

A NETWORK SIMPLEX METHOD FOR THE MAXIMUM BALANCED FLOW PROBLEM

Wentian Cui
University of Tsukuba

(Received February 1, 1988)

Abstract We present a network simplex method for the maximum balanced flow problem, i.e., the problem of finding a maximum flow in a source-to-sink network such that each arc-flow value does not exceed a fixed proportion of the total flow value from the source to the sink. We generalize the notion of strong feasibility in the network simplex method for the maximum flow problem to give a finiteness proof for the new algorithm. We also consider the maximum balanced integral flow problem, and show that the problem can be solved in polynomial-time for a special case when the balancing rate function is constant.

1. Introduction

A flow in a source-to-sink network is called balanced if each arc-flow value does not exceed a fixed proportion of the total flow value from the source to the sink. The maximum balanced flow problem is to find a balanced flow with maximum total flow value from the source to the sink. This problem was introduced by M. Minoux [8], who mentions an application in the reliability consideration of communication networks. In a telephone network between two cities, if we consider the telephone routing as a flow and assign a balanced flow in the network, it is guaranteed that at most the fixed proportion of the total flow value from the source to the sink is blocked when some connection line breaks down. Recently, U. Zimmermann [10] has considered the maximum balanced flow problem with submodular constraints and A. Nakayama [9] has proposed a polynomial-time algorithm for this problem with a constant balancing rate function.

On the other hand, T. Ichimori, H. Ishii and T. Nishida [6] considered the weighted minimax flow problem. S. Fujishige, A. Nakayama and W.-T. Cui [5] have shown that the weighted minimax flow problem is equivalent to the maximum balanced flow problem, and the complexity of Zimmermann's algorithm [10] is the same as that of Ichimori, Ishii and

Nishida's [6].

In spite of thirty years or more research on network flow algorithms, most of the solution method in use is still the simplex method, appropriately specialized and implemented [7]. It has been established (see, for example [1], [3]) that simplex-type procedures are very efficient in practice for solving ordinary and generalized network flow problems. Its major deficiency is that little is known about its theoretical complexity. The main purpose of this paper is to develop a variant of the network simplex algorithm for the maximum balanced flow problem which operates combinatorially on a graph. In particular, the 'strongly convergent' pivot rule developed by W. H. Cunningham [1] for the maximum flow problem is generalized to the maximum balanced flow problem so that a simple pivoting rule ensures the finiteness of the algorithm.

The above mentioned methods all require arc-flows to have nonnegative real values. Until now, no polynomial algorithm is known for solving the corresponding maximum balanced integral flow problem with real balancing rate functions, and hence Zimmermann [10] conjectures that the problem is NP-hard. In Section 5, we show that the maximum balanced integral flow problem can be solved in polynomial-time for a special case when the balancing rate function is a constant function.

2. The Maximum Balanced Flow Problem

Let $G = (V, A)$ denote a connected directed graph with a vertex set V and an arc set A , and I and R_+ , respectively, denote the sets of integers and of nonnegative reals. Given a capacity function $c : A \rightarrow R_+$, a balancing rate function $\alpha : A \rightarrow R_+$ and a function $\beta : A \rightarrow R_+$, consider the source-to-sink network $N = (G = (V, A), c, \alpha, \beta, s, t)$ where s and t are, respectively, the source and the sink of G . Let $a_0 = (t, s) \notin A$ be the arc added to the underlying graph G , and D be the vertex-arc incidence matrix of $G_0 = (V, A \cup \{a_0\})$. Then the *maximum balanced flow problem* is formulated as the following LP problem (P):

$$\begin{aligned}
 (P) \quad & \text{Maximize} \quad \varphi(a_0) \\
 (2.1) \quad & \text{subject to} \quad D\varphi = 0, \\
 (2.2) \quad & 0 \leq \varphi(a) \leq c(a) \quad (a \in A), \\
 (2.3) \quad & \varphi(a) \leq \alpha(a)\varphi(a_0) + \beta(a) \quad (a \in A).
 \end{aligned}$$

Without loss of generality, assume that $\beta(a) \leq c(a)$ ($a \in A$) and if $\alpha(a) = 0$ then $\beta(a) = c(a)$. A function $\varphi : A \rightarrow R$ satisfying (2.1) and (2.2) is called a *flow*, and a flow φ satisfying (2.3) is called a *balanced flow*. The flow value of φ is $\varphi(a_0)$. A *maximum balanced flow* is a balanced flow φ in N of maximum flow value. Note that $\varphi = 0$ is a balanced flow in N .

Let $p(v)$ be the dual variable identified with the equation corresponding to vertex v in (2.1), and $g(a)$ and $y(a)$ be the dual variables identified with the capacity constraint (2.2)

and balancing constraint (2.3) on arc a , respectively. Then the dual problem (DP) of (P) can be written as follows:

$$\begin{aligned}
 (DP) \quad & \text{Minimize} \quad \sum_{a \in A} c(a)g(a) + \sum_{a \in A} \beta(a)y(a) \\
 (2.4) \quad & \text{subject to} \quad g(a) + y(a) + p(\partial^+ a) - p(\partial^- a) \geq 0 \quad (a \in A), \\
 (2.5) \quad & p(t) - p(s) - \sum_{a \in A} \alpha(a)y(a) \geq 1, \\
 (2.6) \quad & g(a), \quad y(a) \geq 0 \quad (a \in A),
 \end{aligned}$$

where $\partial^+ a$ and $\partial^- a$ represent the initial and the terminal end-vertex of a , respectively.

We introduce a parameter $z \geq 0$, and consider the following parametric problem (P_z) for network $N_z = (G = (V, A), c, \alpha, \beta, z, s, t)$.

$$\begin{aligned}
 (P_z) \quad & \text{Maximize} \quad \varphi(a_0) \\
 & \text{subject to (2.1) and (2.2),} \\
 (2.7) \quad & \varphi(a) \leq \alpha(a)z + \beta(a) \quad (a \in A).
 \end{aligned}$$

Let $\varphi^*(\cdot, z)$ be the maximum flow of (P_z) in N_z . Then the maximum flow value $\varphi^*(a_0, z)$ is a monotone non-decreasing, continuous, piecewise linear and concave function of z . The following lemma establishes a connection between problems (P) and (P_z).

Lemma 2.1. Let z^* be the maximum value of z satisfying $\varphi^*(a_0, z) = z$. Then $\varphi^*(\cdot, z^*)$ is a maximum balanced flow in network N . □

Let $c(S)$ denote the capacity of an s - t cut S , where S is a subset of V such that $s \in S$ and $t \in \bar{S} \equiv V - S$. The value of $c(S)$ is defined by

$$(2.8) \quad c(S) = c(\Delta^+(S)) = \sum_{a \in \Delta^+(S)} c(a)$$

where $\Delta^+(S) = \{a \in A \mid \partial^+ a \in S, \partial^- a \in \bar{S}\}$. For problem (P_z) the capacity of each arc $a \in A$ is redefined by

$$(2.9) \quad c(a, z) = \min(c(a), \alpha(a)z + \beta(a)),$$

and the capacity of an s - t cut S is denoted by $c(S, z)$. Then, by the max-flow min-cut theorem [4], we have

$$(2.10) \quad \varphi^*(a_0, z) = \min\{c(S, z) \mid S \text{ is an } s\text{-}t \text{ cut}\}.$$

For an s - t cut S , let

$$(2.11) \quad z(S) = \max\{z \mid z \leq c(S, z)\},$$

then from Lemma 2.1 and equation (2.10), we have

$$(2.12) \quad \varphi^*(a_0) = z^* = \min\{z(S) \mid S \text{ is an } s\text{-}t \text{ cut}\}.$$

The algorithm proposed in the next section is to find z^* satisfying (2.12) by making use of the upper-bound simplex method [2, Chapters 18 and 19].

3. Algorithm for the Maximum Balanced Flow Problem

3.1 Basis structure

Using the upper-bound simplex method [2, Chapters 18 and 19], a basis B for problem (P_z) is a maximal set of linearly independent column vectors of D . The variables associated with the columns of B are considered to be basic variables and all others are nonbasic variables. A basic solution results from assigning each nonbasic variable a value equal to its lower or upper bound. Thereupon, a unique value results for each basic variable from (2.1). It is well known that the set of columns of D indexed by $T \subseteq A \cup \{a_0\}$ is a basis for D if and only if T is a spanning tree of G_0 . Let L and U , respectively, represent the sets of arcs corresponding to the nonbasic variables equal to their lower and upper bounds. We refer to (T, L, U) as a *basis structure*, and also call T a *basis*. In the following discussion, we always assume that $T, L, U \subseteq A$.

An important property of a spanning tree of G is that there is a unique path from any vertex of G to any other using only arcs of the tree. The set $\bar{T} \equiv A - T$ is called the *cotree* associated with T . For each $a \in \bar{T}$ we define $C(T, a)$ to be the subset of A consisting of arc a and the arcs of the path from $\partial^- a$ to $\partial^+ a$ in T . $C(T, a)$ is called the *fundamental circuit* associated with T and a . We partition $C(T, a)$ into two sets $C^+(T, a)$ and $C^-(T, a)$ with respect to a . $C^+(T, a)$ is the set of arcs in $C(T, a)$ with the same orientation as a in the cycle formed by $C(T, a)$, and $C^-(T, a) = C(T, a) - C^+(T, a)$. For each arc $a \in T$ we define a subset $S(T, a)$ of V by

$$(3.1) \quad S(T, a) = \{v \in V \mid \text{the path from } t \text{ to } v \text{ in } T \text{ does not contain } a\},$$

and a subset $K(\bar{T}, a)$ of A by

$$(3.2) \quad K(\bar{T}, a) = \Delta^+(S(T, a)) \cup \Delta^-(S(T, a)),$$

where $\Delta^-(S(T, a))$ represents the set of arcs entering $S(T, a)$. $K(\bar{T}, a)$ is called the *fundamental cut set* associated with \bar{T} and $a \in T$. If $\partial^+ a \in S(T, a)$, we define $K^+(\bar{T}, a) = \Delta^+(S(T, a))$ and $K^-(\bar{T}, a) = \Delta^-(S(T, a))$; otherwise $K^+(\bar{T}, a) = \Delta^-(S(T, a))$ and $K^-(\bar{T}, a) = \Delta^+(S(T, a))$. $K^+(\bar{T}, a)$ is the set of arcs with the same orientation as a in $K(\bar{T}, a)$, and $K^-(\bar{T}, a) = K(\bar{T}, a) - K^+(\bar{T}, a)$.

We say that arc a is *directed toward* t in T if $\partial^- a \in S(T, a)$; otherwise we say that a is *directed away from* t in T (see [1]). For spanning tree T of G , define function $\lambda_T : T \rightarrow \{1, -1\}$ by

$$(3.3) \quad \lambda_T(a) = \begin{cases} 1 & \text{if } a \text{ is directed toward } t \text{ in } T, \\ -1 & \text{if } a \text{ is directed away from } t \text{ in } T. \end{cases}$$

Given a basis structure (T, L, U) , let $\varphi(a_0) = z$. Then for each $a \in T$ arc flow-value $\varphi(a, z)$ of a is determined by

$$(3.4) \quad \varphi(a, z) = \begin{cases} \lambda_T(a)z + c(U \cap K^-(\bar{T}, a), z) - c(U \cap K^+(\bar{T}, a), z) & \text{if } s \notin S(T, a), \\ c(U \cap K^-(\bar{T}, a), z) - c(U \cap K^+(\bar{T}, a), z) & \text{otherwise.} \end{cases}$$

We call basis T a *feasible basis* of (P) if there exists a basis structure (T, L, U) such that $\varphi(a, z)$ ($a \in T$) defined by (3.4) satisfy (2.2) and (2.3) with $\varphi(a_0)$ replaced by z . Clearly, $(T, A - T, \emptyset)$ is a feasible basis structure of (P) for $\varphi(a_0) = z = 0$. We denote by $\bar{z}(T)$ and $\underline{z}(T)$ the maximum and the minimum value of z such that T is a feasible basis of (P) .

3.2 Algorithm

It is easy to see that, if the arc flow-value of each arc in U is maintained to be equal to the changing upper bound as z increases, then the arc flow-value of each basic arc changes in a unique way as follows. Let

$$(3.5) \quad U_\alpha(z) = \{ a \in U \mid \alpha(a)z + \beta(a) < c(a) \},$$

$$(3.6) \quad U_c(z) = U - U_\alpha(z),$$

and for each $a \in T$ let

$$(3.7) \quad \eta(a) = \begin{cases} \lambda_T(a) + \alpha(U_\alpha(z) \cap K^-(\bar{T}, a)) - \alpha(U_\alpha(z) \cap K^+(\bar{T}, a)) & \text{if } s \notin S(T, a), \\ \alpha(U_\alpha(z) \cap K^-(\bar{T}, a)) - \alpha(U_\alpha(z) \cap K^+(\bar{T}, a)) & \text{otherwise,} \end{cases}$$

$$(3.8) \quad \xi(a) = \beta(U_\alpha(z) \cap K^-(\bar{T}, a)) - \beta(U_\alpha(z) \cap K^+(\bar{T}, a)) + c(U_c(z) \cap K^-(\bar{T}, a)) - c(U_c(z) \cap K^+(\bar{T}, a)).$$

Then arc flow-value $\varphi(a, z)$ of $a \in T$ can be written as

$$(3.9) \quad \varphi(a, z) = \eta(a)z + \xi(a).$$

An efficient method to determine $\eta(a)$ and $\xi(a)$ is given in the description of the algorithm. For each $a \in T$ let

$$(3.10) \quad \bar{z}(a) = \max\{ z \mid 0 \leq \varphi(a, z) \leq c(a, z) \},$$

$$(3.11) \quad \underline{z}(a) = \min\{ z \mid 0 \leq \varphi(a, z) \leq c(a, z) \},$$

$$(3.12a) \quad \bar{z}^1(a) = \begin{cases} -\xi(a)/\eta(a) & \text{if } \eta(a) < 0 \\ \infty & \text{otherwise,} \end{cases}$$

$$(3.12b) \quad \bar{z}^2(a) = \begin{cases} (c(a) - \xi(a))/\eta(a) & \text{if } \eta(a) > 0 \\ \infty & \text{otherwise,} \end{cases}$$

$$(3.12c) \quad \bar{z}^3(a) = \begin{cases} (\beta(a) - \xi(a))/(\eta(a) - \alpha(a)) & \text{if } \eta(a) - \alpha(a) > 0 \\ \infty & \text{otherwise,} \end{cases}$$

$$(3.13a) \quad \underline{z}^1(a) = \begin{cases} -\xi(a)/\eta(a) & \text{if } \eta(a) > 0 \\ -\infty & \text{otherwise,} \end{cases}$$

$$(3.13b) \quad \underline{z}^2(a) = \begin{cases} (c(a) - \xi(a))/\eta(a) & \text{if } \eta(a) < 0 \\ -\infty & \text{otherwise,} \end{cases}$$

$$(3.13c) \quad \underline{z}^3(a) = \begin{cases} (\beta(a) - \xi(a))/(\eta(a) - \alpha(a)) & \text{if } \eta(a) - \alpha(a) < 0 \\ -\infty & \text{otherwise.} \end{cases}$$

Then we have

$$(3.14) \quad \bar{z}(a) = \min\{\bar{z}^1(a), \bar{z}^2(a), \bar{z}^3(a)\},$$

$$(3.15) \quad \underline{z}(a) = \max\{\underline{z}^1(a), \underline{z}^2(a), \underline{z}^3(a)\}$$

$$(3.16) \quad \bar{z}(T) = \min_{a \in T} \bar{z}(a)$$

$$(3.17) \quad \underline{z}(T) = \max\{\max_{a \in T} \underline{z}(a), 0\}.$$

When $z = \bar{z}(T)$, the flow value of an arc, say a_q , of T reaches either zero (its lower bound) or its upper bound. The simplex pivot step is performed to obtain an alternate basis structure which gives a new feasible basis when $z = \bar{z}(T)$. The arc a_q is removed from the basis and an eligible nonbasic arc on $K(\bar{T}, a_q)$, say a_e , is selected to enter the basis. This process is repeated until no nonbasic arc is eligible to enter the basis, i.e., the optimal value z^* of z in (2.12) is reached. A detailed description of the algorithm is given below.

Step 0: Let T be a spanning tree of G not containing a_0 . Set $z^* = 0$; $L = A - T$; $U = \emptyset$;
 $\xi(a) = 0$;

$$\eta(a) = \begin{cases} 1 & \text{if } a \in C^+(T, a_0) \\ -1 & \text{if } a \in C^-(T, a_0) \\ 0 & \text{otherwise} \end{cases} \quad (a \in A).$$

Step 1: For each $a \in T$ compute $\bar{z}^1(a)$, $\bar{z}^2(a)$ and $\bar{z}^3(a)$ by (3.12) and for each $a \in U_\alpha(z^*)$ put

$$z^\alpha(a) = \frac{c(a) - \beta(a)}{\alpha(a)}.$$

Let

$$\bar{z}(T) = \min_{a \in T} \{\bar{z}^1(a), \bar{z}^2(a), \bar{z}^3(a)\},$$

$$z^\alpha = \min\{z^\alpha(a) \mid a \in U_\alpha(z^*)\}.$$

If $\bar{z}(T) \geq z^\alpha$, go to Step 2. Otherwise go to Step 3.

Step 2: Identify an arc $a_q \in U_\alpha(z^*)$ for which $z^\alpha = z^\alpha(a_q)$. Change $\eta(a)$ and $\xi(a)$ along circuit $C(T, a_q)$ by

$$\eta(a) := \begin{cases} \eta(a) - \alpha(a_q) & \text{if } a \in C^+(T, a_q) \\ \eta(a) + \alpha(a_q) & \text{if } a \in C^-(T, a_q) \\ \eta(a) & \text{otherwise,} \end{cases}$$

$$\xi(a) := \begin{cases} \xi(a) - (\beta(a_q) - c(a_q)) & \text{if } a \in C^+(T, a_q) \\ \xi(a) + (\beta(a_q) - c(a_q)) & \text{if } a \in C^-(T, a_q) \\ \xi(a) & \text{otherwise.} \end{cases}$$

Set $z^* = z^\alpha$ and go to Step 1.

Step 3: Set $z^* = \bar{z}(T)$ and identify an arc $a_q \in T$ for which $z^* = \min\{\bar{z}^1(a_q), \bar{z}^2(a_q), \bar{z}^3(a_q)\}$.

If $\bar{z}(T) = \bar{z}^2(a_q)$ (or $\bar{z}^3(a_q)$), put

$$\begin{aligned} \Delta\bar{\eta}(a_q) &= \eta(a_q) \quad (\text{or } \eta(a_q) - \alpha(a_q)), \\ \Delta\bar{\xi}(a_q) &= \xi(a_q) - c(a_q) \quad (\text{or } \xi(a_q) - \beta(a_q)), \\ E_q &= (U \cap K^-(\bar{T}, a_q)) \cup (L \cap K^+(\bar{T}, a_q)). \end{aligned} \tag{3.18}$$

Otherwise put

$$\begin{aligned} \Delta\bar{\eta}(a_q) &= \eta(a_q), \quad \Delta\bar{\xi}(a_q) = \xi(a_q), \\ E_q &= (U \cap K^+(\bar{T}, a_q)) \cup (L \cap K^-(\bar{T}, a_q)). \end{aligned} \tag{3.19}$$

If E_q is empty, go to Step 4. Otherwise select an arc a_e from E_q , and change $\eta(a)$ and $\xi(a)$ along $C(T, a_e)$ as follows: If $a_q \in C^-(T, a_e)$, put

$$\eta(a) := \begin{cases} \eta(a) + \Delta\bar{\eta}(a_q) & \text{if } a \in C^+(T, a_e) \\ \eta(a) - \Delta\bar{\eta}(a_q) & \text{if } a \in C^-(T, a_e) \\ \eta(a) & \text{otherwise,} \end{cases} \tag{3.20}$$

$$\xi(a) := \begin{cases} \xi(a) + \Delta\bar{\xi}(a_q) & \text{if } a \in C^+(T, a_e) \\ \xi(a) - \Delta\bar{\xi}(a_q) & \text{if } a \in C^-(T, a_e) \\ \xi(a) & \text{otherwise,} \end{cases} \tag{3.21}$$

and if $a_q \in C^+(T, a)$, put

$$\eta(a) := \begin{cases} \eta(a) - \Delta\bar{\eta}(a_q) & \text{if } a \in C^+(T, a_e) \\ \eta(a) + \Delta\bar{\eta}(a_q) & \text{if } a \in C^-(T, a_e) \\ \eta(a) & \text{otherwise,} \end{cases} \tag{3.22}$$

$$\xi(a) := \begin{cases} \xi(a) - \Delta\bar{\xi}(a_q) & \text{if } a \in C^+(T, a_e) \\ \xi(a) + \Delta\bar{\xi}(a_q) & \text{if } a \in C^-(T, a_e) \\ \xi(a) & \text{otherwise.} \end{cases} \tag{3.23}$$

Remove the arc a_q from tree T , take the arc a_e into T , and go to Step 1.

Step 4: The flow $\varphi(a, z^*) = \eta(a)z^* + \xi(a)$ ($a \in A$) is the maximum balanced flow with the maximum flow value z^* . Stop.

We denote by SA the above algorithm. Let $T' = (T \cup \{a_e\}) - \{a_q\}$ be the next basis generated by algorithm SA, and $\varphi'(a, z) = \eta'(a)z + \xi'(a)$ ($a \in A$) be the flow associated with basis T' , then $\varphi(a, z) = \varphi'(a, z)$ at $z = \bar{z}(T)$. It follows that T' is also a feasible basis at $z = \bar{z}(T)$ and the value of z^* is not decreased in each iteration of Step 3. Note that the number of iterations of Step 2 is at most $m = |A|$. It is clear from the following theorem that the algorithm generates a maximum balanced flow in finite steps if the value of z^* strictly increases in each iteration of Step 3.

Theorem 3.1. If E_q defined by (3.18) and (3.19) is empty, then

- (i) $K(\bar{T}, a_q)$ is an s - t cut set,
- (ii) a_q is directed toward t in T if $\varphi(a_q, \bar{z}(T)) = c(a_q, \bar{z}(T))$ and is directed away from t in T if $\varphi(a_q, \bar{z}(T)) = 0$,
- (iii) $\bar{z}(T)$ is equal to the maximum balanced flow value.

Proof: To prove (i), we suppose on the contrary that $K(\bar{T}, a_q)$ is not an s - t cut set. Then from (3.7) and the definition of E_q we have $\eta(a_q) \geq 0$ if $\bar{z}(T) = \bar{z}^1(a_q)$ and $\eta(a_q) \leq 0$ if $\bar{z}(T) = \bar{z}^2(a_q)$ or $\bar{z}^3(a_q)$. It follows from (3.12) that $\bar{z}(T) = \infty$, a contradiction. Similarly as the above argument, It is easy to show that (ii) is also true.

Next, we prove (iii). Since $\varphi(a, z)$ ($a \in A$) is a balanced flow for $z = \bar{z}(T)$, we see that for any s - t cut S , $\bar{z}(T) \leq c(S, \bar{z}(T))$. Furthermore, from (i), (ii), (3.4) and the definitions of E_q and $\bar{z}(T)$ we have

$$\bar{z}(T) = z(\bar{S}(t, a_q)) = \max\{z \mid z \leq c(\bar{S}(T, a_q), z)\}.$$

It follows from (2.12) that z^* generated by algorithm SA is equal to the maximum balanced flow value. □

For each $v \in V$ let $p(v) = 0$ if $v \in S(T, a_q)$ and $p(v) = \lambda_T(a_q)/\Delta\bar{\eta}(a_q)$ if $v \in \bar{S}(T, a_q)$. Then p satisfies complementary slackness together with $\varphi(a) = \eta(a)\bar{z}(T) + \xi(a)$ ($a \in A$) and $\varphi(a_0) = \bar{z}(T)$. Hence the algorithm can be regarded as a primal simplex method.

In the next section, we describe the notion of strongly feasible basis and the rule of removing a basic arc, and give a proof of the finiteness of the present algorithm.

4. Strongly Feasible Bases

To introduce the notion of strongly feasible basis and to describe the rule of removing a basic arc which ensures the finiteness of the algorithm, we need a few further preliminaries.

Let (T, L, U) be a feasible basis structure of (P), and let $\varphi(a) = \eta(a)z + \xi(a)$ be a flow associated with (T, L, U) . If $\bar{z}(a) < \infty$ and $\underline{z}(a) > -\infty$ in (3.14) and (3.15), we define

$$(4.1) \quad \Delta\bar{\eta}(a) = \begin{cases} \eta(a) & \text{if } \bar{z}(a) = \bar{z}^1(a) \text{ or } \bar{z}^2(a), \\ \eta(a) - \alpha(a) & \text{if } \bar{z}(a) = \bar{z}^3(a), \end{cases}$$

$$(4.2) \quad \Delta\bar{\xi}(a) = \begin{cases} \xi(a) & \text{if } \bar{z}(a) = \bar{z}^1(a), \\ \xi(a) - c(a) & \text{if } \bar{z}(a) = \bar{z}^2(a), \\ \xi(a) - \beta(a) & \text{if } \bar{z}(a) = \bar{z}^3(a), \end{cases}$$

$$(4.3) \quad \Delta\underline{\eta}(a) = \begin{cases} \eta(a) & \text{if } \underline{z}(a) = \underline{z}^1(a) \text{ or } \underline{z}^2(a), \\ \eta(a) - \alpha(a) & \text{if } \underline{z}(a) = \underline{z}^3(a), \end{cases}$$

$$(4.4) \quad \Delta\underline{\xi}(a) = \begin{cases} \xi(a) & \text{if } \underline{z}(a) = \underline{z}^1(a), \\ \xi(a) - c(a) & \text{if } \underline{z}(a) = \underline{z}^2(a), \\ \xi(a) - \beta(a) & \text{if } \underline{z}(a) = \underline{z}^3(a). \end{cases}$$

Note that $\bar{z}(a) = -\Delta\bar{\xi}(a)/\Delta\bar{\eta}(a)$, $\underline{z}(a) = -\Delta\underline{\xi}(a)/\Delta\underline{\eta}(a)$. For each $a \in T$, define an $(n - 1)$ -dimensional vector $x(T, a)$ ($n = |V|$) by

$$(4.5) \quad x_i(T, a) = \begin{cases} 0 & \text{if } v_i \in S(T, a) \\ 1 & \text{otherwise} \end{cases} \quad (i = 1, 2, \dots, n - 1),$$

where $v_1 = s$ and $v_n = t$, and define n -dimensional vectors $\bar{\pi}(T, a)$ and $\underline{\pi}(T, a)$ by

$$(4.6) \quad \bar{\pi}(T, a) = (\bar{z}(a), \lambda_T(a)x(T, a)/\Delta\bar{\eta}(a)),$$

$$(4.7) \quad \underline{\pi}(T, a) = (\underline{z}(a), \lambda_T(a)x(T, a)/\Delta\underline{\eta}(a)).$$

Let $\bar{\pi}(T)$ be the lexicographic minimum of $\bar{\pi}(T, a)$ ($a \in T$), and $\underline{\pi}(T)$ be the lexicographic maximum of $\max\{\underline{\pi}(T, a), 0\}$ ($a \in T$):

$$(4.8) \quad \bar{\pi}(T) = \text{lex} \min_{a \in T} \bar{\pi}(T, a),$$

$$(4.9) \quad \underline{\pi}(T) = \text{lex} \max_{a \in T} \max\{\underline{\pi}(T, a), 0\}.$$

A basis structure (T, L, U) is called a *strongly feasible basis structure* if the following

(i) and (ii) hold.

(i) $\bar{\pi}(T) \succ \underline{\pi}(T)$.

(ii) For any $a \in T$, if $\eta(a) = \xi(a) = 0$, i.e., $\varphi(a, z) = 0$, then a is directed away from t , and if (1) $\eta(a) = \alpha(a)$ and $\xi(a) = \beta(a)$ or (2) $\eta(a) = 0$ and $\xi(a) = c(a)$ then a is directed toward t in T .

Here the symbol ' \succ ' stands for 'lexicographically greater than'.

We refine algorithm SA by initiating it with a strongly feasible basis structure and by making a specific selection of $a \in T$ in Step 3, i.e., by selecting arc a_q such that $\bar{\pi}(T, a_q) = \bar{\pi}(T)$. We denote by SSA the algorithm SA with these two refinements. Let

$T_0 \subseteq A$ be a directed spanning tree of G with source s as its root. Here, we assume without loss of generality that there exists such a directed spanning tree of G . Then it is easy to see that $(T_0, A - T_0, \emptyset)$ is a strongly feasible basis structure of (P).

Let T be a strongly feasible basis of problem (P), $T' = (T \cup \{a_e\}) - \{a_q\}$ be the next feasible basis generated by SSA, and $\varphi(a, z) = \eta(a)z + \xi(a)$ ($\varphi'(a, z) = \eta'(a)z + \xi'(a)$) ($a \in A$) be the flows associated with T (T'). Then we have:

Lemma 4.1. T' is a strongly feasible basis and we have $\bar{\pi}(T) < \bar{\pi}(T')$.

Proof: To prove (i) in the definition of strongly feasible basis, we show that for any $a \in T'$, if $\bar{z}'(a) = \bar{z}(a_q)$, then (4.10) holds and if $\underline{z}'(a) = \bar{z}(a_q)$, then (4.11) holds:

$$(4.10) \quad \frac{\lambda_{T'}(a)x(T', a)}{\Delta\bar{\eta}'(a)} > \frac{\lambda_T(a_q)x(T, a_q)}{\Delta\bar{\eta}(a_q)},$$

$$(4.11) \quad \frac{\lambda_{T'}(a)x(T', a)}{\Delta\underline{\eta}'(a)} < \frac{\lambda_T(a_q)x(T, a_q)}{\Delta\bar{\eta}(a_q)}.$$

Since $\eta(a) = \eta'(a)$ and $\xi(a) = \xi'(a)$, i.e., $\varphi(a, z) = \varphi'(a, z)$ for each $a \in T - C(T, a_e)$, we consider the arcs in $C(T, a_e)$ alone. From (3.20) ~ (3.23), we see that for each $a \in C(T, a_e)$, if the orientation of arc a is opposite to that of a_q in $C(T, a_e)$, then

$$(4.12) \quad \eta'(a) = \eta(a) + \Delta\bar{\eta}(a_q), \quad \xi'(a) = \xi(a) + \Delta\bar{\xi}(a_q),$$

$$(4.13) \quad \lambda_{T'}(a)x(T', a) = \lambda_T(a)x(T, a) + \lambda_T(a_q)x(T, a_q),$$

and if the orientation of arc a is the same as that of a_q in $C(T, a_e)$, then

$$(4.14) \quad \eta'(a) = \eta(a) - \Delta\bar{\eta}(a_q), \quad \xi'(a) = \xi(a) - \Delta\bar{\xi}(a_q),$$

$$(4.15) \quad \lambda_{T'}(a)x(T', a) = \lambda_T(a)x(T, a) - \lambda_T(a_q)x(T, a_q).$$

We prove that (4.10) holds if $\bar{z}'(a) = \bar{z}(a_q)$. Without loss of generality, suppose that

$$(4.16) \quad \bar{z}'(a) = \frac{\beta(a) - \xi'(a)}{\eta'(a) - \alpha(a)}.$$

Then from (4.12) and (4.14) we see that, if $\eta(a) = \alpha(a)$, then $\beta(a) = \xi(a)$, if $\eta(a) - \alpha(a) > 0$, then

$$(4.17) \quad \bar{z}(a) = \frac{\beta(a) - \xi(a)}{\eta(a) - \alpha(a)} = \bar{z}(a_q),$$

and if $\eta(a) - \alpha(a) < 0$, then

$$(4.18) \quad \underline{z}(a) = \frac{\beta(a) - \xi(a)}{\eta(a) - \alpha(a)} = \bar{z}(a_q).$$

It follows from the strong feasibility of T , the definition of a_q and (4.12) ~ (4.15) that (4.10) holds for each $a \in T$. Similarly as the above argument, we can prove that (4.11) holds if $\underline{z}'(a) = \bar{z}(a_q)$

Next we prove (ii) in the definition of strongly feasible basis for T' . We show that for any $a \in C(T, a_e)$ such that $\eta'(a) = \alpha(a)$ and $\xi'(a) = \beta(a)$, a is directed toward t in T' . From (4.12) and (4.14) we have $\eta(a) - \alpha(a) = -\Delta\bar{\eta}(a_q)$ and $\beta(a) - \xi(a) = \Delta\bar{\xi}(a_q)$ if the orientation of arc a is opposite to that of a_q , and we have $\eta(a) - \alpha(a) = \Delta\bar{\eta}(a_q)$ and $\beta(a) - \xi(a) = -\Delta\bar{\xi}(a_q)$ if the orientation of arc a is the same as that of a_q . This implies that

$$(4.19) \quad \frac{\beta(a) - \xi(a)}{\eta(a) - \alpha(a)} = \frac{-\Delta\bar{\xi}(a_q)}{\Delta\bar{\eta}(a_q)} = \bar{z}(a_q).$$

Therefore, from the strong feasibility of T , the definition of a_q and (4.12) ~ (4.15), we see that $\lambda_{T'}(a)x(T', a) > 0$, i.e., a is directed toward t in T' . Similarly, we can prove that if $\eta'(a) = 0$ and $\xi'(a) = c(a)$, then a is directed toward t , and if $\eta'(a) = \xi'(a) = 0$, then a is directed away from t in T' . Thus, we see that T' is also a strongly feasible basis and $\underline{\pi}(T') < \bar{\pi}(T) < \bar{\pi}(T')$. □

From Lemma 4.1 we have:

Theorem 4.1. The algorithm SSA for the maximum balanced flow problem terminates after a finite number of steps. □

5. The Maximum Balanced Integral Flow Problem

In this section, we consider the following maximum balanced integral flow problem (\tilde{P}) for network $N = (G = (V, A), c, \alpha, \beta, s, t)$ with integral capacity function c and integral function β .

$$(\tilde{P}) \quad \begin{array}{l} \text{Maximize } \varphi(a_0) \\ \text{subject to (2.1) and (2.2)} \end{array}$$

$$(5.1) \quad \varphi(a) \leq [\alpha(a)\varphi(a_0)] + \beta(a) \quad (a \in A),$$

$$(5.2) \quad \varphi(a) \in I \quad (a \in A),$$

where $[r]$ represents the greatest integer less than or equal to real number r . A function $\varphi : A \rightarrow I$ satisfying (2.1), (2.2) and (5.1) is called a *balanced integral flow*.

Until now, no polynomial method is known for solving this problem, and U. Zimmermann [10] conjectures that the problem is NP-hard. Here, we point out that the maximum balanced integral flow problem can be solved in polynomial time for a special case of $\alpha(a) = r$ ($a \in A$) for a fixed real r .

We consider the following two problems (\tilde{P}_z) and (P_z) for network N_z .

$$\begin{aligned}
 (5.3) \quad (\tilde{P}_z) \quad & \text{Maximize } \varphi(a_0) \\
 & \text{subject to (2.1), (2.2) and (5.2)} \\
 & \varphi(a) \leq \lfloor rz \rfloor + \beta(a) \quad (a \in A),
 \end{aligned}$$

$$\begin{aligned}
 (5.4) \quad (P_z) \quad & \text{Maximize } \varphi(a_0) \\
 & \text{subject to (2.1) and (2.2)} \\
 & \varphi(a) \leq rz + \beta(a) \quad (a \in A).
 \end{aligned}$$

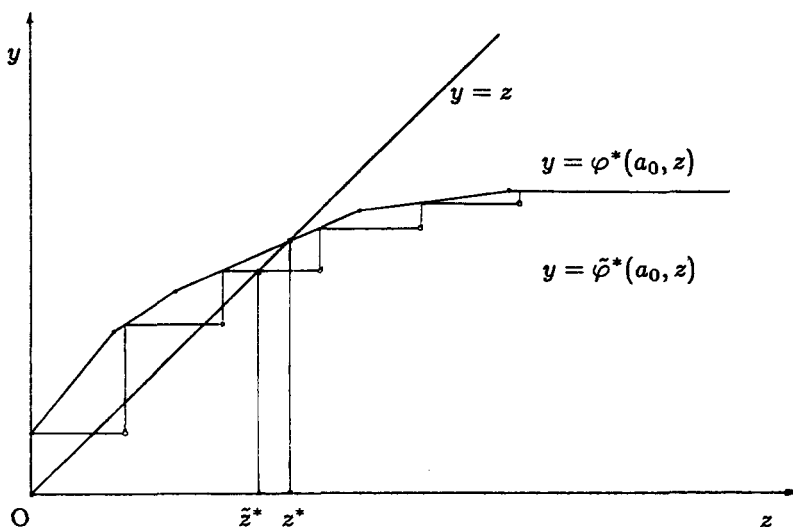


Fig. 1

Let $\tilde{\varphi}^*(\cdot, z)$ and $\varphi^*(\cdot, z)$ denote the maximum flow of problems (\tilde{P}_z) and (P_z) in N_z , respectively. Then we have

$$(5.5) \quad \tilde{\varphi}^*(a_0, z) = \varphi^*(a_0, k/r)$$

for all $z \in [\frac{k}{r}, \frac{k+1}{r})$ ($k = 0, 1, \dots$). For problems (\tilde{P}) and (\tilde{P}_z) , let \tilde{z}^* be the maximum value of z such that $\tilde{\varphi}^*(a_0, z) = z$, then \tilde{z}^* is equal to the maximum balanced integral flow value. It is easy to see from (5.5) that

$$(5.6) \quad \tilde{z}^* = \varphi^*(a_0, \lfloor rz^* \rfloor / r),$$

where z^* is defined by Lemma 2.1 (see Fig. 1). Such z^* can be found in $O(\min(m, \lfloor 1/r \rfloor)T(n, m))$ time by Nakayama's algorithm [9], where $T(m, n)$ is the time required for finding a maximum flow in a network with n vertices and m arcs. Therefore, the maximum

balanced integral flow problem with a constant balancing rate function can be solved in $O(\min(m, \lceil 1/r \rceil)T(n, m))$ time.

Acknowledgement: The author would like to express his gratitude to Professor Satoru Fujishige of the University of Tsukuba for his helpful advice and comments.

References

- [1] Cunningham, W., H.: A network simplex method. *Mathematical Programming*, Vol. 11 (1976), 105-116.
- [2] Dantzig, G.B.: *Linear Programming and Extensions*. Princeton University Press, Princeton, N.J., 1963.
- [3] Elam, J., Glover, F. and Klingman, D.: A strongly convergent primal simplex algorithm for generalized networks. *Mathematics of Operations Research*, Vol. 4 (1979), 39-59.
- [4] Ford, L. R., Jr. and Fulkerson, D. R.: *Flows in Networks*. Princeton University Press, Princeton, N.J., 1962.
- [5] Fujishige, S., Nakayama, A. and Cui, W.-T.: On the equivalence of the maximum balanced flow problem and the weighted minimax flow problem. *Operations Research Letters*, Vol. 5 (1986), 207-209.
- [6] Ichimori, T., Ishii, H. and Nishida, T.: Weighted minimax real-valued flows. *Journal of the Operations Research Society of Japan*, Vol. 24 (1981), 52-60.
- [7] Kennington, J. L. and Helgason, R. V.: *Algorithms for Network Programming*. Wiley, New York, 1980.
- [8] 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.
- [9] 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*, Vol. 29 (1986), 400-409.
- [10] Zimmermann, U.: Duality for balanced submodular flows. Preprint No.89, Fachbereich Mathematik, Universität Kaiserslautern, 1985.

Wentian CUI: Institute of Socio-Economic
Planning, University of Tsukuba, Tsukuba,
Ibaraki 305, Japan