Journal of the Operations Research
Society of Japan
Vol. 29, No. 4, December 1986

A POLYNOMIAL ALGORITHM FOR THE MAXIMUM BALANCED FLOW PROBLEM WITH A CONSTANT BALANCING RATE FUNCTION

Akira Nakayama
Otaru University of Commerce

(Received August 8, 1986; Revised October 30, 1986)

Abstract M. Minoux considered the maximum balanced flow problem, which is a maximum flow problem with an additional constraint described in terms of a balancing rate function. In this paper, we propose an algorithm for the maximum balanced flow problem which is practically fast and simple. When the balancing rate function is constant, the proposed algorithm requires O(mT(n,m)) time, where T(n,m) is the time for the maximum flow computation for a network with n vertices and m arcs.

1. Introduction

The maximum balanced flow problem was introduced by M. Minoux [6]. A flow in a source-to-sink network is called balanced if the flow value of each arc does not exceed a fixed proportion (or balancing rate) of the total flow value from source s to sink t. Then the problem is to find a balanced flow such that the total flow value from s to t is maximized. The maximum balanced flow problem has an application in the telephone routing. This is, in fact, motivated Minoux's research in [6]. Consider a telephone network with its source and sink corresponding to two cities A and B, respectively. When a telephone line joining two adjacent spots breaks down, telephone routes through the broken line from A to B are blocked, but if the telephone routing considered as a flow from the source to the sink is balanced, then it is guaranteed that the number of the blocked routes is at most the fixed proportion of the total number of current routes from A to B. (Statistics shows that few telephone lines break at the same time.) If we have a maximum balanced flow in this network, a reliable telephone routing from A to B will be obtained.

Several algorithms [1], [4], [6] and [7] are proposed for the maximum balanced flow problem. W.-T. Cui [1] showed a variant of the dual simplex method without cycling on the underlying graph of a network, but his algorithm runs possibly in non-polynomial time. Let p_{max} be the maximum number of arc disjoint directed paths from source s to sink t in the underlying graph and $S(n,m)$ the complexity of the shortest path problem for a network with n vertices and m arcs and with a nonnegative arc length function. Then Minoux's algorithm [6] requires $O(p_{max}^2 \cdot S(n,m))$ computation time, where in his original formulation it is assumed that a given balancing rate function is constant. U. Zimmermann [7] has recently considered a generalization of the maximum balanced flow problem.

On the other hand, T. Ichimori, H. Ishii and T. Nishida [4] considered the weighted minimax flow problem, and S. Fujishige, A. Nakayama and W.-T. Cui [3] have recently pointed out the equivalence of the maximum balanced flow problem and the weighted minimax flow problem. The algorithm by Ichimori, Ishii and Nishida [4] has the complexity $O(T(n,m)^2)$ for general balancing rate functions, where $T(n,m)$ is the time for the maximum flow computation for a network with n vertices and m arcs. This complexity is the same as Zimmermann's [7].

In this paper, we consider a special case when the balancing rate function takes on a constant value r as Minoux assumes, and give an $O(\min(m, \lfloor 1/r \rfloor)T(n,m))$ algorithm for the maximum balanced flow problem where $\lfloor 1/r \rfloor$ is the maximum integer less than or equal to $1/r$. Our model is the specialization of those of [4] and [7], but our algorithm is faster than the algorithms in [4] and [7].

2. Maximum Balanced Flow Problem

Let $G=(V,A)$ be a directed graph where V is the vertex set and A is the arc set of G. For the set R_+ of nonnegative reals, we assume that an upper-capacity function $\bar{c}:A \to R_+$ and a lower-capacity function $\underline{c}:A \to R_+$ are given. When a (positive) balancing rate function $\alpha:A \to R_+ - \{0\}$ (we must have $\alpha>0$.) and a function β from A to the set R of reals are given, consider the two terminal network $N=(G=(V,A),\bar{c},\underline{c},\alpha,\beta,s,t)$ where s is the source and t is the sink of G. Then the maximum balanced flow problem (P) is formulated as follows:

$$\begin{array}{lll} & & \text{Maximize } f(a^*) \\ & & \text{subject to} \\ (1) & (\mathrm{P}) & D\, f = 0 \\ (2) & & \underline{c}(a) \le f(a) \le \bar{c}(a) \qquad (\ a \in A\ ) \\ (3) & & f(a) \le \alpha(a) f(a^*) + \beta(a) \qquad (\ a \in A\ ) \end{array}$$

Here $a^*=(t,s)\notin A$ is the arc added to the underlying graph G and D is the vertex-arc incidence matrix of G. If the function $f:A\cup\{a^*\}\to\mathrm{R}_+$ satisfies (1) and (2), f is called a feasible flow (or simply a flow) in network N. The flow f satisfying (3) is called balanced. Let f^* be the value of maximum $f(a^*)$, then f^* is called the maximum balanced flow value or simply the optimal value. Minoux [6] considered the case when for any arc $a\in A$, $\beta(a)=0$, $\underline{c}(a)=0$, $\alpha(a)=\mathrm{r}$ $(0<\mathrm{r}\le 1)$ and $\bar{c}(a)$ is a positive integer in network N.

In this paper, we assume that for each arc $a\in A$ $\alpha(a)=\mathrm{r}$ $(0<\mathrm{r}\le 1)$ as in [6], but that $\bar{c}(a)$, $\underline{c}(a)$ are nonnegative reals and $\beta(a)$ is real. Associated with the problem (P), consider the following problem (P*) for network $N^*=(G=(V,A),\bar{c},\underline{c},s,t)$:

$$\begin{array}{ll} & \text{Maximize } g(a^*) \\ (\mathrm{P}^*) & \text{subject to (1) and (2), where } f \text{ should be replaced by } g. \end{array}$$

And also consider the following problem (P_y) with a parameter y for network $N_y=(G=(V,A),\bar{c},\underline{c},\mathrm{r},\beta,y,s,t)$:

$$\begin{array}{lll} & & \text{Maximize } f(a^*) \\ & (\mathrm{P}_y) & \text{subject to constraints (1), (2) and} \\ (4) & & f(a) \le \mathrm{r}y + \beta(a) \qquad (\ a \in A\ ) \end{array}$$

Then for a sufficiently large y, Problem (P_y) coincides with Problem (P*). Let $f^{**}(y)$ (resp. g^*) be the value of maximum $f(a^*)$ in network N_y (resp. $g(a^*)$ in network N^*). For Problems (P) and (P_y) we have:

Proposition 1. $f^* = \max\{\ y\ :\ f^{**}(y)=y\ \}$. □

Concerning the property of the function $f^{**}(y)$ of y, we have:

Proposition 2. $f^{**}(y)$ is a monotone non-decreasing, continuous, piecewise linear and concave function. □

Any vertex partition $(S,\bar{S})$ with $s\in S$, $t\in\bar{S}$, $S\cup\bar{S}=V$ and $S\cap\bar{S}=\phi$ is called a cut. The capacity of a cut $(S,\bar{S})$ is defined as

$$c(S,\bar{S}) = \Sigma\{\bar{c}(a): a\in A^+(S)\} - \Sigma\{\underline{c}(a): a\in A^-(S)\}.$$

where $A^+(S)=\{a=(i,j)\in A: i\in S,\ j\in\bar{S}\}$ and $A^-(S)=\{a=(i,j)\in A: j\in S,\ i\in\bar{S}\}$.
A minimum cut is defined to be a cut having the minimum capacity. Then the

following theorem 3 is well known as max-flow min-cut theorem.

Theorem 3 [2]. For any network the maximum flow value from the source s to the sink t is equal to the capacity of a minimum cut. □

For network $N_{y'}$, the capacity of each arc $a \in A$ is determined as min $(\bar{c}(a), ry'+\beta(a))$ and let $(S,\bar{S})_{y'}$ be a minimum cut in $N_{y'}$. From Theorem 3, we have:

(5) $f^{**}(y')=\mathrm{r}|K'(y')|y'+\Sigma\{\beta(a):a\in K'(y')\}+\Sigma\{\bar{c}(a):a\in K''(y')\}-\Sigma\{\underline{c}(a):a\in A^{-}(S)\}$.

where $K'(y')=\{a\in A^{+}(S):\bar{c}(a)>\mathrm{r}y'+\beta(a)\}$ and $K''(y')=A^{+}(S)-K'(y')$.

Let $U(S,y')=\mathrm{r}|K'(y')|$ in (5). Then, we see that $U(S,y')$ is the multiple of the balancing rate r and that $|K'(y')|\le|A|$. $U(S,y')$ is called the slope of the minimum cut $(S,\bar{S})_{y'}$.

The function $f^{**}(y)$ is not differentiable in y, but for any $y>0$, we have the left differential coefficient $\sigma^{-}(y)$ defined as

$$\sigma^{-}(y) = \lim_{\Delta y\to 0-}\frac{f^{**}(y+\Delta y)-f^{**}(y)}{\Delta y} \tag{6}$$

Similarly, we define the right differential coefficient $\sigma^{+}(y)$, and call $\sigma^{-}(y)$, $\sigma^{+}(y)$ the slopes of the function $f^{**}(y)$ at y. Then there is the following relation between $\sigma^{-}(y)$ and the collection of the minimum cuts $(S,\bar{S})_{y}$ in network N_y. (See [6].)

Proposition 4. $\sigma^{-}(y) = \max\{U(S,y):(S,\bar{S})_y$ is a minimum cut in $N_y\}$. □

Without loss of generality, we assume $\bar{c}(a)>\beta(a)$ for each arc $a\in A$. Then we define $\underline{b}$ and $\bar{b}$ by

$\underline{b}=\max(\max\{(\underline{c}(a)-\beta(a))/\mathrm{r}:a\in A\},0)$, $\bar{b}=\max\{(\bar{c}(a)-\beta(a))/\mathrm{r}:a\in A\}$.

Note that the value $(\underline{c}(a)-\beta(a))/\mathrm{r}$ ($(\bar{c}(a)-\beta(a))/\mathrm{r}$) is derived from the equality case in the inequality $\underline{c}(a)\le \mathrm{r}y+\beta(a)$ ($\bar{c}(a)\ge \mathrm{r}y+\beta(a)$), respectively.

3. Algorithm for the Maximum Balanced Flow Problem

We can draw the graph of the two functions $z=f^{**}(y)$ and $z=y$ in a (y,z)-plane. We shall solve the maximum balanced flow problem based on Proposition 1. Let $Q(y)=(y,f^{**}(y))$ be the point on the graph of the function $z=f^{**}(y)$ at y. Then an outline of our algorithm is described as follows, though the detailed description will be given in subsequent sections: for a sufficiently large y^0, calculate the maximum flow value $f^{**}(y^0)$ and the slope $U(S^0,y^0)$

for some $S^0 \subset V$. If $y^0 = f^{**}(y^0)$, then we have the optimal value $f^* = y^0$ and stop. Otherwise, find the line L with slope $U(S^0, y^0)$ containing the point $Q(y^0)$. Then we obtain the intersection point (y^1, y^1) of the two lines L and $z=y$. Compute $f^{**}(y^1)$, and if $y^1 = f^{**}(y^1)$, then we have the optimal value $f^* = y^1$. Otherwise, we repeat the above procedure until we have $y^1 = f^{**}(y^1)$.

A point $Q(y'')$ is called the corner point of the function $z = f^{**}(y)$ if $f^{**}(y)$ is not differentiable at $y = y''$. A (line) segment $Ls(h', h'')$ on the graph $z = f^{**}(y)$ is $z = dy + b$ $(h' \le y \le h'')$, where the two points $Q(h')$ and $Q(h'')$ are the adjacent corners. The following proposition can be obtained from Theorem 3 and Proposition 4. It will be used to estimate the complexity of the proposed algorithms. Given the two slopes $U(S, y)$ and $\sigma^-(y)$ for any $y > 0$, we have:

Proposition 5. Let $Ls(h, h')$ and $Ls(h', h'')$ be the segments on the graph $z = f^{**}(y)$ such that $z = dy + b$ $(h \le y \le h')$ and $z = d'y + b'$ $(h' \le y \le h'')$.

(i) If $h < y < h'$, then we have $U(S, y) = \sigma^-(y)$.

(ii) At the corner point $Q(h')$ we have $d' \le U(S, h') \le d$. □

We show algorithms for finding the intersection point (y^*, y^*) of maximum y^* for three cases in the following sections 3.1∿3.3, respectively. The complexity of our algorithms will be considered in Section 3.3.

3.1. The Case When $\underline{c}(a)=0$ for any $a \in A$ and $\underline{b}=0$

In the present case, the zero flow ($f=0$) is feasible in network N_y for any $y \ge 0$. Hence from Proposition 2 there exists an optimal value y^* and we have the graphs $z = f^{**}(y)$ and $z = y$ as in Fig.1.

Now, we state the way of finding the optimal value $y^* \ge 0$ and show that the slope $U(S^i, y^i)$ for some $S^i \subset V$ at i-th repetition is a monotone increasing function with respect to i.

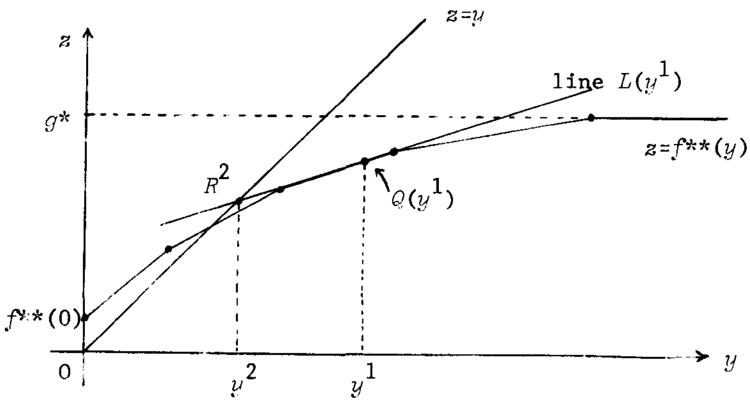

Fig.1

Suppose that $f^{**}(y^1)\le y^1$ for some y^1 and that we have not obtained the optimal value yet. (We may put $y^1=g^*$ at the initial step.) If $y^1=f^{**}(y^1)$, then we stop since y^1 is the optimal value. Otherwise, from Proposition 5 we have

(7) $\quad 1>U(S^1,y^1)=\sigma^-(y^1)$ if $Q(y^1)$ is not a corner point,

(8) $\quad 1>\sigma^-(y^1)\ge U(S^1,y^1)\ge\sigma^+(y^1)$ if $Q(y^1)$ is a corner point.

First, we consider the case when $Q(y^1)$ is not a corner point. The line $L(y^1)$ with slope $U(S^1,y^1)$ containing the point $Q(y^1)$ is given by

(9) $\quad L(y^1)\colon\ z-f^{**}(y^1)=U(S^1,y^1)(y-y^1)$

Note that the line $L(y^1)$ is the tangent line of $z=f^{**}(y)$. Then we can find the intersection point R^2 of the two lines $L(y^1)$ and $z=y$:

(10) $\quad R^2=(y^2,y^2)$,

where $y^2=(f^{**}(y^1)-\sigma^-(y^1)y^1)/(1-\sigma^-(y^1))$. After calculating $f^{**}(y^2)$ and $U(S^2,y^2)$, if $y^2=f^{**}(y^2)$, then we have the optimal value y^2. Otherwise (i.e. $y^2>f^{**}(y^2)$), we have

(11) $\quad 1>U(S^2,y^2)=\sigma^-(y^2)>\sigma^-(y^1)$ if $Q(y^2)$ is not a corner point,

(12) $\quad 1>U(S^2,y^2)\ge\sigma^+(y^2)>\sigma^-(y^1)$ if $Q(y^2)$ is a corner point.

Hence from (7), (11) and (12), we have $U(S^1,y^1)<U(S^2,y^2)$.

Next, we consider the case when $Q(y^1)$ is a corner point. Note that $\sigma^-(y^1)\ge U(S^1,y^1)\ge\sigma^+(y^1)$ from (8). Let $L'(y^1)$ be the line defined by

(13) $\quad L'(y^1)\colon\ z-f^{**}(y^1)=U(S^1,y^1)(y-y^1)$

and $R'^2=(y'^2,y'^2)$ be the intersection point of the two lines $L'(y^1)$ and $z=y$ where $y'^2=(f^{**}(y^1)-U(S^1,y^1)y^1)/(1-U(S^1,y^1))$. For $U(S^1,y^1)=\sigma^+(y^1)$ we have $U(S'^2,y'^2)>U(S^1,y^1)$. So assume that $U(S^1,y^1)>\sigma^+(y^1)$. If $y'^2>f^{**}(y'^2)$, then from $U(S'^2,y'^2)>\sigma^-(y^1)$ for some $S'^2\subset V$, we have $U(S'^2,y'^2)>U(S^1,y^1)$. Otherwise, we have the optimal value y'^2.

Each point $Q(y^i)$ obtained by the i-th repetition is always under the line $z=y$ and the slope $U(S^i,y^i)<1$ is a monotone increasing function of i. The above argument validates the following algorithm.

Algorithm 3.1:

Step 1: Calculate the maximum flow value g^* for network N^* and put $y=g^*$.

Step 2: (2.1): Find the maximum flow value $f^{**}(y)$ and the slope $U(S,y)$ in network N_y.

(2.2): If $y=f^{**}(y)$, then we stop since $y^*=y$ is the optimal value. Otherwise, put $y\leftarrow(f^{**}(y)-U(S,y)y)/(1-U(S,y))$ and go to Step 2.

3.2. The Case When $\underline{c}(a)=0$ for any $a\in A$ and $\underline{b}>0$

For any $y\geq\underline{b}$, the zero flow is feasible in network N_y. Comparing $f^{**}(\underline{b})$ with $\underline{b}$, there are two possible cases.

Case 1: $f^{**}(\underline{b})\geq\underline{b}$; It is easy to see that we have the optimal value $y^*>0$ and that we can apply Algorithm 3.1.

Case 2: $f^{**}(\underline{b})<\underline{b}$; Algorithm 3.1 works for the case when we have at least one intersection point of the two graphs $z=f^{**}(y)$ and $z=y$ as in Fig.2. Taking the case shown in Fig.3 into account, Algorithm 3.1 may be modified slightly. The following algorithm can be used in this section.

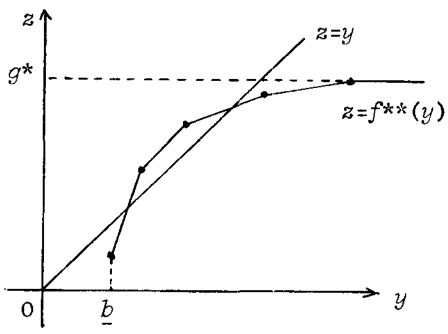

Fig.2

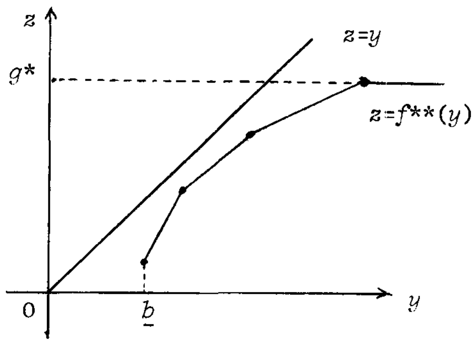

Fig.3

Algorithm 3.2:

Step 1: Find the maximum flow value g^* in network N^*. If $0\leq g^*<\underline{b}$, then stop and Problem (P) is infeasible. Otherwise, let $y=g^*$.

Step 2: Calculate the maximum flow value $f^{**}(y)$ and the slope $U(S,y)$ in network N_y. If $y<\underline{b}$ or $f^{**}(y)=0$ or $U(S,y)\geq 1$, then the algorithm terminates and Problem (P) is infeasible.

Step 3: If $y=f^{**}(y)$, then we have the optimal value $y^*=y>0$ and stop. Otherwise, put $y\leftarrow(f^{**}(y)-U(S,y)y)/(1-U(S,y))$ and go to Step 2.

3.3. General Case

The following proposition is easy to see:

Proposition 6. For the parametric problem (P_y) we have (a) and (b):

(a) If (P_y) has a flow for $y=\tilde{y}$, then for any $y\geq\tilde{y}$ (P_y) has a flow.

(b) If (P_y) has no flows for $y=\tilde{y}$, then for any $y\leq\tilde{y}$ (P_y) has no flows. □

Note that from Proposition 6, Problem (P) is infeasible if Problem (P*) is infeasible. In the following discussion, we assume that the problem (P*) has the maximum flow value $g^*>0$.

Now, we proceed our algorithm in the following way. If $f^{**}(y) \geq y$ for $y=\bar{b}$, then we have the optimal value $y^*=g^*$. Otherwise, find the intersection point of two lines $z=y$ and $z=g^*$. If for $y=g^*$, $y<\underline{b}$ or $y>\bar{b}$ or there are no feasible flows in network N_y, then the problem (P) is infeasible. Otherwise, apply either Algorithm 3.1 or 3.2. Note that when for some y'' there is a feasible flow in network $N_{y''}$ and we also have $\underline{b} \leq y'' \leq \bar{b}$, either Algorithm 3.1 or 3.2 can be used. Hence the general algorithm for the maximum balanced flow problem is as follows.

Algorithm 3.3.

Step 1: If Problem (P*) is infeasible, then the algorithm terminates since Problem (P) is infeasible.

Step 2: Find the maximum flow value g^* in network N^*. If $g^* \geq \bar{b}$, then we have the optimal value $y^*=g^*>0$ and stop.

Step 3: If $g^* \geq \underline{b}$, then let $y=g^*$. Otherwise, stop and the problem (P) is infeasible.

Step 4: (4.1): If Problem (P_y) is infeasible, then the algorithm terminates and the problem (P) is infeasible. Otherwise, compute the maximum flow value $f^{**}(y)$ and the slope $U(S,y)$ in network N_y.

(4.2): If $y<\underline{b}$ or $U(S,y) \geq 1$, then stop since the problem (P) is infeasible.

Step 5: If $y=f^{**}(y)$, then we have the optimal value $y^*=y$ and stop. Otherwise, put $y \leftarrow (f^{**}(y)-U(S,y)y)/(1-U(S,y))$ and go to Step 4.

In regard to the complexity of Algorithm 3.3 which generalizes Algorithms 3.1 and 3.2, we have:

Proposition 7. The over all computational complexity of Algorithm 3.3 is $\min(|A|, \lfloor 1/r \rfloor)T(|V|,|A|)$.
where $T(|V|,|A|)$ is the complexity of the maximum flow problem for a network with $|V|$ vertices and $|A|$ arcs.

Proof: Let $U(S^i,y^i)$ be the slope obtained from a minimum cut at the i-th repetition in Algorithm 3.3. Then we may assume that $U(S^i,y^i)=n_i r$ for some positive integer n_i. If the algorithm takes J repetitions, then we have

$$0<U(S^1,y^1)<U(S^2,y^2)<\ldots<U(S^J,y^J)<1 \quad \text{i.e. } 0<n_1<n_2<\ldots<n_J<1/r.$$

From Proposition 5 and the fact that $n_i \leq |A|$, it follows that at most $\min(|A|, \lfloor 1/r \rfloor)T(|V|,|A|)$ computation time is required. □

Example. Consider the network $N=(G=(V,A),\underline{c},\bar{c},r,\beta,s,t)$ with an additional arc $a^*=(t,s)$ in Fig.4. The underlying graph G has the vertex set $V=\{s, 1,2,3,4,t\}$ and the arc set $A=\{a_i:1 \leq i \leq 8\}$. The ordered pair attached to each arc $a\epsilon A$ means $(\bar{c}(a),\beta(a))$. We assume that a balancing rate r is equal to 1/5 and that $\underline{c}(a)=0$ for any $a\epsilon A$. Then we have $\underline{b}=0$ and $\bar{b}=375$. First, find the maximum flow value g^* in network N^*, and we have $g^*=100$. Second, calculate the maximum flow value $f^{**}(100)$ and the slope $U(S,100)$ of the network in Fig.5 where the value attached to each arc a is equal to $\min(\bar{c}(a),1/5\cdot 100+\beta(a))$. Fig.6 shows the maximum flow of the parametric problem (P_y) for $y=100$. Since $U(\{s,1,2\},100)=3/5$, we renew y as $y=(87-3/5\cdot 100)/(1-3/5)=135/2$, and continue Algorithm 3.3. Finally we have the maximum balanced flow with optimal value $f^*=135/2$ as shown in Fig.7. (The flow value $f(a)$ of each arc a is given in Figs. 6 and 7.)

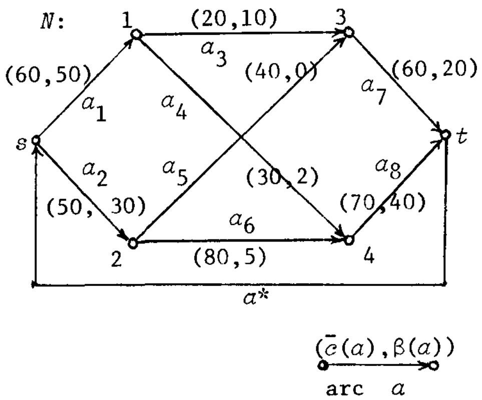

Fig.4

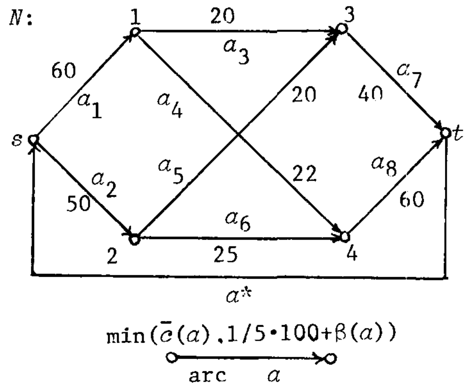

Fig.5

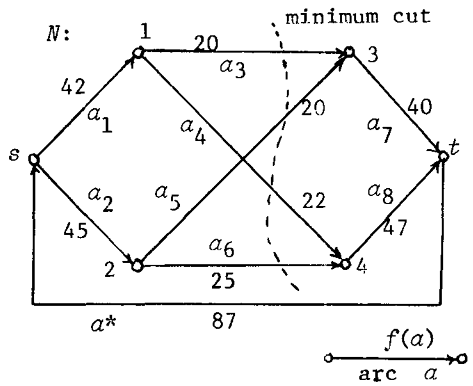

Fig.6

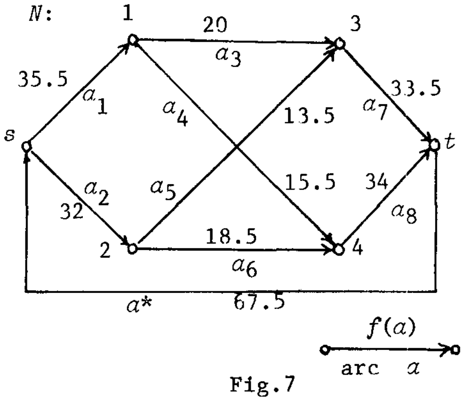

Fig.7

Acknowledgments: We would like to express our thanks to Professor Satoru Fujishige of the University of Tsukuba for his useful advice and comments.

References

[1] Cui, W.-T.: An algorithm for the maximum balanced flow problem. *Second Year Essay, Doctoral Program in Socio-Economic Planning, University of Tsukuba,* 1986.

[2] Ford, L. R. Jr., Fulkerson, D. R.: *Flows in Networks.* Princeton University Press. Princeton, N.J., 1962.

[3] Fujishige, S., Nakayama, A. and Cui, W.-T.: On the equivalence of the maximum balanced flow problem and the weighted minimax flow problem. (to appear).

[4] Ichimori, T., Ishii, H. and Nishida, T.: Weighted minimax real-valued flows. *Journal of the Operations Research Society of Japan*, Vol.24, No. 1 (1981), 52-60.

[5] Lawler, E. L.: *Combinatorial Optimization - Networks and Matroids.* Holt, Rinehart and Winston, New York, 1976.

[6] Minoux, M.: Flots équilibrés et flots avec sécurité. *E.D.F.-Bulletin de la Direction des Études et Recherches, Série C - Mathématiques, Informatique,* Vol.1 (1976), 5-16.

[7] Zimmermann, U.: Duality for balanced submodular flows. *Preprint* No.89, *Fachbereich Mathematik, Universität Kaiserslautern*, 1985.

Akira NAKAYAMA: Department of Management Sciences, Faculty of Commerce,
Otaru University,
Midori, Otaru, Hokkaido, 047, Japan.