Journal of the Operations Research
Society of Japan
Vol. 34, No. 1, March 1991

NP-COMPLETENESS AND APPROXIMATION ALGORITHM FOR THE MAXIMUM INTEGRAL VERTEX-BALANCED FLOW PROBLEM

Akira Nakayama
Fukushima University

(Received September 28, 1988; Revised April 24, 1990)

Abstract Minoux considered *the maximum balanced flow problem* of a *two-terminal* network, which is the problem of finding a maximum flow f in the network such that each arc-flow $f(a)$ $(a \in A)$ is bounded by a fixed proportion of the total flow value from the source to the sink, where A is the arc set of the network. He also proposed an algorithm for finding a maximum *integral* balanced flow, i.e., a maximum balanced flow satisfying that the value of each arc-flow of the network is integral. Integral balanced flows defined by Minoux can be regarded as one way to balance flows in the network. In this paper, we propose another way to balance flows in a two-terminal network N. To be exact, we consider *the maximum vertex-balanced flow problem* in network N, i.e., the problem of finding a maximum flow f' in N such that for each vertex $v \in V$ any arc-flow $f'(a)$ $(a \in \delta^-(v))$ entering v is bounded by a fixed proportion of the total flow $\sum\{f'(a) : a \in \delta^-(v)\}$ entering v, where V is the vertex set of N and $\delta^-(v)$ is the set of the arcs entering v. We intended to propose an algorithm for finding a maximum integral vertex-balanced flow in network N, but we found that the maximum integral vertex-balanced flow problem $(IVBP)$ is difficult.

Our main purpose in this paper is to prove that problem $(IVBP)$ is *NP-complete* and to propose a polynomial-time approximation algorithm for $(IVBP)$.

1. Introduction

Minoux [5] considered *the maximum balanced flow problem* (BP), i.e., the problem of finding a maximum flow in a *two-terminal* network with an additional constraint described in terms of a *balancing rate function* $\alpha : A \to \mathbf{R}_+ - \{0\}$, where A is the arc set of the network and $\mathbf{R}_+$ is the set of nonnegative reals. To put it another way, problem (BP) in the network is the problem of finding a maximum flow f in the network such that each arc-flow $f(a)$ $(a \in A)$ is bounded by a fixed proportion $\alpha(a)$ of the total flow value from the source to the sink of the network. This problem (BP) is motivated by Minoux's research of reliability analysis of communication networks. For example, 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 the number of the blocked routes is at most the fixed proportion of the total number of current routes from A to B. Statistics in [5] shows that few telephone lines break at the same time. If a flow from source s to sink t is balanced, then it is guaranteed that the value of the blocked arc-flow is at most the fixed proportion of the total flow value from s to t. Several algorithms

[1,5,6,7,9] are proposed for the maximum balanced flow problem.

Minoux [5] also considered the problem of finding a maximum *integral* balanced flow of the network, i.e., that of finding a maximum balanced flow f satisfying that each arc-flow of f is integral, and proposed an algorithm for this problem. By the way, Zimmermann [9] generalized Minoux's formulation of problem (BP) and conjectured the maximum integral balanced flow problem is *NP-hard*. If a balancing rate function α is constant, Cui's $O(\min\{m, \lfloor 1/r \rfloor\}T(n,m))$ algorithm is known, where $m = \mid A \mid$, $\alpha(a) = r$ for any arc $a \in A$, $\lfloor \theta \rfloor$ is the minimum integer greater than or equal to $\theta \in \mathbf{R}_+$ and $T(n,m)$ is the time for the maximum flow computation for a network with n vertices and m arcs.

In the present paper we propose another way to balance flows in a network. More precisely, we consider *the maximum vertex-balanced flow problem* of a two-terminal network N i.e., a maximum flow problem with an additional constraint described in terms of a *vertex-balancing rate function* $\gamma : V - \{s\} \to \mathbf{R}_+ - \{0\}$ where V is the vertex set and s is the source of N. The difference between a maximum balanced flow f and a maximum vertex-balanced flow g is as follows : each arc-flow of f is bounded by a fixed proportion of the total flow value from the source to the sink of the underlying graph $G = (V, A)$, while that of g entering each vertex $v \in V$ is bounded by a fixed proportion of the total flow entering v.

Especially, we consider the maximum *integral* vertex-balanced flow problem $(IVBP)$ of finding the maximum vertex-balanced flow g of network N such that each arc-flow of g is integral. Let $n_{i\,j}$ be the number of telephone lines between two adjacent spots i and j in a telephone network. The integrality condition C_{int} of integral vertex-balaned flows of N would be justified from such a situation that $n_{i\,j}$ corresponds to arc-flow $g((i,j))$ in the network. (In the case when $n_{i\,j}$ is very large it would be sufficient that we have a vertex-balanced flow without the condition C_{int} of problem (VBP). We will discuss this case in future researches.) At first, we intended to give an algorithm for problem $(IVBP)$, but we found that $(IVBP)$ is difficult. The main purpose of this paper is to prove that problem $(IVBP)$ is *NP-complete* and to propose a polynomial-time approximation algorithm for $(IVBP)$. The NP-completeness of problem $(IVBP)$ means that $(IVBP)$ is in class NP and that every problem in NP is reducible to $(IVBP)$ in polynomial-time, where NP is the class of problems that can be solved by *nondeterministic Turing machine* in polynomial-time.

2. The Maximum Vertex-Balanced Flow Problem

In this section we give the motivation of considering the maximum vertex-balanced flow problem, compared to the maximum balanced flow problem by Minoux. First we describe the maximum balanced flow problem in the following subsection.

2.1. The Maximum Balanced Flow Problem

Let $G = (V, A)$ be a *directed graph* with vertex set V and arc set A, where $V(G)$ ($A(G)$) will also be used as the vertex (arc) set of G. Given a directed graph $G = (V, A)$, a *capacity function* $c : A \to \mathbf{Z}_+$ and a *balancing rate function* $\alpha : A \to \mathbf{R}_+ - \{0\}$, consider a *two-terminal* network $N' = (G = (V, A), c, \alpha, s, t)$ where s is the source and t is the sink of G. Given two-terminal network N', *the maximum balanced flow problem* (BP) by Minoux is formulated as follows.

(BP): Maximize $f(a^*)$

subject to

$$D \cdot f = 0, \tag{2.1}$$

$$0 \le f(a) \le c(a) \quad (a \in A), \tag{2.2}$$

$$f(a) \le \alpha(a) f(a^*) \quad (a \in A), \tag{2.3}$$

where arc $a^* = (t, s) \notin A$ is added to the underlying graph G and D is the *vertex-arc incidence matrix* of G. If a function $f : A^* \to \mathbf{R}_+$ satisfies (2.1) and (2.2), then f is called a *feasible flow* in network N', where $A^* = A \cup \{a^*\}$. If a feasible flow f satisfies (2.3), then f is called *balanced.* The *value* of a feasible balanced flow f is defined as value $f(a^*)$. Problem (BP) is the problem of finding a feasible balanced flow of maximum value, called a maximum balanced flow. *The maximum integral balanced flow* is a maximum balanced flow f with the property that each arc-flow $f(a)$ ($a \in A$) is integral. In section 4 we will employ Cui's algorithm when we describe an approximation algorithm for the maximum integral vertex-balanced flow problem.

2.2. The Maximum Vertex-Balanced Flow Problem

A maximum balanced flow f in the previous subsection has the property that the value of each arc-flow of f in network N' is bounded by a fixed proportion of the total flow value from the source to the sink of N'. On the other hand, we give an another formulation of "balanced flows" in a network from slightly different point of view. We consider a feasible flow g such that for each vertex $v \in V$ any arc-flow $g(a)$ ($a \in \delta^-(v)$) entering vertex v, is bounded by a fixed proportion of the total flow entering v. We call g *a vertex-balanced flow.* Now we show a precise formulation of the maximum vertex-balanced flow problem.

Let $G = (V, A)$ be a directed graph. The number of the arcs coming in (going out of) a vertex $v \in V$ is called *in-degree* (*out-degree*) of v. Given a two-terminal network $N = (G = (V, A), c, \gamma, s, t)$ *the maximum vertex-balanced flow problem* (VBP) is formulated as follows:

(VBP): Maximize $g(a^*)$
subject to constraints (2.1), (2.2) and

$$(2.4) \qquad \max\{g(a) : a \in \delta^-(v)\} \le \gamma(v) g(\delta^-(v)) \qquad (v \in V - \{s\}),$$

where f should be replaced by g in (2.1) and (2.2), and $\gamma : V - \{s\} \to \mathbf{R}_+ - \{0\}$ is a *vertex-balancing rate function.* $\delta^+(v)$ ($\delta^-(v)$) is the set of the arcs with v as their *initial* (*terminal*) vertices in G, respectively and $g(\delta^-(v)) = \sum\{g(a) : a \in \delta^-(v)\}$. If a function $g : A^* \to \mathbf{R}_+$ satisfies (2.1), (2.2) and (2.4), then g is called a *feasible vertex-balanced flow* in network N. The *value* of a feasible vertex-balanced flow g is defined as value $g(a^*)$. The maximum vertex-balanced flow problem is the problem of finding a feasible vertex-balanced flow of maximum value.

In subsequent sections, we consider *the maximum integral vertex-balanced flow problem* ($IVBP$), i.e., the problem of finding a maximum vertex-balanced flow g satisfying that the value $g(a)$ of each arc $a \in A$ is integral. The example of the telephone routing problem in the previous section would need this integrality condition. The following proposition states the relation between in-degree of each vertex in N and a feasible vertex-balanced flow in N.

Proposition 1. Let g be a feasible vertex-balanced flow in the network N. For each vertex $v \in V - \{s\}$, we have the following (2.5) and (2.6).

(2.5) If $\gamma(v) < 1/ \mid \delta^-(v) \mid$, then we have $g(a) = 0 \quad (a \in \delta^-(v))$.

(2.6) If $\gamma(v) = 1/ \mid \delta^-(v) \mid$, then we have $g(a) = g(\delta^-(v))\gamma(v) \quad (a \in \delta^-(v))$.

Proof: Let g be a feasible vertex-balanced flow in N. From (2.3), for each vertex $v \in V - \{s\}$ we have

$$(2.7) \qquad g(a) \le \gamma(v) g(\delta^-(v)) \qquad (a \in \delta^-(v)\ ).$$

Adding each inequality in (2.7), we have

$$(2.8) \qquad \sum\{g(a) : a \in \delta^-(v)\} \le \gamma(v) g(\delta^-(v)) \mid \delta^-(v) \mid.$$

If $g(\delta^-(v)) > 0$ and $\gamma(v) < 1/ \mid \delta^-(v) \mid$, then from (2.8) we have a contradiction. Hence we have (2.5). Assume $g(\delta^-(v)) > 0$ and $\gamma(v) = 1/ \mid \delta^-(v) \mid$. From (2.8), we have equlity case for the inequality of any arc $a \in \delta^-(v)$ in (2.7). Hence we have (2.6). ■

In the following section, we will prove that the maximum integral vertex-balanced flow problem is in a class of difficult problems where we assume $\gamma(v) \ge$

$1/\mid \delta^-(v) \mid \ (v \in V - \{s\})$.

3. NP-Completeness of the Maximum Integral Vertex-Balanced Flow Problem

In this section we prove that a maximum vertex-balanced flow problem is *NP-complete*. The definition of NP-complete is described with additional preliminaries in the following subsection.

3.1. Additional Preliminaries

Given a two-terminal network $N = (G = (V, A), c, \gamma, s, t)$, Consider the following yes-no question (BQ):

(3.1) (BQ): *Instance*: $J \in \mathbf{Z}_+ - \{0\}$ and network N.
Question : Is there a feasible integral vertex-balanced flow g in N such that $g(a^*) \leq J$?

where $\mathbf{Z}_+$ is the set of nonnegative integers. Let NP be the class of yes-no questions that can be solved by *nondeternimistic Turing machine* in polynomial-time. We say that a yes-no question L' is polynomial-time reducible to a yes-no question L if we have a polynomial-time bounded Turing machine that for each input x produces an output y that is in L if x is in L'. A yes-no question L is called *NP-complete* if L is in class NP and each yes-no question in NP is polynomial-time reducible to L. In subsequent sections we will prove that the question (BQ) in (3.1) is NP-complete.

3.2. The Problem of Integral Network Flows with Multipliers

We introduce one of NP-complete problems, called the problem of *integral network flows with multipliers*. Given a directed graph $G' = (V', A')$, a capacity function $c : A \to \mathbf{Z}_+ - \{0\}$ and a *multiplier function* $h : V - \{s, t\} \to \mathbf{Z}_+ - \{0\}$, consider a two-terminal network $N_h' = (G' = (V', A'), c, h, s, t)$, where $\delta^-(s) = \delta^+(t) = \phi$. The problem (MP) of integral network flows with multipliers is defined as follows:

(MP): Maximize $g'(\delta^-(t))$
subject to

$$0 \leq g'(a) \leq c(a) \qquad (a \in A'), \tag{3.2}$$

$$h(v) g'(\delta^-(v)) = g'(\delta^+(v)) \qquad (v \in V' - \{s, t\}), \tag{3.3}$$

where $g'(B) = \sum \{g'(a) : a \in B\}$ for $B \subset A'$. If a function $g' : A' \to \mathbf{R}_+$ satisfies (3.2) and (3.3), then g' is called a *feasible multiplier flow* in the network N_h'. If each arc-flow $g'(a)$ $(a \in A')$ of a feasible multiplier flow g' is integral, then g' is called integral. The value of a feasible multiplier flow g' is defined as value $g'(\delta^-(t))$.

The problem of integral network flows with multipliers is that of finding a feasible integral multiplier flow of maximum value, called a *maximum integral multiplier flow*. Consider the following yes-no question (MQ):

(3.4) (MQ): *Instance* : $J \in \mathbf{Z}^+ - \{0\}$ and network N'_h.

Question : Is there a feasible integral multiplier flow g' in N'_h such that $g'(\delta^-(t)) \geq J$?

By the way, the question (MQ) is known as one of NP-complete problems:

Theorem 2. [3,8] Question (MQ) in (3.4) is NP-complete. ∎

Consider the following question $(MQ)'$, a special case of (MQ) :

(3.5) $(MQ)'$: Question (MQ) with function h' satisfying

$$\max\{h'(v) : v \in V' - \{s,t\}\} \leq \max\{\lfloor \log c(a) \rfloor : a \in A'\},$$

where functin h in (MQ) should be replaced by h' and $\lfloor \theta \rfloor$ is the minimum integer greater than or equal to $\theta \in \mathbf{R}_+$. To show that the question (BQ) is NP-complete, we will first prove that the question $(MQ)'$ in place of (MQ) is NP-complete and then that $(MQ)'$ is polynomial-time reducible to (BQ) in the following sections.

3.3. NP-Completeness of a Subproblem in Problem of Integral Network Flows with Multipliers

We transform *SUBSET SUM* (SS) to $(MQ)'$, where *SUB- SET SUM* is a yes-no question defined as follows:

(3.6) (SS): *Instance* : $J \in \mathbf{Z}_+ - \{0\}$, $s(e) \in \mathbf{Z}_+ - \{0\}$ $(e \in S)$.

Question : Is there a subset $S' \subset S$ such that $\sum\{s(e) : e \in S'\} = J$?

where S is a finite set. For the question (SS) we have:

Theorem 3. [3] Question (SS) is NP-complete. ∎

Given positive integers $s(i)$ $(1 \leq i \leq r)$, let

(3.7) $s(i) = \sum\{b(i,j)2^j : 0 \leq j \leq \lfloor \log s(i) \rfloor, b(i,j) \in \{0,1\}\}$,

(3.8) $\theta(s(i)) = \{j+1 : 0 \leq j \leq \lfloor \log s(i) \rfloor, b(i,j) = 1\}$,

where $r = | S |$ and $S = \{i : 1 \leq i \leq r\}$. Note that $b(i,j)$ is the coefficient of binary expansion of $s(i)$. Simply we put $\theta(s(i)) = \{\theta_{i\ j} \in \mathbf{Z}_+ - \{0\} : \theta_{i\ j} < \theta_{i\ j+1}, 1 \leq j \leq n_i\}$, where $n_i = | \theta(s(i)) |$. Let $P(k) = (a_1, a_2, ..., a_k)$ be (an *elementary* directed) path of *length* k, where the terminal vertex of arc a_i is equal to the initial vertex of arc a_{i+1} for each i $(1 \leq i \leq k-1)$ and the length is the number of the arcs. Now do the following construction:

Construction-I:

Given positive integers J and $s(i)$ $(1 \leq i \leq r)$, make a network $N''_{h''} = (G'' = (V'', A''), c'', h'', s, t)$ as follows:

Step 1: Construct the graph $G'' = (V', A')$ as in Fig. 1.1, i.e.,

$V'' = \{s, w, t\} \cup \{\bigcup\{V(H_i) : 1 \leq i \leq r\},$

$A'' = \{(s, s_i^+), (s_i^-, w) : 1 \leq i \leq r\} \cup \{(w, t)\} \cup A_r,$

where $A_r = \bigcup\{A(H_i) : 1 \leq i \leq r\}$ and $V(H_i)$ $(A(H_i))$ is the vertex (arc) set of graph H_i in Fig. 1.2. Each H_i $(1 \leq i \leq r)$ is defined as follows.

$V(H_i) = \bigcup\{V(P_{i\ j}) : 1 \leq j \leq n_i\},$

$A(H_i) = \bigcup\{A(P_{i\ j}) : 1 \leq j \leq n_i\},$

where $P_{i\ j} = P(\theta_{i\ j})$ and $V(P_{i\ j})$ $(A(P_{i\ j}))$ is the vertex (arc) set of the path of length $\theta_{i\ j}$ in Fig. 1.3. $P_{i\ j}$ is defined as follows.

$V(P_{i\ j}) = \{v_{i\ j}(m) : 0 \leq m \leq \theta_{i\ j}\},$

$A(P_{i\ j}) = \{(v_{i\ j}(m-1), v_{i\ j}(m)) : 1 \leq m \leq \theta_{i\ j}\},$

where $v_{i\ j}(0) = s_i^+$ and $v_{i\ j}(\theta_{i\ j}) = s_i^-$.

Step 2: Define the capacity function c'' as

$c''((s, s_i^+)) = 1, \quad c''((s_i^-), w)) = s(i) \quad (1 \leq i \leq r), \quad c''((w, t)) = J,$

For arcs of each path $P_{i\ j}$ $(1 \leq i \leq r,\ 1 \leq j \leq n_i)$, define

$c''((v_{i\ j}(m-1), v_{i\ j}(m)) = 2^m \quad (1 \leq m \leq \theta_{i\ j}).$

graph $G_i = (V'', A'')$:

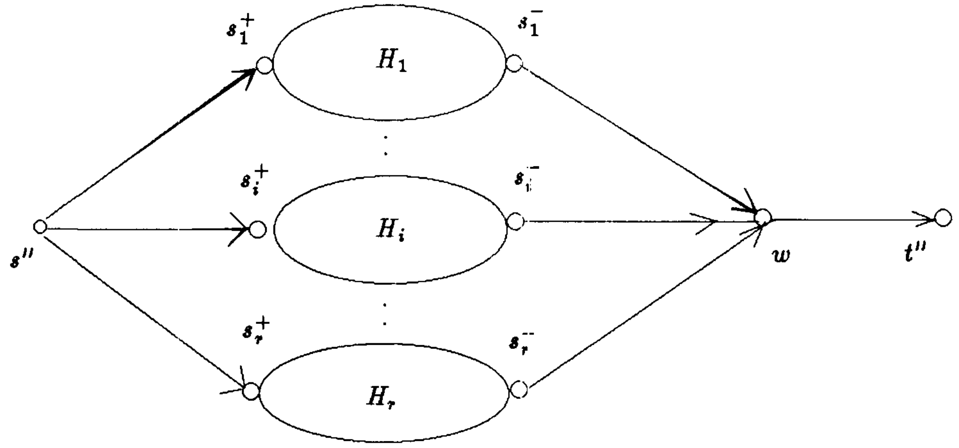

Fig. 1.1

Step 3: Define multiplier function h'' as

$$h''(s_i^+) = n_i, \quad h''(s_i^-) = 1 \quad (1 \le i \le r), \quad h''(w) = 1,$$

For vertices of each path $P_{i\ j}$ $(1 \le i \le r,\ 1 \le j \le n_i)$, define

$$h''(v_{i\ j}(m)) = 2 \quad (0 \le m \le \theta_{i\ j} - 1).$$

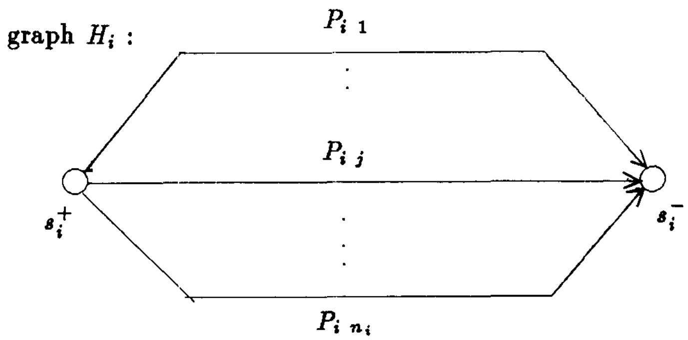

Fig. 1.2

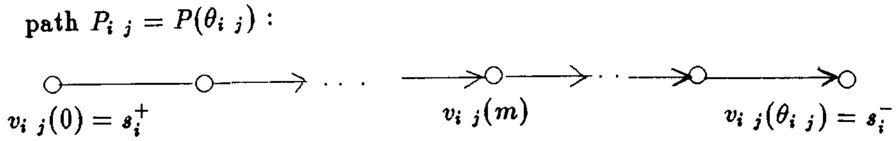

Fig. 1.3

We easily have the following proposition, which shows that we may assume $\{i : s(i) \ge 3, 1 \le i \le r\} \ne \phi$.

Proposition 4. If $s(i) \le 2$ for each i $(1 \le i \le r)$, then we can find a polynomial-time algorithm for the problem (SS). ■

From Proposition 4 we have:

Proposition 5. (SS) is polynomial-time reducible to $(MQ)'$.

Proof: Assume $\{i : s(i) \ge 3, 1 \le i \le r\} \ne \phi$. Given positive integers J and $s(i)$ $(1 \le i \le r)$, we see that it takes $O(r\eta^2)$ to construct network $N''_{h''}$, where $\eta = \max\{\lfloor \log s(i) \rfloor : 1 \le i \le r\}$. For the function h'' we have

$$(3.9) \qquad \max\{h''(v) : v \in V'' - \{s,t\}\} \le \max\{\eta, 2\} \le \max\{\lfloor \log c''(a) \rfloor : a \in A''\}.$$

From *Construction-I* and (3.9), we can use $N''_{h''}$ as an instance of $(MQ)'$. If there is a subset $S' \subset S$ such that $\sum\{s(i) : i \in S'\} = J$, then define a function g satisfying the following (3.10) $\sim$ (3.12):

(3.10) $g((w,t)) = c''((w,t))$,

(3.11) For each $i \in S'$,

$g((s,s_i^+)) = c''((s,s_i^+))$, $g((s_i^-,w)) = c''((s_i^-,w))$,

$g(a) = c''(a) \quad (a \in A(H_i)$).

(3.12) For each $i \in S - S'$, $g((s,s_i^+)) = g((s_i^-,w)) = 0$, $\quad g(a) = 0 \quad (a \in A(H_i))$.

Then it is easy to see that g is a feasible integral multiplier flow in $N''_{h''}$ such that $g(\delta^-(t)) = J$. Assume that we have a feasible integral multiplier flow g' in $N''_{h''}$ satisfying $g'(\delta^-(t)) \geq J$. Then we have $g'(\delta^-(t)) = J$. Let $S'' = \{i : g'((s,s_i^+)) = 1,\ 1 \leq i \leq r\}$, and we have (3.10) $\sim$ (3.12) for g' and S'' where g and S' should be replaced by g' and S''. From (3.10) $\sim$ (3.12) and $h''(w) = 1$, we have $\sum\{s(i) : i \in S''\} = J$. ■

Now we have the following theorem needed in later discussion:

Theorem 6. The question $(MQ)'$ in (12) is NP-complete. ■

Proof: It is easy to see that $(MQ)' \in NP$. From Theorem 3 and Proposition 5 we see that $(MQ)'$ is NP-complete.

In the following section, we present how to reduce $(MQ)'$ to (BQ).

3.4. Reduction of the Subproblem to the Maximum Integral Vertex-Balanced Flow problem

Given a network $N'_{h'} = (G' = (V', A'), c, h', s, t,)$ of $(MQ)'$, we construct a new network $N_{N'_{h'}} = (H = (X, E), c^o, \gamma, s'', t'')$ with source s'' and sink t'' ($s'', t'' \in X$) from $N'_{h'}$ by the following construction, where we have $\delta^-(s) = \delta^+(t) = \phi$ for the graph G'. Moreover we may assume that G' has no arcs joining s and t.

Construction-II:

Step 1 (Definition of the vertex set X of H) : Taking triple vertices v_1,v_2 and v_3 for each $v \in V' - \{s,t\}$, define

$X_{123} = X_1 \cup X_2 \cup X_3 \cup \{s'', t''\}$,

where $X_i = \{v_i : v \in V' - \{s,t\}\} \quad (1 \leq i \leq 3)$.

Step 2 (Definition of the arc set E of H) :

(2.1): Define five sets E_1, $E_{2\ i}$ $(i = 1, 2)$, E_s and E_t by

$E_1 = \{(v_3, w_1) \in X_{123} \times X_{123} : (v, w) \in A', v, w \notin \{s,t\}\}$,

$E_{2\ i} = \{(v_i, v_{i+1}) \in X_{123} \times X_{123} : v \in V' - \{s,t\}\}$,

$$E_s = \{(s'', u_1) \in \{s''\} \times X_{123} : (s, u) \in A', u \in V'\},$$

$$E_t = \{(u_3, t'') \in X_{123} \times \{t''\} : (u, t) \in A', u \in V'\}.$$

(2.2): Make the multiple arc set $E(v)$ defined by

$$E(v) = \{(s'', v_2)_i : 1 \le i \le h'(v) - 1\},$$

where $v \in W \equiv \{w \in V' - \{s, t\} : h'(w) > 1\}$. Let

$$E = E_1 \cup E_{2\ 1} \cup E_{2\ 2} \cup E_s \cup E_t \cup E_W,$$

where $E_W = \bigcup\{E(v) : v \in W\}$.

Step 3: Define a vertex-balancing rate function $\gamma : X - \{s''\} \to \mathbf{R}_+$ and a capacity function $c^o : E \to \mathbf{R}_+$ as follows.

$$\gamma(v_2) = 1/h'(v) \ (v_2 \in X_2),$$
$$\gamma(w) = 1 \quad (w \in X_1 \cup X_3 \cup \{t''\}),$$
$$c^o((v_3, w_1)) = c(v, w) \quad ((v_3, w_1) \in E_1),$$
$$c^o((v_1, v_2)) = c(\delta^-(v)) \quad ((v_1, v_2) \in E_{2\ 1}),$$
$$c^o((v_2, v_3)) = c(\delta^+(v)) \quad ((v_2, v_3) \in E_{2\ 2}),$$
$$c^o((s'', u_1)) = c((s, u)) \quad ((s'', u_1) \in E_s),$$
$$c^o((u_3, t'')) = c((u, t)) \quad ((u_3, t'') \in E_t),$$

For each $v \in W$, $c^o(e) = c(\delta^-(v)) \quad (e \in E(v))$.

Then we have:

Proposition 7. **We have a feasible integral vertex-balanced flow $f : E \to \mathbf{R}_+$ in the network $N_{N'_{h'}}$ satisfying $f(\delta^-(t'') \ge J$ if and only if we have a feasible integral multiplier flow $g : A' \to \mathbf{R}_+$ satisfying $g(\delta^-(t)) \ge J$ in the network $N'_{h'}$.**

Proof: Let $g : A' \to \mathbf{R}_+$ be a feasible integral multiplier flow satisfying $g(\delta^-(t)) \ge J$ in $N'_{h'}$. Then we define a function $f : E \to \mathbf{R}_+$ by using g as follows:

(3.13) $f((v_3, w_1)) = g((v, w)) \quad ((v_3, w_1) \in E_1)$,

(3.14) $f((v_1, v_2)) = g(\delta^-(v)) \quad ((v_1, v_2) \in E_{2\ 1})$,

(3.15) $f((v_2, v_3)) = g(\delta^+(v)) \quad ((v_2, v_3) \in E_{2\ 2})$,

(3.16) $f((s'', u_1)) = g((s, u)) \quad ((s'', u_1) \in E_s)$,

(3.17) $f((u_3, t'')) = g((u, t)) \quad ((u_3, t'') \in E_t)$,

(3.18) For each $E(v)$ $(v \in W)$, $f(e) = g(\delta^-(v)) \quad (e \in E(v))$.

From (3.13) $\sim$ (3.18) and the property of the flow g, we have a feasible integral vertex-balanced flow f satisfying $f(\delta^-(t'')) \ge J$. Let $f : E \to \mathbf{R}_+$ be a feasible integral vertex-balanced flow satisfying $f(\delta^-(t'')) \ge J$ in $N_{N'_{h'}}$. Then define a function $g : A' \to \mathbf{R}_+$ by using f as follows:

(3.19) $g((v,w)) = f((v_3, w_1)) \quad ((v,w) \in A',\ v \neq s,\ w \neq t)$,

(3.20) $g((s,u)) = f((s'', u_1)) \quad ((s,u) \in A')$,

(3.21) $g((u,t)) = f((u_3, t'')) \quad ((u,t) \in A')$.

From (3.19) $\sim$ (3.21), Proposition 1 and the property of the flow f, it is easy to see that g is a feasible integral multiplier flow satisfying $g(\delta^-(t)) \geq J$. Note that the function h' of $(MQ)'$ is bounded by a polynomial function of the size of input data and thus it takes $O(|V'| \max\{\lfloor \log c(a) \rfloor : a \in A'\} + |A'|)$ time to construct $N_{N'_{h'}}$. ■

We give our main theorem:

Theorem 8. Question (BQ) is NP-complete.

Proof: We reduce the question $(MQ)'$ to (BQ). It is easy to see $(BQ) \in NP$. From Proposition 7 and *Construction-II*, we see that $(MQ)'$ is polynomial-time reducible to (BQ). From Theorem 6, we conclude that (BQ) is NP-complete. ■

Here, we give an example showing the relation between a feasible integral vertex-balanced flow with requirement in $N_{N'_{h'}}$ and a feasible integral multiplier flow with requirement in $N'_{h'}$.

Example. Consider network $N'_{h'} = (G' = (V', A'), c, h', s, t)$ in Fig.2 and $J = 15$ as an instance of $(MQ)'$, where G' with source s and sink t has $V' = \{s, u, v, t\}$ and $A' = \{(s,u), (s,v), (u,v), (v,u), (u,t), (v,t)\}$. The ordered pair $(c(a), g(a))$ is attached to each arc $a \in A'$, where $g(a)$ $(a \in A')$ is the value of arc-flow of a feasible integral multiplier flow g satisfying $g(\delta^-(t)) \geq 15$. After *Construction-II*, we have the network $N_{N'_{h'}} = (H = (X, E), c^o, \gamma, s'', t'')$ in Fig.3, where the ordered pair attached to each arc $a \in E$ is $(c^o(a), f(a))$ and $f(a)$ is the value of arc-flow of a feasible integral vertex-balanced flow f satisfying $f(\delta^-(t)) \geq 15$, which is obtained from g.

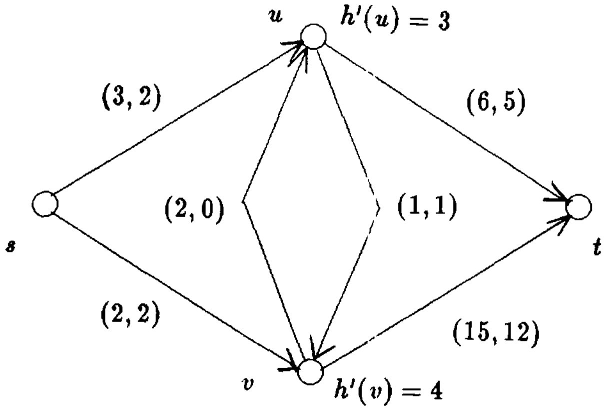

Fig.2

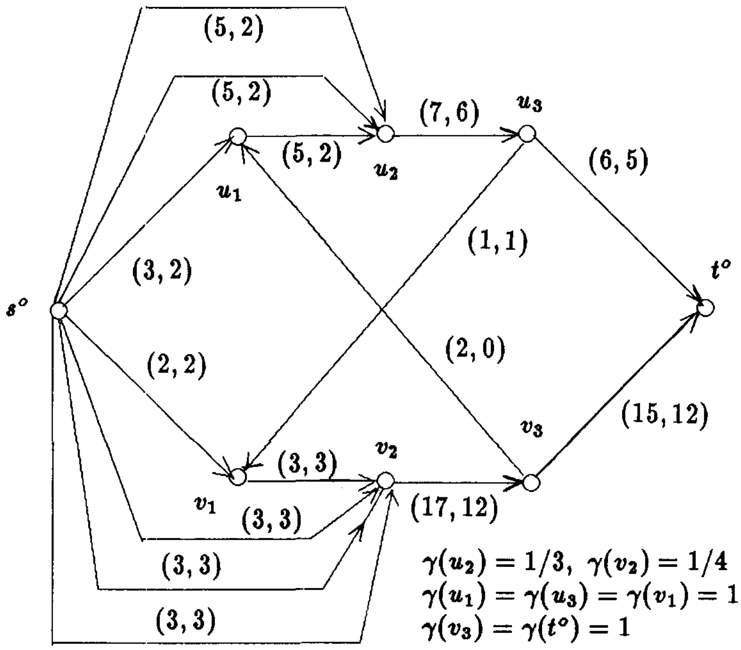

Fig.3

4. Approximation Algorithm for the Maximum Integral Vertex-Balanced Flow Problem

In the previous section we showed the maximum integral vertex-balanced flow problem $(IVBP)$ is NP-complete. Here, we propose an approximation algorithm for problem $(IVBP)$. Let f be an integral vertex-balanced flow in network $N = (G = (V, A), c, \gamma, s, t)$. Then we put the following assumption:

(4.1) $\max\{f(a) : a \in A\} \leq f(a^*)$,

where $a^* = (t, s) \notin A$. The assumption (4.1) means that any arc-flow of network N is less than or equal to the total net flow $f(a^*)$ from the source s to the sink t of N. We can have (4.1) since any arc-flow on cycles in the underlying graph G does not affect the total flow from s to t, though it may affect constraint (2.4). Moreover we omit constraint (2.4) to employ an algorithm for the maximum balanced flow problem (BP). For network N, define function $\alpha' : A \to \mathbf{R}_+ - \{0\}$ by

(4.2) For each $v \in V - \{s\}$, $\alpha'(a) = \gamma(v) \quad (a \in \delta^-(v))$,

(4.3) For each $(u, s) \in A \quad \alpha'((u, s)) = 1$,

where s is the source of N. If we apply Minoux's algorithm of the maximum integral balanced flow problem (IBP) for network $N_{\alpha'} = (G = (V, A), c, \alpha', s, t)$ with constant balancing rate function α', then we have an approximate maximum integarl vertex-balanced flow of network N. We also give another approximate approach of problem $(IVBP)$ for network N. We use Cui's algorithm for problem (IBP) with a constant balancing rate function. First we relax balancing rate function α' as follows:

Proposition 9. We have positive integers $(d_a : a \in A)$ such that

(4.4) $d_a/n < \alpha'(a) \leq (d_a + 1)/n,$

where $n = | V |$.

Proof: We assumed $\gamma(v) \geq 1/| \delta^-(v) |$ for each $v \in V - \{s\}$ in Section 2. From $1/n \leq 1/ | \delta^-(v) |$ ($v \in V - \{s\}$), we have this proposition. ■

Then do the following *Construction-III*, where $A' = \{a \in A : \alpha'(a) \neq 1\}$.

Construction-III

Input: $(G = (V, A),\ ((d_a + 1)/n : a \in A),\ \alpha', c)$
Output: $(G^* = (V^*, A^*), \alpha^*, c^*)$

Step 1: (Definition of G^*) For each $a \in A'$, replace a by arc $(\partial^+(a), x_a)$ and multiple-arc set $A_a \equiv \{(x_a, \partial^-(a))_i : 1 \leq i \leq d_v + 1\}$, where $\partial^+(a)$ ($\partial^-(a)$) is the *initial* (*terminal*) vertex of arc a and $x_a \notin V$ is a new vertex. Define

$$V^* = V \cup \{x_a : a \in A'\},$$
$$A^* = \{(\partial^+(a), x_a) : a \in A'\} \cup \{a : a \in A - A'\} \cup (\bigcup\{A_a : a \in A'\}).$$

Step 2: Define a balancing rate function α^* by

$$\alpha^*(a) = 1/n \quad (a \in \bigcup\{A_a : a \in A'\}),$$
$$\alpha^*(a) = 1 \quad (a \in \{(\partial^+(a), x_a) : a \in A'\} \cup \{a : a \in A - A'\}).$$

Step 3: Define a capacity function c^* by

$$c^*((\partial^+(a), x_a)) = c(a) \quad ((\partial^+(a), x_a) \in \{(\partial^+(a), x_a) : a \in A'\}),$$
$$c^*(a) = c(a) \quad (a \in \{a : a \in A - A'\}),$$

For each $a \in A'$, $c^*(e) = c(a)$ ($e \in A_a$).

Consider two networks N and $N^* = (G^* = (V^*, A^*), c^*, \alpha^*, s, t)$. Then we have the following proposition :

Proposition 10. Let v_N (v_{N^*}) be the value of a maximum integral vertex-balanced (balanced) flow in network N (N^*), respectively. Then we can find the value v_{N^*} such that $v_N \leq v_{N^*}$ in polynomial time.

Proof: From *Construction-III*, we have:

(4.5) $|V^*| \le |V| + |A|$,

(4.6) $|A^*| \le |V||A|$,

where $G = (V, A)$ (resp. $G^* = (V^*, A^*)$) is the underlying graph of network N (resp. N^*). By using Cui's algorithm for the maximum integral balanced flow problem with a constant balancing rate function, it takes $O(nT(n+m, nm))$ time to obtain the value v_{N^*} where $n = |V|$, $m = |A|$ and $T(n, m)$ is the time for the maximum flow computation for a network with n vertices and m arcs. It is easy to see that we have $v_N \le v_{N^*}$. ■

5. Conclusion and Future Research

In this paper we showed that the maximum integral vertex-balanced flow problem $(IVBP)$ is NP-complete, and proposed an polynomial approximation algorithm for problem $(IVBP)$. In this section we give one direction of the generalizations of the maximum vertex-balanced flow problem (VBP). Zimmermann [9] generalized the maximum balanced flow problem (BP) for network $N' = (G = (V, A), \alpha, \beta, s, t)$, where $G = (V, A)$ is a directed graph, $\alpha : A \to \mathbf{R}_+ - \{0\}$ is a balancing rate function, $c_o : A \to \mathbf{R}_+$ and $c^o : A \to \mathbf{R}_+$ are two capacity functions, $\beta : A \to \mathbf{R}$ is a function, s is the source and t is the sink of G. The constraints in Minoux's formulation are (2.1) $\sim$ (2.3), while those of Zimmermann's are (2.1), together with

(5.1) $c_o(a) \le f(a) \le c^o(a) \quad (a \in A)$,

(5.2) $f(a) \le \alpha(a) f(a^*) + \beta(a) \quad (a \in A)$,

where constraint (5.2) means that each arc-flow $f(a)$ ($a \in A$) of network N' is bounded by a fixed proportion $\alpha(a)$ of the total flow $f(a^*)$ from source s to sink t plus a fixed constant $\beta(a)$. Zimmermann conjectured that the maximum integral balanced flow problem (IBP) is NP-hard. On the other hand, we generalize problem (VBP) for network $N'' = (G = (V, A), \gamma, \eta, s, t)$, where $\gamma : V - \{s\} \to \mathbf{R}_+ - \{0\}$ is a vertex-balancing rate function and $\eta : V - \{s\} \to \mathbf{R}$. The constraints in our formulation are (2.1), (5.1) and

(5.3) $\max\{f(a) : a \in \delta^-(v)\} \le \gamma(v) f(\delta^-(v)) + \eta(v) \quad (v \in V - \{s\})$.

By modifying *Construction-III* slightly and employing Cui's algorithm for problem (IBP) with a constant balancing rate function, we can also have an approximate maximum integral vertex-balanced flow of network N'' in polynomial-time.

Acknowledgement

The auther wishes to thank Professor Satoru Fujishige of University of Tsukuba and referees for their helpfull comments and valuable advice.

References

[1] Cui, W.-T.: A Network Simplex Method for the Maximum Balanced Flow Problem. *Journal of the Operations Research Society of Japan* **31** (1988) 551-564.

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

[3] Garey, M. R. and Johnson, D. S. : *Computers and Intractability*; A Guide to the Theory of NP-Completeness, Freeman, 1977.

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

[5] Minoux, M.: Flots Équilibrés et Flots avec Sécurité. E.D.F.-Bullentin de la Direction des Études et Recherches, Série C-Mathématiques, *Informatique* **1** (1976) 5-16.

[6] Nakayama, A: A Polynomial Algorithm for the Maximum Balanced Flow Problem with a Constant Balancing Rate Function. *Journal of the Operations Research Society of Japan* **29** (1986) 400-410.

[7] Nakayama, A: A Polynomial-Time Binary Search Algorithm for the Maximum Balanced Flow Problem. *Journal of the Operations Research Society of Japan* **33** (1990) 1-11.

[8] Sahni, S.: Computational Related Problems. *SIAM Journal Computing* **3** (1974).

[9] Zimmermann, U.: Duality for Balanced Submodular Flows. Preprint No.89, Fachbereich Mathematik, Universität Kaiserslautern, 1985.

Akira NAKAYAMA: Faculty of Administration and Social Sciences, Fukushima University, Sugumichi, Asakawa, Matsukawa, Fukushima, 960-12, Japan.