \ell(\overline{X}, X) \geq 0\}$ to L_k yields the polytope L_{k+1}. Then we find all vertices $W(L_{k+1})$ of polytope L_{k+1} using the efficient method in [6]. Checking the values of $\gamma(\cdot)$ on $W(L_{k+1})$, we obtain the minimum value γ_{k+1} of $\gamma(\cdot)$ on L_{k+1}. If a vertex of $W(L_{k+1})$ with the minimum value of $\gamma(\cdot)$ belongs to L, then it is an optimal solution of Problem P. Otherwise $\ell_{k+1} \notin L$ will be found and the procedure is repeated. we assume that $c \notin L$ as before and then c is taken as ℓ_0.

Algorithm II

Step 0: Let $L_0 := \{\ell | 0 \le \ell \le c\}$; $\ell_0 := c$; $k := 0$;

Step 1: Construct a cut $(X, \overline{X})$ satisfying (2.2) for ℓ_k;

Let $L_{k+1} := L_k \cap \{\ell | c(X, \overline{X}) - \ell(\overline{X}, X) \ge 0\}$; Find $W(L_{k+1})$;

Step 2: $\gamma_{k+1} := \min\{\gamma(w) | w \in W(L_{k+1})\}$;

$W_{k+1} := \{w \in W(L_{k+1}) | \gamma(w) = \gamma_{k+1}\}$;

if $\gamma_{k+1} \ne -\infty$

then $\overline{\gamma} := \gamma_{k+1}$; Let $\overline{\ell}$ be an arbitrary element of $W_{k+1} \cap L$; Stop

else Choose ℓ_{k+1} from W_{k+1};

$k := k + 1$; goto Step 1.

Theorem 2.12. *Algorithm II finds an optimal solution of Problem P within finitely many iterations.*

Proof. It is easy to see that $\overline{\gamma} = \gamma^*$ and $\overline{\ell}$ is an optimal solution of Problem P when the algorithm terminates. From $\ell_k \in L_k$ and $\ell_k \notin L_{k+i}$ for $i \ge 1$, we see that the same cut $(X, \overline{X})$ will not appear more than once in Algorithm II. Note that there are only a finite number of cuts of $N'(\ell)$, then Algorithm II terminates in finitely many iterations. □

The cut satisfying (2.2) can be found in the following way (see e.g., [1]). A new network is constructed by

(i) adding two nodes S (super source) and T (super sink) to G;

(ii) linking S to v and v to T for every node v of G to make n edges (S, v) and n edges (v, T);

(iii) linking t to s to make new edge (t, s), and setting $\ell(t, s) = 0$, $c(t, s) = +\infty$.

For ℓ, the capacity c' of the new network is determined by

$$
\begin{aligned}
&0 \le c'(e) = c(e) - \ell(e) \text{ for all } e \in E,\\
&0 \le c'(S, v) = \sum_{\partial^- e = v} \ell(e) \text{ for all } v \in V,\\
&0 \le c'(v, T) = \sum_{\partial^+ e = v} \ell(e) \text{ for all } v \in V.
\end{aligned} \tag{2.3}
$$

We then determine a maximum flow of the new network. Meanwhile, we find a minimum cut $(X, \overline{X})$. If the value of the maximum flow is equal to $\sum_{v \in V} \ell(e)$, we see that $\ell \in L$. Otherwise, i.e., the value of the maximum flow is less that $\sum_{v \in V} \ell(e)$, then the cut $(X, \overline{X})$ satisfies (2.2).

3. Minimum maximal matching problem

In this section, we consider a special case of Problem P: Minimum maximal matching problem (see e.g., [3]). This problem can be stated as follows:

Instance: Graph $G = (V, E)$, positive integer $K \le |E|$.

Question: Is there a subset $M \subseteq E$ with $|M| \le K$ such that M is a maximal matching, i.e., no two edges in M share a common endpoint and every edge in $E \setminus M$ shares a common endpoint with some edge in M?

Even for a bipartite graph, this problem is $\mathcal{NP}$-complete (see, e.g., [3]). Throughout this section, we assume G is a bipartite graph. That is, node set V is partitioned into two subsets V_1 and V_2 such that for each edge $e \in E$ its two endpoints belong to the distinct set V_1 and V_2, respectively. To transform the minimum maximal matching problem to minimum maximal flow problem, we first make a directed version of the underlying graph G by designating all edges as pointing from the nodes in V_1 to the nodes in V_2. Then, we add a source node s and a sink node t, with edges connecting s to each node in V_1 and edges connecting each node in V_2 to t. Denote by $\overline{V}$ the node set enlarged with s and t, by $\overline{E}$ the enlarged edge set. For each edge in the network, we set the capacity c to 1. Denote the transformed network by $\overline{N} = ((\overline{V}, \overline{E}), 1)$. Note that in $\overline{N}$, every node in V_1 has one incoming edge and every node in V_2 has one outgoing edge. Therefore, a matching with cardinality K has a one-to-one correspondence to an integral flow of value K in $\overline{N}$.

Clearly, $\overline{N}$ satisfies Assumption 2.2. Now we focus on the relationship between maximal matching problem and maximal flow problem.

Lemma 3.1. *If matching M is maximal of G, then ξ defined by*

$$\xi(e) = \begin{cases} 1 & \text{if } e \in M, \\ 1 & \text{if } e = (s, \partial^+ f) \text{ for some } f \in M, \\ 1 & \text{if } e = (\partial^- f, t) \text{ for some } f \in M, \\ 0 & \text{otherwise,} \end{cases}$$

is a maximal flow of $\overline{N}$.

Proof. It is clear that ξ is a flow. Suppose ξ is not maximal. Then there exists a feasible flow η such that $\eta \geq \xi$, $\eta \neq \xi$. Let

$$\begin{aligned} E_1 &= \{e | e = (s, \partial^+ f) \text{ for some } f \in M\}, \\ E_2 &= \{e | e \in M\}, \\ E_3 &= \{e | e = (\partial^- f, t) \text{ for some } f \in M\}. \end{aligned}$$

From $\eta \geq \xi$, $\eta \neq \xi$, we see that exists an edge e^0 such that $\eta(e^0) > \xi(e^0)$. Suppose that $e^0 \in E_2$. Then there exist $e^1 \in E_1$ and $e^2 \in E_3$ such that $\eta(e^1) > \xi(e^1)$, $\eta(e^2) > \xi(e^2)$. Therefore $\xi(e^1) = \xi(e^0) = \xi(e^2) = 0$. Since (e^1, e^0, e^2) is a s-t path, $M' = M \cup \{e^0\}$ is still a matching of G. This is a contradiction. For the cases of $e^0 \in E_1$ and $e^0 \in E_3$, the proof will be the same as above. □

Then we have

Lemma 3.2. *If ξ is a maximal integral flow of $\overline{N}$, then $\{e | e \in E, \xi(e) = 1\}$ is a maximal matching of G.*

Proof. It is easy too see that $M = \{e | e \in E, \xi(e) = 1\}$ is a matching of G. Suppose that M is not maximal. Then there exists an edge $f \in E \setminus M$ such that $M \cup \{f\}$ is still a matching. Let

$$\eta(e) = \begin{cases} 1 & \text{if } e = f, \text{ or } e = (s, \partial^+ f), \text{ or } e = (\partial^- f, t), \\ \xi(e) & \text{otherwise.} \end{cases}$$

Then η is a feasible flow. This contradicts the maximality of ξ. □

From Lemmas 3.1 and 3.2, we see that minimum maximal matching problem is equivalent to the following problem:

$$R: \qquad \gamma^*_{\overline{N}} = \min\{\|\xi\| \; | \xi \text{ is a maximal integral flow of } \overline{N}\}.$$

Denote by $\overline{N}(\ell)$ the network $\overline{N}$ with lower bound $0 \leq \ell \leq 1$.

Lemma 3.3. *If ξ is a maximal integral flow of $\overline{N}$, then there exists a lower bound ℓ such that ξ is a maximum flow of $\overline{N}(\ell)$.*

Proof. Similar to the proof of Lemma 2.1. □

Lemma 3.4. *If ξ is a maximum integral flow of $\overline{N}(\ell)$ for some ℓ, then ξ is a maximal integral flow of $\overline{N}$.*

Proof. Note that $\overline{N}$ satisfies Assumption 2.2. The lemma follows from Lemma 2.3. □

Denote

$$L_Z = \{\ell |\ 0 \leq \ell \leq 1, \text{ there exists an integral feasible flow of } \overline{N}(\ell)\}.$$

Moreover, let

$$\gamma_Z(\ell) = \max\{\|\xi\| \mid \xi \text{ is an integral flow of } \overline{N}(\ell)\}.$$

Similar to Theorem 2.4, we have

Lemma 3.5.

$$\gamma^*_{\overline{N}} = \min\{\gamma_Z(\ell) |\ \ell \in L_Z\}.$$

Proof. From Lemmas 3.3 and 3.4. □

Let us denote the set of feasible flows of $\overline{N}$ by $\Xi_{\overline{N}}$ and

$$L_{\overline{N}} = \{\ell |\ 0 \leq \ell \leq 1, \text{ there exists a feasible flow } \xi \text{ of } \overline{N}(\ell)\}.$$

Lemma 3.6. *If $\ell \in L_Z$ is integral, then $\gamma_Z(\ell) = \gamma(\ell)$.*

Proof. By the definition of $\gamma_Z(\ell)$ and $\gamma(\ell)$, we see that $\gamma_Z(\ell) \leq \gamma(\ell)$. Since ℓ is integral, by Integrality Theorem (Theorem 6.5 in [1]) there is an integral solution ξ of $\gamma(\ell)$, i.e., $\gamma(\ell) = \|\xi\|$. This implies that $\gamma_Z(\ell) \geq \|\xi\| = \gamma(\ell)$. Therefore $\gamma_Z(\ell) = \gamma(\ell)$. □

Theorem 3.7. $\min\{\gamma_Z(\ell) |\ \ell \in L_Z\} = \min\{\gamma(\ell) | \ell \in L_{\overline{N}}\}$.

Proof. Let $\overline{\ell} = \arg\min\{\gamma_Z(\ell) | \ell \in L_Z\}$ and $\overline{\xi}$ be a solution of $\gamma_Z(\overline{\ell})$. Then $\overline{\xi}$ is integral and $\overline{\xi} \in L_Z$. From Lemma 3.6, $\gamma_Z(\overline{\xi}) = \gamma(\overline{\xi})$. From $\overline{\ell} \leq \overline{\xi}$ and the definition of $\gamma_Z(\cdot)$, we see that $\gamma_Z(\overline{\ell}) \geq \gamma_Z(\overline{\xi})$. Therefore

$$\min\{\gamma_Z(\ell) |\ \ell \in L_Z\} = \gamma_Z(\overline{\ell}) \geq \gamma_Z(\overline{\xi}) = \gamma(\overline{\xi}) \geq \min\{\gamma(\ell) |\ \ell \in L_{\overline{N}}\}.$$

Now we prove the converse inequality. From Theorem 2.9, we have

$$\min\{\gamma(\ell)|\ \ell \in L_{\overline{N}}\} = \min\{\gamma(\xi)|\xi \in \Xi_{\overline{N}}\}.$$

From Lemma 2.5, we see that $\gamma(\cdot)$ attains the minimum at a vertex of $\Xi_{\overline{N}}$. Note that every vertex of $\Xi_{\overline{N}}$ is integral. Then there exists an integral ℓ^* such that

$$\gamma(\ell^*) = \min\{\gamma(\ell)|\ \ell \in L_{\overline{N}}\}.$$

By Integrality Theorem in [1], $\gamma(\ell^*)$ has an integral solution. It means $\ell^* \in L_Z$. By Lemma 3.6, $\gamma(\ell^*) = \gamma_Z(\ell^*)$. Therefore

$$\min\{\gamma_Z(\ell)|\ell \in L_Z\} \le \gamma_Z(\ell^*) = \gamma(\ell^*) = \min\{\gamma(\ell)|\ \ell \in L_{\overline{N}}\}.$$

Thus, the converse inequality holds. □

By the above theorem, we can use Algorithms I and II to solve minimum maximal matching problem on a bipartite graph.

Theorem 3.8. *Problem P is $\mathcal{NP}$-hard.*

Proof. Note that minimum maximal matching problem is $\mathcal{NP}$-complete. Therefore Problem R is an $\mathcal{NP}$-hard problem. From Lemma 3.5 and Theorem 3.7, we see that $\min\{\gamma(\ell)|\ \ell \in L_{\overline{N}}\}$ is $\mathcal{NP}$-hard. Note that $\min\{\gamma(\ell)|\ \ell \in L_{\overline{N}}\}$ is a special case of $\min\{\gamma(\ell)|\ \ell \in L\}$. Hence $\min\{\gamma(\ell)|\ \ell \in L\}$ is also $\mathcal{NP}$-hard. It means that P is an $\mathcal{NP}$-hard problem by Theorem 2.4. □

4. Numerical experiments

In this section, we report some computational results for Algorithms I and II. The programs for the two algorithms were coded in Sun Pascal and were run on a Sun SPARCstation LX at School of Management, Science University of Tokyo. To make clear the behavior of the proposed algorithms, we consider three types of data for experiments:

Type 1: A s-t network with fixed number of nodes, $n = 8$, and varied number of edges, $m = 8, 9, 10, 11, 12, 13, 14$. The edges were chosen randomly with uniform distribution from the edge set $V \times V$. To focus on edge capacity's influence over running time, we created three kinds of capacities for each network:

Type 1.1: Every capacity is 1

Type 1.2: Every capacity is an integer generated randomly with uniform distribution from 1 to 10.

Type 1.3: Every capacity is a real number created randomly with uniform distribution between 1 to 10.

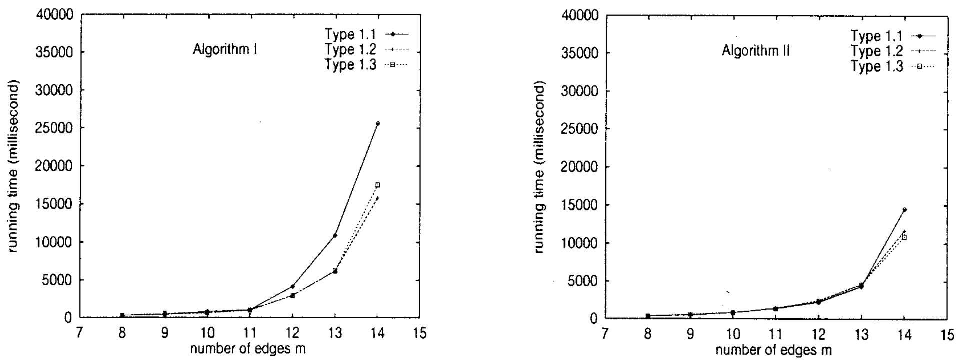

Figure 1. Average running time vs. number of edges m

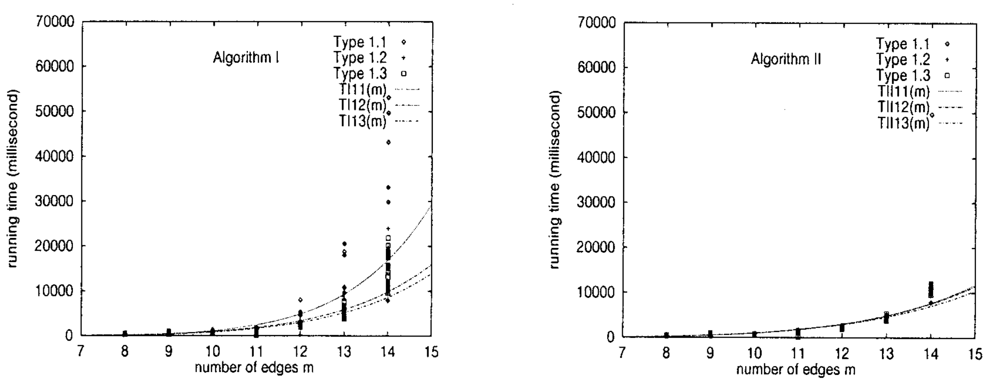

Figure 2. Distribution and approximate functions of the running time

Namely, Types 1.1, 1.2 and 1.3 have the same graphical structure but different capacities.

Type 2: A s-t network with fixed number of edge, $m = 12$, and varied number of nodes, $n = 6, 7, 8, 9, 10$. The 12 edges in this type were also created randomly with uniform distribution from the corresponding edge set $V \times V$. Every capacity is 1.

For demonstrating the efficiency of the algorithms for minimum maximal matching problem for a bipartite graph, we designed the following problem:

Table 1. Approximate functions of running time for Type 1

	Algorithm I	Algoritm II
Type 1.1	Tl11(m) $= 10^{-5}m^{8.0509}*$	Tll11(m) $= 7 \times 10^{-4}m^{6.1463}*$
Type 1.2	Tl12(m) $= 10^{-4}m^{6.9735}*$	Tll12(m) $= 12 \times 10^{-4}m^{5.936}*$
Type 1.3	Tl13(m) $= 10^{-4}m^{6.9242}*$	Tll13(m) $= 21 \times 10^{-4}m^{5.6944}*$

$*$ = millisecond

Type 3: A bipartite graph satisfying $|V_1| = |V_2|$. We varied the number of nodes of V_1 (or V_2) as 4, 5, 6, 7, 8. For each number of $|V_1|$, $[|V_1|^2 \times 0.3] + 1$ edges were generated randomly with uniform distribution from edge set $V_1 \times V_2$, where $[a]$ stands for the largest integer not greater than a.

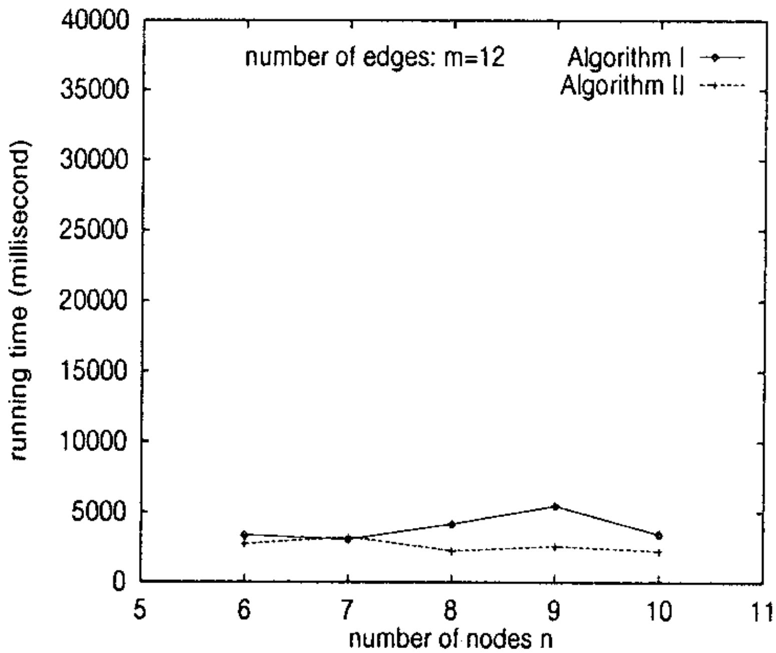

Figure 3. Average running time vs. number of nodes n for Type 2

All figures in the section are the average of ten instances except for Figure 5.

Figure 1 is the plots of average running time of Algorithms I and II vs. the number of edges m for Types 1.1. 1.2 and 1.3. The approximate functions of the running time are shown in Table 1. No certain conclusion should be drawn from the above limited experiments, however, we can see that Algorithm II is slightly better than Algorithm I; and different types of capacity do not influence on the running time sensitively and intensely.

Figure 2 illustrates the distribution of the running time of ten instances. The curves depict the approximate functions in Table 1.

Figure 3 indicates that the average running time of Type 2 changes gently as node number n grows from 6 to 10 for both Algorithms I and II when edge number is fixed to 12. Changing the capacity to integer or real number, we obtained the similar results.

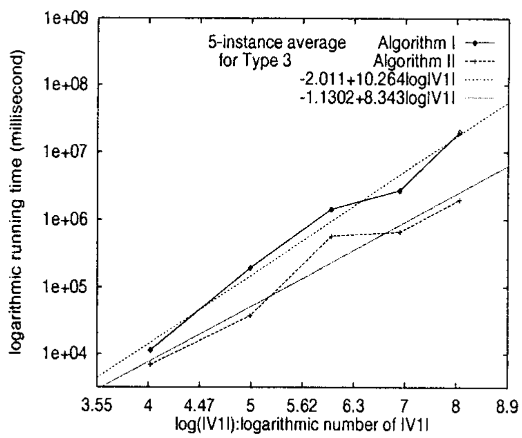

Figure 4. Tatal running time vs. running time till the appearance of an optimal solution for Type 1.1

We observed that the optimal solution appeared at almost the half of total running time. This phenomenon, illustrated in Figure 4 for Type 1.1, implies that is possible to obtain a rather good approximate solution even the programs are not executed to the end.

Figure 5 is the logarithmic plots of five-instance average running time of Algorithms I and II for Type 3. Note that $[|V_1|^2 \times 0.3] + 1$ edges were generated. The straight lines are approximate functions of running time with respect to $|V_1|$. Algorithm II again is better than Algorithm I with the approximate function $\log(\text{time}) = -1.1302 + 8.343 \log(|V_1|)$ versus $\log(\text{time}) = -2.011+10.264 \log(|V_1|)$, that is, $\text{time} = 7.4097\times 10^{-2}|V_1|^{8.343}$ versus $\text{time} = 9.7499 \times 10^{-3}|V_1|^{10.264}$.

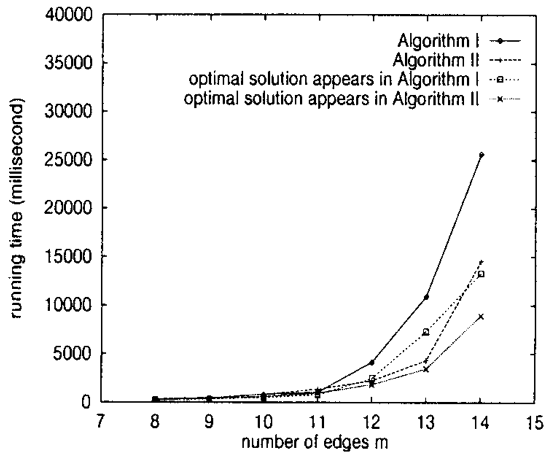

Figure 5. Average running time and approximate functions of running time for Type 3

5. Conclusion

In this paper, we set up a connection between the global optimization and an $\mathcal{NP}$-complete problem: minimum maximal flow problem. This connection can be exploited to solve the *u-flow (uncontrollable flow)* problem raised by Iri [4]:

> For an s-t network N with capacity c, an s-t flow ξ is called a u-flow if ξ is represented as a positive combination of elementary s-t paths. A u-flow ξ is said maximal if there does not exist a u-flow η in N such that $\eta \geq \xi$ and $c \geq \eta \neq \xi$. How to solve $\min\{\|\xi\| \mid \xi \text{ is a maximal u-flow of } N\}$?

This problem is called *minimum maximal u-flow* problem. As shown in [4, 8], problem

$$\min\{\|\xi\| \mid \xi \text{ is a maximal u-flow of } N\}$$

is $\mathcal{NP}$-hard. Under Assumption 2.2, one can see that

$$\begin{aligned} &\min\{\|\xi\| \mid \xi \text{ is a feasible maximal u-flow of } N\} \\ &= \min\{\|\xi\| \mid \xi \text{ is a feasible maximal flow of } N\}. \end{aligned}$$

It means that the approaches in this paper can be exploited to solve minimum maximal u-flow problem on an acyclic network. Due to $\mathcal{NP}$-hardness of the problems, it is difficult to estimate the order of complexity of the proposed algorithms. So as to investigate the efficiency, computational experiments of the proposed algorithms are carried out for small size problems. The experiments indicate that the problems can be solved in reasonable time if the underlying network is fairly small.

Acknowledgments

The authors wish to thank Professor Y. Dai of Kobe University of Commerce for her helpful comments on this paper.

References

1. R. K. Ahuja, T. L. Magnanti and J. B. Orlin, *Network Flows: Theory, Algorithms, and Apllications*, Prentice-Hall, London, 1993.
2. K. G. Murty, *Linear Complementary, Linear and Nonlinear Programming*, Sigma Series in Applied Mathematics **3**, Heldermann Verlag, Berlin 1988.
3. M. R. Garey and David S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, 1979.
4. 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).
5. K. R. Hoffman, *A method for globally minimizing concave function over convex sets*, Mathematical Programming **20** (1981), 22-32.
6. R. Horst, J. D. Vries and N. V. Thoai, *On finding new vertices and redundant constraints in cutting plane algorithms for global optimization*, Operations Research Letter **7** (1988), 85-90.
7. R. Horst and H. Tuy, *Global Optimization: Deterministic Approaches*, second edition, Springer-Verlag, 1993.
8. T. Matsui, *Is a given flow uncontrollable ?*, IEICE Trans. Fundamentals **E79**-A (1996), 448-451.
9. P. M. Pardalos and J. B. Rosen, *Constrained Global Optimization: Algorithms and Applications*, Lecture Note in Computer Science **268**, Springer-Verlag, 1987.

School of Management, Science University of Tokyo,
Kuki, Saitama 346, JAPAN

Institute of Policy and Planning Sciences, University of Tsukuba,
Tsukuba, Ibaraki 305, JAPAN