No. **632**

A Push/Relabel Framework for Submodular Flows and Its Refinement for 0-1 Submodular Flows

by

Satoru Fujishige and Xiaodong Zhang

June 1995

A Push/Relabel Framework for Submodular Flows and Its Refinement for 0-1 Submodular Flows

Satoru Fujishige[1] and Xiaodong Zhang[2]

June 1995

Abstract. We consider the submodular flow problem of Edmonds and Giles. A submodular flow is a flow in a network satisfying capacity constraints and flow-boundary constraints given in terms of the base polyhedron of a submodular system. A cost scaling framework is constructed by using ε-optimality concept associated with dual variables of a flow, originally due to Tardos and Bertsekas. The framework is a generalization of Goldberg and Tarjan's push/relabel algorithm for minimum-cost flows and also a generalization of Fujishige and Zhang's algorithm for the submodular intersection problem. Each phase of the cost scaling, called procedure Refine, improves a 2ε-optimal submodular flow to an ε-optimal submodular flow. Furthermore, we devise a faster hybrid algorithm of procedure Refine for the 0-1 submodular flow problem which is a natural generalization of Fujishige and Zhang's algorithm for the independent assignment problem. For a network with n vertices, m arcs and integer arc costs bounded by Γ, an optimal 0-1 submodular flow can be found in $O(\sqrt{m}n^2 \log(n\Gamma))$ time by our algorithm under oracles for the dependence function and the exchange capacity of the given submodular system.

1. Introduction

In this paper we consider the submodular flow problem of J. Edmonds and R. Giles [4] and construct a cost scaling framework for the problem as a generalization of the algorithm for the minimum-cost flow problem devised by A.V. Goldberg and R. E. Tarjan [17]. Furthermore, we apply this framework to the 0-1 submodular flow problem and give a hybrid-version algorithm for it.

The submodular flow problem includes as special cases many combinatorial optimization problems such as the ordinary minimum-cost flow problem, directed cut covering problem [19, 5], the orientation problem [5, 6], the dijoin problem [5, 7] and the intersection problem of two submodular systems [3, 9] (also see [11]). Polynomial and strongly polynomial time algorithms for the submodular flow problem

[1]Institute of Socio-Economic Planning, University of Tsukuba, Tsukuba, Ibaraki 305, Japan.
[2]Computer Science Program, University of Texas at Dallas, Richardson, Texas 75083-0688 USA.

have been presented in [2, 8, 12, 21, 23] (also see [11]). Efficient algorithms for 0-1 submodular flows were proposed by A. Frank [6] and H. N. Gabow [15]. Recently, a cost-splitting algorithm for 0-1 submodular flows has also been given by M. Shigeno and S. Iwata [20].

2. Definitions and Preliminaries

We give some definitions and basic preliminary results related to submodular systems (also see [11]). Let V be a nonempty finite set and $\mathcal{D}$ be a collection of subsets of V which forms a distributive lattice with set union $\cup$ and intersection $\cap$ as the lattice operations, join and meet, i.e., for each $X,\ Y \in \mathcal{D}$ we have $X \cup Y,\ X \cap Y \in \mathcal{D}$. Let $\mathbf{R}$ be the set of reals and $f : \mathcal{D} \to \mathbf{R}$ be a *submodular function* on the distributive lattice $\mathcal{D}$, i.e.,

$$\forall X,\ Y \in \mathcal{D}:\ f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y). \tag{2.1}$$

If $\emptyset,\ V \in \mathcal{D}$ and $f(\emptyset) = 0$, we call the pair $(\mathcal{D}, f)$ a *submodular system* on V. Function f is called the *rank function* of $(\mathcal{D}, f)$.

Define a polyhedron $\mathrm{B}(f)$ by

$$\mathrm{B}(f) = \{x \mid x \in \mathbf{R}^V,\ x(V) = f(V),\ \forall X \in \mathcal{D}:\ x(X) \leq f(X)\}, \tag{2.2}$$

where $\mathbf{R}^V = \{x \mid x : V \to \mathbf{R}\}$, $x(X) = \sum_{e \in X} x(e)$ for each $X \in \mathcal{D}$ and $x(\emptyset) = 0$. We call $\mathrm{B}(f)$ the *base polyhedron* associated with submodular system $(\mathcal{D}, f)$. Also a vector in the base polyhedron $\mathrm{B}(f)$ is called a *base* of $(\mathcal{D}, f)$.

For any base $x \in \mathrm{B}(f)$ and $v \in V$ define

$$\mathrm{dep}(x, v) = \bigcap \{X \mid v \in X \in \mathcal{D},\ x(X) = f(X)\}. \tag{2.3}$$

We call dep: $\mathrm{B}(f) \times V \to 2^V$ the *dependence function.*

Directly from the definition of the dependence function, we have

Lemma 2.1: *For a base $x \in \mathrm{B}(f)$ and $u,\ v,\ w \in V$, if $u \in \mathrm{dep}(x, v) - \{v\}$ and $v \in \mathrm{dep}(x, w) - \{w\}$, then we have $u \in \mathrm{dep}(x, w) - \{w\}$.* □

For any $x \in \mathrm{B}(f)$, $v \in V$ and $u \in \mathrm{dep}(x, v) - \{v\}$ the *exchange capacity* $\tilde{c}(x, v, u)$ is defined by

$$\tilde{c}(x, v, u) = \min\{f(X) - x(X) \mid v \in X \in \mathcal{D},\ u \notin X\}. \tag{2.4}$$

For a nonnegative α, we have $x + \alpha(\chi_v - \chi_u) \in \mathrm{B}(f)$ if and only if $0 \leq \alpha \leq \tilde{c}(x, v, u)$, where for any $s \in E$ $\chi_s \in \mathbf{R}^V$ is the unit vector defined by $\chi_s(s) = 1$ and $\chi_s(s') = 0$ for $s' \in V - \{s\}$.

The following lemma is obtained by a direct adaptation of the results shown in [9] for polymatroids (see [11] for a detailed proof).

Lemma 2.2: *For any $x \in \mathrm{B}(f)$ let u_i, v_i ($i = 1, 2, \cdots, q$) be $2q$ distinct elements of V such that*

$$u_i \in \mathrm{dep}(x, v_i) \quad (i = 1, 2, \cdots, q), \tag{2.5}$$

$$u_i \notin \mathrm{dep}(x, v_j) \quad (1 \leq i < j \leq q). \tag{2.6}$$

For any α_i ($i = 1, 2, \cdots, q$) satisfying $0 < \alpha_i \leq \tilde{c}(x, v_i, u_i)$ ($i = 1, 2, \cdots, q$) define a vector $y \in \mathbf{R}^V$ by

$$y = x + \sum_{i=1}^{q} \alpha_i (\chi_{v_i} - \chi_{u_i}). \tag{2.7}$$

Then,

$$y \in \mathrm{B}(f). \tag{2.8}$$

□

Lemma 2.3 ([11, p. 119]): *For an arbitrary $x \in \mathrm{B}(f)$, let u, $v \in E$, $0 < \alpha \leq \tilde{c}(x, v, u)$ and $y \in \mathrm{B}(f)$ be such that*

$$u \in \mathrm{dep}(x, v), \quad y = x + \alpha(\chi_v - \chi_u). \tag{2.9}$$

Suppose that there exist w, $s \in V$ such that

$$w \notin \mathrm{dep}(x, s), \quad w \in \mathrm{dep}(y, s). \tag{2.10}$$

Then we have

$$u \in \mathrm{dep}(x, s), \quad w \in \mathrm{dep}(x, v). \tag{2.11}$$

□

Next, we introduce a fundamental operation, called the contraction by a vector, on a submodular system $(\mathcal{D}, f)$. For a submodular system $(\mathcal{D}, f)$ and a vector $x \in \mathbf{R}^V$ such that there exists a base $z \in \mathrm{B}(f)$ satisfying $x \leq z$, define $f_x : 2^V \to \mathbf{R}$ by

$$f_x(X) = \min\{f(Z) - x(Z - X) \mid X \subseteq Z \in \mathcal{D}\} \tag{2.12}$$

for each $X \subseteq V$. The function f_x is a submodular function on 2^V. We call the submodular system $(2^V, f_x)$ the *contraction* of $(\mathcal{D}, f)$ by vector x. Define

$$\mathrm{B}(f)_x = \{y \mid y \in \mathrm{B}(f),\ y \geq x\}. \tag{2.13}$$

Then we can show that $\mathrm{B}(f_x) = \mathrm{B}(f)_x$. Therefore, we have

Lemma 2.4: *For each $z \in \mathrm{B}(f_x)$, we have $z - x \geq 0$.* □

Given a submodular system $(\mathcal{D}, f)$ on V and a weight function $w : V \to \mathbf{R}$, consider a linear optimization problem described as

$$(\mathrm{P}_s^1) \quad \text{Minimize} \quad \sum_{v \in V} w(v)x(v) \tag{2.14}$$

$$\text{subject to} \quad x \in \mathrm{B}(f). \tag{2.15}$$

Theorem 2.5 (see [11]): *A base $x \in \mathrm{B}(f)$ is an optimal solution of Problem (P_s^1) if and only if for each $u, v \in V$ such that $v \in \mathrm{dep}(x, u)$ we have $w(u) \geq w(v)$.* □

In our algorithms presented in the sequel we assume oracles for the dependence function and the exchange capacity associated with the given submodular system. Also, we assume that f is an integer-valued function.

3. The Submodular Flow Problem and the Optimality Condition

Let $G = (V, A)$ be a directed graph with a vertex set V ($|V| = n$) and an arc set A ($|A| = m$). Also let $\bar{c} : A \to \mathbf{Z}$ (the set of all integers) be an upper capacity function, $\underline{c} : A \to \mathbf{Z}$ be a lower capacity function and $\gamma : A \to \mathbf{Z}$ be a cost function. Let $(\mathcal{D}, f)$ be a submodular system on V with an integer-valued rank function f such that $f(V) = 0$. Denote the thus defined network by $\mathcal{N} = (G = (V, A), \underline{c}, \bar{c}, \gamma, (\mathcal{D}, f))$.

Define $\delta^+ v = \{a \mid a \in A, \partial^+ a = v\}$ and $\delta^- v = \{a \mid a \in A, \partial^- a = v\}$ for each $v \in V$, where, $\partial^+ a$ denotes the initial vertex (or tail) of an arc a and $\partial^- a$ denotes the terminal vertex (or head) of a. Let $\varphi : A \to \mathbf{R}$ be a flow in $\mathcal{N}$. The function $\partial\varphi : V \to \mathbf{R}$ defined by

$$\partial\varphi(v) = \sum_{a \in \delta^+ v} \varphi(a) - \sum_{a \in \delta^- v} \varphi(a) \quad (v \in V) \tag{3.16}$$

is called the *boundary* of φ.

Now, the submodular flow problem of Edmonds and Giles [4] is described as follows.

$$(\mathrm{P}_s) \quad \text{Minimize} \quad \sum_{a \in A} \gamma(a)\varphi(a) \tag{3.17}$$

$$\text{subject to} \quad \underline{c}(a) \leq \varphi(a) \leq \bar{c}(a) \quad (a \in A). \tag{3.18}$$

$$\partial\varphi \in \mathrm{B}(f). \tag{3.19}$$

A flow φ satisfying (3.18) and (3.19) is called a *submodular flow* in $\mathcal{N}$. An *optimal submodular flow* is an optimal solution φ of Problem (P_s). When $\bar{c} \equiv 1$ (i.e., $\bar{c}(a) = 1$ $(a \in A)$), $\underline{c} \equiv 0$ (i.e., $\underline{c}(a) = 0$ $(a \in A)$) and f is an integer-valued function, Problem (P_s) is called a *0-1 submodular flow problem.*

Since the capacity functions take on finite real values, the set of all the feasible solutions of Problem (P_s) is a bounded polyhedron. It follows that there exists an optimal solution of Problem (P_s) if the problem is feasible.

We adopt a theorem in [11], which shows an optimality condition for submodular flows. Any function $p : V \to \mathbf{R}$ is called a *potential*.

Theorem 3.1 ([11, p. 136]): *A submodular flow $\varphi : A \to \mathbf{R}$ for Problem (P_s) is optimal if and only if there exists a potential $p : V \to \mathbf{R}$ such that, defining $\gamma_p : A \to \mathbf{R}$ by*

$$\gamma_p(a) = \gamma(a) + p(\partial^+ a) - p(\partial^- a) \quad (a \in A), \tag{3.20}$$

we have for each $a \in A$

$$\gamma_p(a) > 0 \Longrightarrow \varphi(a) = \underline{c}(a), \tag{3.21}$$

$$\gamma_p(a) < 0 \Longrightarrow \varphi(a) = \bar{c}(a) \tag{3.22}$$

and such that the boundary $\partial\varphi : V \to \mathbf{R}$ is a maximum-weight base of $\mathrm{B}(f)$ with respect to the weight function p, i.e.,

$$\sum_{v \in V} p(v)\partial\varphi(v) = \max\{\sum_{v \in V} p(v)x(v) \mid x \in \mathrm{B}(f)\}. \tag{3.23}$$

□

Let $\Delta = (\varphi, z)$ be a pair of a flow $\varphi : A \to \mathbf{R}$, satisfying (3.18), and a base $z \in \mathrm{B}(f)$. We call such $\Delta = (\varphi, z)$ a *submodular pseudoflow*. Note that a submodular pseudoflow $\Delta = (\varphi, z)$ gives a submodular flow φ if $\partial\varphi = z$. We define the *auxiliary network* $\mathcal{N}_\Delta = (G_\Delta = (V, A_\Delta), c_\Delta, \gamma_\Delta)$ associated with a submodular pseudoflow $\Delta = (\varphi, z)$ as follows. G_Δ is a directed graph with vertex set V and arc set A_Δ defined by

$$\begin{aligned} A_\Delta &= A_\varphi \cup B_\varphi \cup C_z, & (3.24)\\ A_\varphi &= \{a \mid a \in A, \varphi(a) < \bar{c}(a)\}, & (3.25)\\ B_\varphi &= \{\bar{a} \mid a \in A, \varphi(a) > \underline{c}(a)\} \ \ (\bar{a} : \text{a reorientation of } a), & (3.26)\\ C_z &= \{(u, v) \mid u, v \in V,\ u \in \mathrm{dep}(z, v) - \{v\}\}. & (3.27) \end{aligned}$$

The capacity function $c_\Delta : A_\Delta \to \mathbf{R}$ is given by

$$c_\Delta(a) = \begin{cases} \bar{c}(a) - \varphi(a) & (a \in A_\varphi) \\ \varphi(\bar{a}) - \underline{c}(\bar{a}) & (a \in B_\varphi,\ \bar{a}\ (\in A) : \text{a reorientation of } a) \\ \tilde{c}(z, v, u) & (a = (u, v) \in C_z) \end{cases} \tag{3.28}$$

and $\gamma_\Delta : A_\Delta \to \mathbf{R}$ is the length function given by

$$\gamma_\Delta(a) = \begin{cases} \gamma(a) & (a \in A_\varphi) \\ -\gamma(\bar{a}) & (a \in B_\varphi,\ \bar{a}\ (\in A) : \text{a reorientation of } a) \\ 0 & (a = (u, v) \in C_z). \end{cases} \tag{3.29}$$

Theorem 3.2 ([11, p. 137]): *A submodular flow $\varphi : A \to \mathbf{R}$ for Problem* ($\mathrm{P_s}$) *is optimal if and only if there exists no directed cycle of negative length, relative to the length function γ_Δ, in the auxiliary network $\mathcal{N}_\Delta = (G_\Delta = (V, A_\Delta), c_\Delta, \gamma_\Delta)$ where $\Delta = (\varphi, z)$ with $z = \partial\varphi$.* □

Using the auxiliary network $\mathcal{N}_\Delta = (G_\Delta = (V, A_\Delta), c_\Delta, \gamma_\Delta)$ and Lemma 2.5, we can rewrite the optimality condition given by Theorem 3.1 as follows.

Theorem 3.3: *A submodular flow $\varphi : A \to \mathbf{R}$ for Problem* ($\mathrm{P_s}$) *is optimal if and only if there exists a potential $p : V \to \mathbf{R}$ such that, defining $\gamma_{\Delta,p} : A_\Delta \to \mathbf{R}$ by*

$$\gamma_{\Delta,p}(a) = \gamma_\Delta(a) + p(\partial^+ a) - p(\partial^- a) \quad (a \in A_\Delta). \tag{3.30}$$

we have $\gamma_{\Delta,p}(a) \geq 0$ for each $a \in A_\Delta$, where $\Delta = (\varphi, z)$ with $z = \partial\varphi$. □

For any positive real number ε we define the ε-optimality for a submodular pseudoflow $\Delta = (\varphi, z)$. This concept is fundamental for our cost scaling algorithm.

Definition 3.4: A submodular pseudoflow $\Delta = (\varphi, z)$ is said to be *ε-optimal* if there exists a potential $p : V \to \mathbf{R}$ such that $\gamma_{\Delta,p}(a) \geq -\varepsilon$ for all $a \in A_\Delta$. *For such a potential p we say that $\Delta = (\varphi, z)$ is ε-optimal with respect to potential p.* □

Put $\Gamma = \max_{a \in A} |\gamma(a)|$. Then, we have

Lemma 3.5: *Any submodular pseudoflow $\Delta = (\varphi, z)$ is ε-optimal for $\varepsilon \geq \Gamma$ and any ε-optimal submodular flow φ with $\varepsilon < 1/n$ is an optimal submodular flow.*

Proof: The first part of the lemma can be verified by taking $p \equiv 0$. For the second part of the lemma, we see that if $\varepsilon < 1/n$, then there is no negative directed cycle in $\mathcal{N}_\Delta = (G_\Delta = (V, A_\Delta), c_\Delta, \gamma_\Delta)$ for $\Delta = (\varphi, z)$, since the length $\sum_{a \in C} \gamma_\Delta(a) = \sum_{a \in C} \gamma_{\Delta,p}(a)$ of each cycle C is an integer and is greater than or equal to $-\varepsilon n > -1$. Hence, the optimality of the submodular flow φ follows from Theorem 3.2. □

4. A Cost Scaling Framework

In our cost scaling algorithm we execute a procedure called *Refine* which converts a 2ε-optimal submodular flow to an ε-optimal submodular pseudoflow and then converts it to an ε-optimal submodular flow. Two basic operations called Relabel and Push are performed in procedure Refine. Given a submodular pseudoflow $\Delta = (\varphi, z)$ and the associated auxiliary network $\mathcal{N}_\Delta = (G_\Delta = (V, A_\Delta), c_\Delta, \gamma_\Delta)$, suppose that we have a potential $p : V \to \mathbf{R}$ such that Δ is ε-optimal with respect to p. For each $v \in V$ let $e(v) = z(v) - \partial\varphi(v)$, which is called the *excess* on v. If $e(v) > 0$, then v is called an *active vertex*.

For an ε-optimal submodular pseudoflow $\Delta = (\varphi, z)$ with respect to a potential p, an arc $a \in A_\Delta$ is called an *admissible arc* in $\mathcal{N}_\Delta = (G_\Delta = (V, A_\Delta), c_\Delta, \gamma_\Delta)$ if $-\varepsilon \le \gamma_{\Delta,p}(a) < 0$. Note that in our algorithm given below $p(v)/\varepsilon$ for any $v \in V$ is always an integer, and hence for each $a \in C_z$ a is an admissible arc if and only if $\gamma_{\Delta,p}(a) = -\varepsilon$.

The relabeling operation on $v \in V$ is defined as follows.

Relabel(v): Applicability: $v \in V$ and for any $a \in A_\Delta$ with $\partial^+ a = v$ we have $\gamma_{\Delta,p}(a) \ge 0$;
Action: $p(v) \leftarrow p(v) - \varepsilon$.

The push operations **Push1**(a) and **Push2**(a) for $a \in A_\Delta$ are defined as follows. Here, $\bar{a}$ denotes a reorientation of a.

Push1(a): Applicability: $a \in A_\varphi \cup B_\varphi$, $e(\partial^+ a) > 0$ and $\gamma_{\Delta,p}(a) < 0$;
Action:
If $a \in A_\varphi$, then $\varphi(a) \leftarrow \varphi(a) + \min(e(\partial^+ a), c_\Delta(a))$.
If $a \in B_\varphi$, then $\varphi(\bar{a}) \leftarrow \varphi(\bar{a}) - \min(e(\partial^+ a), c_\Delta(a))$ for $\bar{a} \in A$.

Push2(a): Applicability: $a \in C_z$, $e(\partial^+ a) > 0$ and $\gamma_{\Delta,p}(a) = -\varepsilon$;
Action: $z \leftarrow z + \alpha(\chi_{\partial^- a} - \chi_{\partial^+ a})$ where $\alpha = \min(e(\partial^+ a), c_\Delta(a))$.

We can easily see the following.

Lemma 4.1: *If v is an active vertex, then either a push for some $a \in A_\Delta$ with $\partial^+ a = v$ or a relabel of v is applicable.* □

An algorithm for the minimum-cost submodular flow problem is described as follows. The integer L given in the input can be any positive integer at the moment and will be appropriately determined in the next section.

Algorithm Submodular Flow
Input: $\mathcal{N} = (G = (V, A), \bar{c}, \underline{c}, \gamma, (\mathcal{D}, f))$, a positive integer L, a potential $p \equiv 0$ and $\varepsilon = \Gamma = \max\{|\gamma(a)| \mid a \in A\}$.
Output: An optimal submodular flow φ in $\mathcal{N}$.
Step 1: While $\varepsilon \ge 1/n$, put $\varepsilon \leftarrow \varepsilon/2$ and perform procedure Refine(ε, L, p).
(End)

Procedure Refine(ε, L, p).
Input: ε, L, and p such that there exists a 2ε-optimal submodular flow φ with respect to p.
Output: A potential p and an ε-optimal submodular flow φ with respect to p.
Step 0: For the current p, find an integer vector z_0 in $\mathrm{B}(f)$ such that

$$\sum_{v \in V} p(v) z_0(v) = \max_{z' \in \mathrm{B}(f)} \sum_{v \in V} p(v) z'(v). \tag{4.31}$$

Put $z \leftarrow z_0$. For each $a \in A$, if $\gamma_p(a) < 0$ then put
$\varphi(a) \leftarrow \bar{c}(a)$,
otherwise put
$\varphi(a) \leftarrow \underline{c}(a)$.
Put $\Delta \leftarrow (\varphi, z)$.
Step 1: While there exists an active vertex $v \in V$ (satisfying $e(v) > 0$) that has been relabeled less than L times, choose one such vertex v and do the following (1-1)~(1-3) (if there exists no such vertex, then the procedure terminates and let the current φ, ε and p be the output):
(1-1) Applicability: For any $a \in A_\Delta$ with $\partial^+ a = v$ we have $\gamma_{\Delta,p}(a) \geq 0$;
$p(v) \leftarrow p(v) - \varepsilon$.
(1-2) Applicability: For some $a \in A_\varphi \cup B_\varphi$ with $\partial^+ a = v$ we have $\gamma_{\Delta,p}(a) < 0$;
Perform Push1(a).
(1-3) Applicability: For some $a \in C_z$ with $\partial^+ a = v$ we have $\gamma_{\Delta,p}(a) = -\varepsilon$;
Perform Push2(a).
(End)

It should be noted that we can easily find a minimum-weight base z_0 with respect to the weight function p by a greedy algorithm (see [11]).

We have the following lemmas.

Lemma 4.2: *The submodular pseudoflow Δ obtained in Step 0 of procedure Refine is 0-optimal with respect to the potential in the input.*

Proof: The fact that $\gamma_{\Delta,p}(a) \geq 0$ for each $a \in A_\varphi \cup B_\varphi$ directly follows from the definition of φ. Also we have $\gamma_{\Delta,p}(a) \geq 0$ for each $a \in C_z$ from Lemma 2.5. □

Lemma 4.3: *The relabeling operation in procedure Refine keeps the ε-optimality of $\Delta = (\varphi, z)$ with respect to the updated potential p.*

Proof: Immediate from the definition of a relabel operation. □

Lemma 4.4: *Both two types of push operations keep $\Delta = (\varphi, z)$ a submodular pseudoflow and the ε-optimality of $\Delta = (\varphi, z)$ with respect to the current potential p.*

Proof: Since the potential p is not changed, it is enough to prove that $\gamma_{\Delta,p}(a) \geq -\varepsilon$ for any new arc generated by a push.

Suppose a' is a new arc with $\partial^+ a' = w$ and $\partial^- a' = s$ after a push operation on an admissible arc $a \in C_z$ with $\partial^+ a = u$ and $\partial^- a = v$. By Lemma 2.3 we have
(i) $u = s$ or there exists an arc $a_1 \in C_z$ with $\partial^+ a_1 = u$ and $\partial^- a_1 = s$ and
(ii) $v = w$ or there exists an arc $a_2 \in C_z$ with $\partial^+ a_2 = w$ and $\partial^- a_2 = v$.

From (i) and (ii) we have $p(u) - p(s) \geq -\varepsilon$ and $p(w) - p(v) \geq -\varepsilon$, respectively. Hence,

$$p(w) - p(s) \geq p(v) - p(s) - \varepsilon = p(u) + \varepsilon - p(s) - \varepsilon \geq -\varepsilon. \tag{4.32}$$

It follows that $\gamma_{\Delta,p}(w, s) \geq -\varepsilon$.

Furthermore, a push on an admissible arc $a \in A_\varphi \cup B_\varphi$ only produces a new arc $\bar{a}$, a reorientation of a, for which we have $\gamma_\Delta(\bar{a}) + p(\partial^- a) - p(\partial^+ a) > 0$.

This completes the proof. □

Lemma 4.5: *At the end of procedure Refine, if there exists no active vertex, then φ in the output is an ε-optimal submodular flow with respect to the then obtained potential p.*

Proof: The present lemma follows from Lemmas 4.3 and 4.4 and the fact that $\partial\varphi = z$, since $\partial\varphi \geq z$ (due to $e(v) \leq 0$ $(v \in V)$) and $z(V) = f(V) = \partial\varphi(V) = 0$. □

Lemma 4.6: *If for a current submodular pseudoflow $\Delta = (\varphi, z)$ in procedure Refine there exists $v \in V$ such that $e(v) > 0$ and $\{a \mid a \in A_\Delta, \partial^+ a = v\} = \emptyset$, then Problem (P_s) is infeasible.*

Proof: Under the assumption of the present lemma we can easily show that $z(v)$ is equal to the minimum value (i.e., $f(V) - f(V - \{v\})$) of $z'(v)$ for all bases $z' \in B(f)$ and that $\partial\varphi(v)$ is equal to the maximum value (i.e., $\bar{c}(\delta^+ v) - \underline{c}(\delta^- v)$) of $\partial\varphi'(v)$ for all flows φ' satisfying (3.18). Therefore, Problem (P_s) is infeasible since $z(v) > \partial\varphi(v)$. □

In the next section the L in the input will be appropriately given so that at the end of procedure Refine there exists no active vertex.

5. The Number of Relabeling Operations

Our cost scaling algorithm repeatedly performs procedure Refine. Obviously, the iteration number of procedure Refine is $O(\log(n\Gamma))$, without considering the validity of the algorithm. In this section we give an appropriate value of L that bounds the number of relabelings on each vertex in V during an execution of procedure Refine and that validates the whole algorithm.

We first give a lemma and its corollary where $f(V) = 0$ is not assumed. In fact, the following lemma is meaningful only if $f(V) > 0$.

Lemma 5.1: *For a submodular system $(\mathcal{D}, f)$ on V, let z_1, z_2 be nonnegative bases in $\mathrm{B}(f)$. Suppose that for a potential $p : V \to \mathbf{R}$ and a real $\varepsilon > 0$ we have $p(u) - p(v) \geq -\varepsilon$ for any $u, v \in V$ with $u \in \mathrm{dep}(z_1, v) - \{v\}$. Then,*

$$\sum_{v \in V} p(v)(z_1(v) - z_2(v)) \geq -\varepsilon f(V). \tag{5.33}$$

Proof: Define a bipartite graph $G_b = (V, V'; A_{z_1})$ where V' is a copy of V and the arc set is given by $A_{z_1} = \{(u, v') \mid u, v \in V, u \in \mathrm{dep}(z_1, v)\}$. The upper capacities of the arcs in A_{z_1} are assumed to be infinity and the lower capacities of the arcs in A_{z_1} are assumed to be zero. For any subset U of V let $W = \{w \mid w \in V,\ u' \in U',\ (w, u') \in A_{z_1}\}$. It follows from the definition of A_{z_1} and W that $z_1(W) = f(W)$ and $U \subseteq W$. Hence, $z_2(U) \leq z_2(W) \leq f(W) = z_1(W)$. Consequently, from a theorem in [16], there exists a function $g : A_{z_1} \to \mathbf{R}_+$ such that

$$g(\delta^+ u) = z_1(u) \quad (u \in V), \qquad g(\delta^- v') = z_2(v') \quad (v' \in V'), \tag{5.34}$$

where

$$\delta^+ u = \{(u, v') \mid v' \in V', (u, v') \in A_{z_1}\}, \tag{5.35}$$

$$\delta^- v' = \{(u, v') \mid u \in V, (u, v') \in A_{z_1}\} \tag{5.36}$$

and $z_2(v') = z_2(v)$ for $v' \in V'$.

Define $p(v') = p(v)$ for $v' \in V'$. Then,

$$\begin{aligned}
\sum_{u \in V} p(v)(z_1(v) - z_2(v)) &= \sum_{u \in V} z_1(u)p(u) - \sum_{v' \in V'} z_2(v')p(v') \\
&= \sum_{u \in V} g(\delta^+ u)p(u) - \sum_{v' \in V'} g(\delta^- v')p(v') \\
&= \sum_{a \in A_{z_1}} (p(\partial^+ a) - p(\partial^- a))g(a) \\
&\geq -\varepsilon \sum_{a \in A_{z_1}} g(a) \\
&= -\varepsilon \sum_{u \in V} z_1(u) \\
&= -\varepsilon f(V).
\end{aligned} \tag{5.37}$$

□

From Lemma 5.1 we can easily show the following.

Corollary 5.2: *For any submodular system $(\mathcal{D}, f)$ on V, let $z_1, z_2 \in \mathrm{B}(f)$ and $d \in \mathbf{R}^V$ be such that $z_1(v) + d(v) \geq 0$ and $z_2(v) + d(v) \geq 0$ for all $v \in V$. Suppose that for a potential $p : V \to \mathbf{R}$ and a real $\varepsilon > 0$ we have $p(u) - p(v) \geq -\varepsilon$ for any $u, v \in V$ with $u \in \mathrm{dep}(z_1, v) - \{v\}$. Then,*

$$\sum_{v \in V} p(v)(z_1(v) - z_2(v)) \geq -\varepsilon(f(V) + d(V)). \tag{5.38}$$

□

Let $\Delta' = (\varphi', z' = \partial\varphi')$ be a 2ε-optimal submodular flow with respect to p' and $\Delta = (\varphi, z)$ be an ε-optimal submodular pseudoflow with respect to p, where p' is the input of an execution of procedure Refine and $\Delta = (\varphi, z)$ and p are, respectively, the current submodular pseudoflow and the corresponding potential in the execution of procedure Refine. Define

$$\begin{aligned}
S^+ &= \{v \in V \mid z(v) - \partial\varphi(v) > 0\}, && (5.39)\\
S^- &= \{v \in V \mid z(v) - \partial\varphi(v) < 0\}, && (5.40)\\
E_+ &= \{(u, v) \in A_\varphi \mid \varphi'(u, v) > \varphi(u, v)\} \\
&\quad \cup \{(u, v) \in B_\varphi \mid \varphi(v, u) > \varphi'(v, u)\}, && (5.41)\\
E_- &= \{(u, v) \in A_{\varphi'} \mid \varphi(u, v) > \varphi'(u, v)\} \\
&\quad \cup \{(u, v) \in B_{\varphi'} \mid \varphi'(v, u) > \varphi(v, u)\}. && (5.42)
\end{aligned}$$

Note that $p'(v) = p(v)$ for $v \in S^-$ since we only relabel active vertices. Recall that in the following equations $\bar{a}$ denotes the reorientation of an arc a. Now,

$$\begin{aligned}
&\sum_{a \in E_+ \cap A_\varphi} (p(\partial^+ a) - p(\partial^- a))(\varphi'(a) - \varphi(a)) + \sum_{a \in E_+ \cap B_\varphi} (p(\partial^+ a) - p(\partial^- a))(\varphi(\bar{a}) - \varphi'(\bar{a}))\\
&= -\sum_{v \in V} p(v)(\partial\varphi(v) - \partial\varphi'(v))\\
&= -\sum_{v \in V} p(v)(\partial\varphi(v) - z(v)) - \sum_{v \in V} p(v)(z(v) - z'(v))\\
&= -\sum_{v \in S^+} p(v)(\partial\varphi(v) - z(v)) - \sum_{v \in S^-} p(v)(\partial\varphi(v) - z(v))\\
&\quad -\sum_{v \in V} p(v)(z(v) - z'(v)).
\end{aligned} \tag{5.43}$$

Therefore, we have

$$\sum_{a \in E_+ \cap A_\varphi} \gamma_{\Delta, p}(a)(\varphi'(a) - \varphi(a)) + \sum_{a \in E_+ \cap B_\varphi} \gamma_{\Delta, p}(a)(\varphi(\bar{a}) - \varphi'(\bar{a}))$$

$$
\begin{aligned}
&= \sum_{a\in E_+\cap A_\varphi} \gamma_\Delta(a)(\varphi'(a)-\varphi(a)) + \sum_{a\in E_+\cap B_\varphi} \gamma_\Delta(a)(\varphi(\bar a)-\varphi'(\bar a)) \\
&\quad + \sum_{a\in E_+\cap A_\varphi} (p(\partial^+ a)-p(\partial^- a))(\varphi'(a)-\varphi(a)) \\
&\quad + \sum_{a\in E_+\cap B_\varphi} (p(\partial^+ a)-p(\partial^- a))(\varphi(\bar a)-\varphi'(\bar a)) \\
&= \sum_{a\in E_+\cap A_\varphi} \gamma_\Delta(a)(\varphi'(a)-\varphi(a)) + \sum_{a\in E_+\cap B_\varphi} \gamma_\Delta(a)(\varphi(\bar a)-\varphi'(\bar a)) \\
&\quad - \sum_{v\in S^+} p(v)(\partial\varphi(v)-z(v)) - \sum_{v\in S^-} p(v)(\partial\varphi(v)-z(v)) \\
&\quad - \sum_{v\in V} p(v)(z(v)-z'(v)). \qquad (5.44)
\end{aligned}
$$

On the other hand, we have

$$
\begin{aligned}
&\sum_{a\in E_-\cap A_{\varphi'}} (p'(\partial^+ a)-p'(\partial^- a))(\varphi(a)-\varphi'(a)) + \sum_{a\in E_-\cap B_{\varphi'}} (p'(\partial^+ a)-p'(\partial^- a))(\varphi'(\bar a)-\varphi(\bar a)) \\
&= -\sum_{v\in V} p'(v)(\partial\varphi'(v)-\partial\varphi(v)) \\
&= -\sum_{v\in V} p'(v)(z(v)-\partial\varphi(v)) - \sum_{v\in V} p'(v)(z'(v)-z(v)) \\
&= -\sum_{v\in S^+} p'(v)(z(v)-\partial\varphi(v)) - \sum_{v\in S^-} p'(v)(z(v)-\partial\varphi(v)) \\
&\quad - \sum_{v\in V} p'(v)(z'(v)-z(v)). \qquad (5.45)
\end{aligned}
$$

Hence, we have

$$
\begin{aligned}
&\sum_{a\in E_-\cap A_{\varphi'}} \gamma_{\Delta',p'}(a)(\varphi(a)-\varphi'(a)) + \sum_{a\in E_-\cap B_{\varphi'}} \gamma_{\Delta',p'}(a)(\varphi'(\bar a)-\varphi(\bar a)) \\
&= \sum_{a\in E_-\cap A_{\varphi'}} \gamma_{\Delta'}(a)(\varphi(a)-\varphi'(a)) + \sum_{a\in E_-\cap B_{\varphi'}} \gamma_{\Delta'}(a)(\varphi'(\bar a)-\varphi(\bar a)) \\
&\quad + \sum_{a\in E_-\cap A_{\varphi'}} (p'(\partial^+ a)-p'(\partial^- a))(\varphi(a)-\varphi'(a)) \\
&\quad + \sum_{a\in E_-\cap B_{\varphi'}} (p'(\partial^+ a)-p'(\partial^- a))(\varphi'(\bar a)-\varphi(\bar a)) \\
&= \sum_{a\in E_-\cap A_{\varphi'}} \gamma_{\Delta'}(a)(\varphi(a)-\varphi'(a)) + \sum_{a\in E_-\cap B_{\varphi'}} \gamma_{\Delta'}(a)(\varphi'(\bar a)-\varphi(\bar a)) \\
&\quad - \sum_{v\in S^+} p'(v)(z(v)-\partial\varphi(v)) - \sum_{v\in S^-} p'(v)(z(v)-\partial\varphi(v)) \\
&\quad - \sum_{v\in V} p'(v)(z'(v)-z(v)). \qquad (5.46)
\end{aligned}
$$

From (5.44) and Corollary 5.2,

$$\begin{aligned}
&\sum_{a\in E_+\cap A_\varphi} \gamma_\Delta(a)(\varphi'(a)-\varphi(a)) + \sum_{a\in E_+\cap B_\varphi} \gamma_\Delta(a)(\varphi(\bar a)-\varphi'(\bar a)) \\
&\quad - \sum_{v\in S^+} p(v)(\partial\varphi(v)-z(v)) - \sum_{v\in S^-} p(v)(\partial\varphi(v)-z(v)) \\
&= \sum_{a\in E_+\cap A_\varphi} \gamma_{\Delta,p}(a)(\varphi'(a)-\varphi(a)) + \sum_{a\in E_+\cap B_\varphi} \gamma_{\Delta,p}(a)(\varphi(\bar a)-\varphi'(\bar a)) \\
&\quad + \sum_{v\in V} p(v)(z(v)-z'(v)) \\
&\geq -\varepsilon d(V) - \varepsilon\{\sum_{a\in E_+\cap A_\varphi} (\varphi'(a)-\varphi(a)) + \sum_{a\in E_+\cap B_\varphi} (\varphi(\bar a)-\varphi'(\bar a))\}, \quad (5.47)
\end{aligned}$$

where d is a vector in $\mathbf{R}^V$ such that $z+d\geq \mathbf{0}$ and $z'+d\geq \mathbf{0}$. Note that $f(V)=0$. Also, from (5.46) and Corollary 5.2,

$$\begin{aligned}
&\sum_{a\in E_-\cap A_{\varphi'}} \gamma_{\Delta'}(a)(\varphi(a)-\varphi'(a)) + \sum_{a\in E_-\cap B_{\varphi'}} \gamma_{\Delta'}(a)(\varphi'(\bar a)-\varphi(\bar a)) \\
&\quad - \sum_{v\in S^+} p'(v)(z(v)-\partial\varphi(v)) - \sum_{v\in S^-} p'(v)(z(v)-\partial\varphi(v)) \\
&= \sum_{a\in E_-\cap A_{\varphi'}} \gamma_{\Delta',p'}(a)(\varphi(a)-\varphi'(a)) + \sum_{a\in E_-\cap B_{\varphi'}} \gamma_{\Delta',p'}(a)(\varphi'(\bar a)-\varphi(\bar a)) \\
&\quad + \sum_{v\in V} p'(v)(z'(v)-z(v)) \\
&\geq -2\varepsilon d(V) - 2\varepsilon\{\sum_{a\in E_-\cap A_{\varphi'}} (\varphi(a)-\varphi'(a)) + \sum_{a\in E_-\cap B_{\varphi'}} (\varphi'(\bar a)-\varphi(\bar a))\}. (5.48)
\end{aligned}$$

Putting $C=\max_{a\in A}(|\bar c(a)|+|\underline{c}(a)|)$ and adding the above two inequalities (5.47) and (5.48), we have

$$\sum_{v\in S^+} (p'(v)-p(v))(z(v)-\partial\varphi(v)) \leq 3\varepsilon d(V)+6\varepsilon mC, \quad (5.49)$$

where note that $p'(v)=p(v)$ for $v\in S^-$ and

$$E_-\cap A_{\varphi'}=\{\bar a\,|\,a\in E_+\cap B_\varphi\},\quad E_-\cap B_{\varphi'}=\{\bar a\,|\,a\in E_+\cap A_\varphi\}, \quad (5.50)$$

$$\gamma_{\Delta'}(a)=\gamma_\Delta(\bar a)\quad (a\in A_{\varphi'}). \quad (5.51)$$

If z and φ are integer vectors and procedure Refine terminates when each active vertex is relabeled L (an integer) times, then from (5.49) we have

$$\sum_{v\in S^+} L\varepsilon \leq \sum_{v\in S^+} (p'(v)-p(v))(z(v)-\partial\varphi(v)) \leq 3\varepsilon d(V)+6\varepsilon mC. \quad (5.52)$$

Hence, if $S^+\neq\emptyset$, then

$$L\leq (3d(V)+6mC)/|S^+|. \quad (5.53)$$

Theorem 5.3: *If we choose L such that $L > 3d(V) + 6mC$ and if each vertex is relabeled at most L times, then procedure Refine terminates with (φ, z) such that $z = \partial\varphi$.*

Proof: It follows from the assumption and (5.53) that $S^+ = \emptyset$, i.e., $z = \partial\varphi$ for the output $\Delta = (\varphi, z)$. □

Define a vector $x_0 \in \mathbf{R}^V$ by $x_0(v) = -C|\delta^+ v \cup \delta^- v|$ for each $v \in V$. If Problem (P_s) has a feasible solution φ, then $x_0 \leq \partial\varphi \in \mathrm{B}(f)$. Let $(2^V, f_{x_0})$ be the contraction of $(\mathcal{D}, f)$ by the vector x_0. Replacing f by f_{x_0} in Problem (P_s) does not change the set of all feasible submodular flows. For given z, $z' \in \mathrm{B}(f_{x_0})$ as above, we have $z - x_0 \geq \mathbf{0}$ and $z' - x_0 \geq \mathbf{0}$ from Lemma 2.4. Then, since $-x_0(V) = \sum_{v\in V} C|\delta^+ v \cup \delta^- v| = 2mC$, putting $d = -x_0$, we have from (5.49)

$$\sum_{v\in S^+} (p'(v) - p(v))(z(v) - \partial\varphi(v)) \leq 12\varepsilon mC. \tag{5.54}$$

Theorem 5.4: *If we take $L = 12mC + 1$ and relabel each vertex at most L times, then procedure Refine terminates with (φ, z) such that $z = \partial\varphi$.*

Proof: Put $d = -x_0$, using x_0 defined above. The present theorem follows from Theorem 5.3. □

For the estimation of the number of pushes, we have obtained an implementation of procedure Refine which performs at most $\mathrm{O}(n^3 mC)$ push operations (see [22]), but we do not get into its detail here. In the next section we shall consider an efficient implementation of it for 0-1 submodular flows.

6. A Refinement for 0-1 Submodular Flows

In this section, for the 0-1 submodular flow problem we give a refinement of our cost scaling algorithm by introducing a label for each vertex in V. We use a hybrid version for procedure Refine, which consists of two subprocedures: PushRelabel and SuccessiveShortestPath.

From now on we assume without loss of generality that the vertex set V is indexed as $V = \{v_1, v_2, \cdots, v_n\}$ and that the underlying graph $G = (V, A)$ does not have any selfloops or any two arcs $a, a' \in A$ such that $\{\partial^+ a, \partial^- a\} = \{\partial^+ a', \partial^- a'\}$. The latter assumption ensures that for each distinct two vertices v_i, v_j there exist at most two arcs from v_i to v_j in the auxiliary graph associated with any submodular pseudoflow $\Delta = (\varphi, z)$, possibly one from $A_\varphi \cup B_\varphi$ and one from C_z. For convenience, we write $a = (u, v)$ for an arc a to mean that u is the initial vertex of the arc a and v is the terminal vertex of a.

For each $i \in \{1, 2, \cdots, n\}$ we have a label $\nu(i)$ that takes on values in $\{1, 2, \cdots, n\}$.

Algorithm 0-1 Submodular Flow
Input: $\mathcal{N} = (G = (V, A), \gamma, (\mathcal{D}, f))$, a positive integer L, a potential $p \equiv 0$ and $\varepsilon = \Gamma = \max\{|\gamma(a)| \mid a \in A\}$.
Output: An optimal 0-1 submodular flow φ in $\mathcal{N}$.
Step 1: While $\varepsilon \geq 1/n$, put $\varepsilon \leftarrow \varepsilon/4$, perform procedure Refine(ε, L, p) and put $\varepsilon \leftarrow 2\varepsilon$.
(End)

The input L can be any positive integer at the moment and will be optimized later.

Procedure Refine(ε, L, p)
Input: $\mathcal{N}$, L, ε, and p such that there exists a 4ε-optimal 0-1 submodular flow with respect to p, and $\nu(i) = 1$ $(i = 1, 2, \cdots, n)$.
Output: A potential p and a 2ε-optimal 0-1 submodular flow $\Delta = (\varphi, z)$ of $\mathcal{N}$ with respect to p.
Step 1: Perform procedure PushRelabel(ε, L, p, ν).
Step 2: Perform procedure SuccessiveShortestPath$(\varepsilon, p, \Delta = (\varphi, z))$.
(End)

We first consider procedure PushRelabel for 0-1 submodular flows.

Procedure PushRelabel(ε, L, p, ν).
Input: ε, L, and p such that there exists a 4ε-optimal 0-1 submodular flow φ with respect to p, and a label ν.
Output: A potential p and an ε-optimal 0-1 submodular pseudoflow $\Delta = (\varphi, z)$ with respect to p.
Step 0: For the current p, find an integral vector z_0 in $\mathrm{B}(f)$ such that

$$\sum_{v \in V} p(v) z_0(v) = \max_{z' \in \mathrm{B}(f)} \sum_{v \in V} p(v) z'(v). \tag{6.55}$$

Put $z \leftarrow z_0$. For each $a \in A$, if $\gamma_p(a) < 0$ then put
$\varphi(a) \leftarrow 1$,
otherwise put
$\varphi(a) \leftarrow 0$.
Put $\Delta \leftarrow (\varphi, z)$.
Step 1: If there exists no active vertex relabeled less than L times, then output the current potential p, ε-optimal submodular pseudoflow $\Delta = (\varphi, z)$ with respect to p and label ν and return to procedure Refine. Otherwise, let v_i be an active vertex relabeled less than L times.
Step 2:

(2-1) If there exists an arc $a \in A_\Delta$ such that $a = (v_i, v_{\nu(i)})$ and $\gamma_{\Delta,p}(a) < 0$, then perform Push1(a) or Push2(a) according as $a \in A_\varphi \cup B_\varphi$ or $a \in C_z$, and go to Step 1.
(2-2) If there exists no arc $a \in A_\Delta$ such that $a = (v_i, v_{\nu(i)})$ or if for each such arc a we have $\gamma_{\Delta,p}(a) \geq 0$, then
  (2-2a) if $\nu(i) < n$, then put $\nu(i) \leftarrow \nu(i) + 1$ and go to Step 1;
  (2-2b) if $\nu(i) = n$, then put $\nu(i) \leftarrow 1$ and $p(v_i) \leftarrow p(v_i) - \varepsilon$ and go to Step 1.
(End)

Note that the greedy algorithm finds an integral z_0 satisfying (6.55). Also note that at the end of Step 0, $\Delta = (\varphi, z)$ is a 0-optimal submodular pseudoflow with respect to the current potential p and that during procedure PushRelabel the current submodular pseudoflow Δ is always ε-optimal with respect to the current potential p.

Lemma 6.1: *Throughout the algorithm the following property* $(*)$ *is maintained:*
$(*)$ *For any vertex* $v_i \in V$ *and any arc* $a \in A_\Delta$ *with* $a = (v_i, v_j)$ *satisfying* $j < \nu(i)$ *for the current label* $\nu(i)$ *we have* $\gamma_{\Delta,p}(a) \geq 0$.

Proof: Suppose that currently $(*)$ holds and that the next basic operation is a relabeling operation for a vertex v_i. This operation does not generate any new arc. Denote the current potential by p' and the one after the operation by p. Note that $p'(w) \geq p(w)$ $(w \in V)$. For v_i, the current label $\nu(i)$ is made equal to 1. Furthermore, for any other label $\nu(j)$ $(j \neq i)$ and any arc $a \in A_\Delta$ such that $a = (v_j, v_k)$ and $k < \nu(j)$ we have $\gamma_\Delta(a) + p'(\partial^+ a) - p'(\partial^- a) \geq 0$, $p(\partial^+ a) = p'(\partial^+ a)$ $(= p'(v_j))$ and $p(\partial^- a) \leq p'(\partial^- a)$. Hence, $(*)$ holds after the relabeling operation.

Next, suppose that currently $(*)$ holds and that the next basic operation is a push for an arc a such that $a = (v_i, v_j)$ with $j = \nu(i)$. Note that potential p is not changed by the push. Therefore, it suffices to show that after the push operation any new arc $\hat{a} = (v_{i'}, v_{j'})$ with $j' \leq \nu(i')$ satisfies $\gamma_{\Delta'}(\hat{a}) + p(\partial^+ \hat{a}) - p(\partial^- \hat{a}) \geq 0$ where Δ' is the submodular pseudoflow obtained after the push on a. We first prove this for the case when $a \in C_z$. Suppose, on the contrary, that some such new arc $\hat{a} = (v_{i'}, v_{j'})$ with $j' \leq \nu(i')$ satisfies

$$p(v_{i'}) - p(v_{j'}) = -\varepsilon. \tag{6.56}$$

We show that (6.56) leads us to a contradiction. Recall that $a = (v_i, v_j)$ and $\hat{a} = (v_{i'}, v_{j'})$. From Lemma 2.3, before the push on arc $a = (v_i, v_j)$ we have
(i) $v_i = v_{j'}$ or there exists an arc in C_z from v_i to $v_{j'}$ and
(ii) $v_j = v_{i'}$ or there exists an arc in C_z from $v_{i'}$ to v_j.
Therefore,

$$p(v_{i'}) - p(v_j) \geq -\varepsilon, \quad p(v_i) - p(v_j) = -\varepsilon, \quad p(v_i) - p(v_{j'}) \geq -\varepsilon, \quad p(v_{i'}) - p(v_{j'}) = -\varepsilon. \tag{6.57}$$

Note that the last equation in (6.57) is (6.56). Since from (6.57)

$$p(v_{i'}) = p(v_{j'}) - \varepsilon \le p(v_i) = p(v_j) - \varepsilon \le p(v_{i'}). \tag{6.58}$$

we have

$$p(v_i) - p(v_{j'}) = -\varepsilon, \quad p(v_{i'}) - p(v_j) = -\varepsilon, \tag{6.59}$$

It follows from (6.59) that
(i) $v_i \neq v_{j'}$ and hence there is an arc $a_1 \in C_\xi$ with $a_1 = (v_i, v_{j'})$,
(ii) $v_j \neq v_{i'}$ and hence there is an arc $a_2 \in C_\xi$ with $a_2 = (v_{i'}, v_j)$.
From the induction hypothesis, (i) implies that $j \le j'$, whereas (ii) implies that $j' \le j$ since $j' \le \nu(i')$ and $\nu(i') \le j$ by the induction hypothesis. Hence, we have $j = j'$, i.e., $v_j = v_{j'}$, a contradiction.

For the case when $a \in A_\varphi \cup B_\varphi$, the only new arc $\hat{a}$ is a reorientation of a. Then $\gamma_{\Delta',p}(\hat{a}) = -\gamma_{\Delta,p}(a) \ge 0$. □

In the above proof of Lemma 6.1 we have also shown the following.

Lemma 6.2: *After a push operation, for any new arc $a = (v_i, v_j)$ with $\gamma_{\Delta,p}(a) < 0$ we have $j > \nu(i)$.* □

As in Golberg and Tarjan [17] we define saturating and nonsaturating pushes as follows.

Definition 6.3: A push on an arc $a \in A_\Delta$ with $a = (v, w)$ is called a *saturating push* if $e(v) \ge c_\Delta(a)$, i.e., $a = (v, w)$ satisfies one of the following two conditions:

(a) $a \in A_\varphi \cup B_\varphi$,

(b) $a \in C_\xi$ and $e(v) \ge c_\Delta(a)$.

Here, recall that we are dealing with 0-1 submodular pseudoflows, so that a push on any arc $a \in A_\varphi \cup B_\varphi$ is saturating.

Definition 6.4: A push on $(v, w) \in A_\Delta$ is called a *nonsaturating push* if it is not saturating (i.e., $a \in C_\xi$ with $a = (v, w)$ and $e(v) < c_\Delta(a)$).

Lemma 6.5: *The number of saturating push operations is at most $2n^2L$.*

Proof: By a saturating push on an arc a such that $a = (v_i, v_{\nu(i)})$ the arc a disappears from the current auxiliary graph and a possible new arc $a' = (v_i, v_j)$ with $\gamma_{\Delta,p}(a') < 0$ we have $j > \nu(i)$ for the current label $\nu(i)$ due to Lemma 6.2. From Lemmas 6.1 and 6.2 we see that between two successive relabeling operations on v_i there are at most $2n$ saturating pushes on arcs going out from v_i. So the total number of saturating pushes on arcs going out from v_i $(i = 1, 2, \cdots, n)$ is at most $2nL$. This proves the lemma. □

For the estimation of the number of nonsaturating pushes, we define two subsets of V for given $\Delta = (\varphi, z)$ and p by

$$D^+_{\Delta,p} = \{v \mid \exists a \in C_z : \partial^+ a = v,\ p(\partial^+ a) - p(\partial^- a) = -\varepsilon\}, \tag{6.60}$$

and

$$D^-_{\Delta,p} = \{v \mid \exists a \in C_z : \partial^- a = v,\ p(\partial^+ a) - p(\partial^- a) = -\varepsilon\}. \tag{6.61}$$

Lemma 6.6: *If a submodular pseudoflow $\Delta = (\varphi, z)$ is ε-optimal with respect to potential p, then we have $D^+_{\Delta,p} \cap D^-_{\Delta,p} = \emptyset$.*

Proof: Suppose on the contrary that there exists a vertex $v \in D^+_{\Delta,p} \cap D^-_{\Delta,p}$. It follows that there exist two arcs $a_1, a_2 \in C_z$ such that

$$\partial^+ a_1 = v,\ p(\partial^+ a_1) - p(\partial^- a_1) = -\varepsilon, \tag{6.62}$$

and

$$\partial^- a_2 = v,\ p(\partial^+ a_2) - p(\partial^- a_2) = -\varepsilon. \tag{6.63}$$

From Lemma 2.1 and equations (6.62)∼(6.63) there is an arc $a_3 \in C_z$ such that $a_3 = (\partial^+ a_2, \partial^- a_1)$ with $p(\partial^+ a_2) - p(\partial^- a_1) = p(\partial^+ a_2) - p(\partial^- a_2) + p(\partial^+ a_1) - p(\partial^- a_1) = -2\varepsilon$. This contradicts the ε-optimality of Δ. □

Lemma 6.7: *The number of nonsaturating pushes is at most $(n-1)nL + mL$.*

Proof: Let us denote by $d^+_{\Delta,p}$ the number of active vertices in $D^+_{\Delta,p}$. We show that a nonsaturating push reduce $d^+_{\Delta,p}$ by at least one. Let a nonsaturating push be performed on an arc $(u, v) \in C_z$. After the push, vertex u becomes inactive. Suppose that the push has introduced a new arc (w, s) satisfying $p(w) - p(s) = -\varepsilon$. In the proof of Lemma 6.1, we have shown that $(w, v) \in C_z$ and $p(w) - p(v) = -\varepsilon$ before the push. This implies that $w \in D^+_{\Delta,p}$ before the push. That is, a push on an arc in C_z does not add any new vertex to $D^+_{\Delta,p}$ (whether it is a saturating push or not).

On the other hand, each push on an arc in $A_\varphi \cup B_\varphi$ may increase $d^+_{\Delta,p}$ by at most one. Therefore, between two successive relabeling operations, the number of nonsaturating pushes is not greater than $n-1$ $(d^+_{\Delta,p} \leq n-1)$ plus the number of pushes on arcs in $A_\varphi \cup B_\varphi$ in the same period.

Consequently, the total number of nonsaturating pushes during the execution of procedure Refine is at most $n(n-1)L$ plus the total number of pushes on arcs in $A_\varphi \cup B_\varphi$ in procedure Refine. The latter number is at most mL, which can be shown by an argument similar to the proof of Lemma 6.5, where recall that a push on an arc in $A_\varphi \cup B_\varphi$ is saturating. □

Starting with an ε-optimal submodular pseudoflow and the corresponding potential p at the end of procedure PushRelabel, we perform procedure SuccessiveShortestPath described below. We get a 2ε-optimal submodular flow at the termination of procedure SuccessiveShortestPath. In procedure SuccessiveShortestPath, the cost function γ and the potential p obtained at the end of procedure PushRelabel are modified into $\tilde{\gamma}$ and $\hat{p}$ such that the initial submodular pseudoflow in procedure SuccessiveShortestPath is 0-optimal with respect to $\tilde{\gamma}$ and $\hat{p}$. Through successive shortest path augmentation steps, the given submodular pseudoflow is transformed into a submodular flow.

Before describing procedure SuccessiveShortestPath we show the following.

Lemma 6.8: *Let $\Delta = (\varphi, z)$ be ε-optimal with respect to potential p and cost γ. Then, $\Delta = (\varphi, z)$ is 0-optimal with respect to potential $\hat{p}$ and cost $\tilde{\gamma}$ given by*

$$\hat{p}(v) = \begin{cases} p(v) + \varepsilon & \text{for } v \in D^+_{\Delta,p} \\ p(v) & \text{for } v \notin D^+_{\Delta,p} \end{cases} \tag{6.64}$$

for each $v \in V$, and

$$\tilde{\gamma}(a) = \begin{cases} \max\{\hat{p}(\partial^- a) - \hat{p}(\partial^+ a), \gamma(a)\} & \text{if } \varphi(a) = 0 \\ \min\{\hat{p}(\partial^- a) - \hat{p}(\partial^+ a), -\gamma(a)\} & \text{if } \varphi(a) = 1 \end{cases} \tag{6.65}$$

for each $a \in A$.

Furthermore, defining $\hat{\gamma}$ by $\hat{\gamma}(a) = \lfloor \tilde{\gamma}(a)/\varepsilon \rfloor \varepsilon$ for $a \in A$ ($\lfloor x \rfloor$ is the largest integer not exceeding x), then $\Delta = (\varphi, z)$ is also 0-optimal with respect to potential $\hat{p}$ and cost $\hat{\gamma}$.

Proof: For each $a \in C_z$ with $p(\partial^+ a) - p(\partial^- a) = -\varepsilon$, we have $\partial^+ a \in D^+_{\Delta,p}$ and $\partial^- a \notin D^+_{\Delta,p}$. It follows that $\hat{p}(\partial^+ a) - \hat{p}(\partial^- a) = p(\partial^+ a) + \varepsilon - p(\partial^- a) = 0$ for such a.

For each $a \in C_z$ with $p(\partial^+ a) - p(\partial^- a) = 0$, we have by Lemma 2.1 that if $\partial^- a \in D^+_{\Delta,p}$, then $\partial^+ a \in D^+_{\Delta,p}$. In this case $\hat{p}(\partial^+ a) - \hat{p}(\partial^- a) = p(\partial^+ a) + \varepsilon - (p(\partial^- a) + \varepsilon) = 0$. If $\partial^- a \notin D^+_{\Delta,p}$, then $\hat{p}(\partial^+ a) - \hat{p}(\partial^- a) \geq p(\partial^+ a) - p(\partial^- a) = 0$.

For each $a \in C_z$ with $p(\partial^+ a) - p(\partial^- a) \geq \varepsilon$, we have $\hat{p}(\partial^+ a) - \hat{p}(\partial^- a) \geq p(\partial^+ a) - \hat{p}(\partial^- a) \geq p(\partial^+ a) - p(\partial^- a) - \varepsilon \geq 0$.

Therefore, for any $a \in C_z$, we have $\hat{p}(\partial^+ a) - \hat{p}(\partial^- a) \geq 0$.

For each $a \in A_\varphi$ ($\varphi(a) = 0$) we have

$$\begin{aligned} &\tilde{\gamma}_\Delta(a) + \hat{p}(\partial^+ a) - \hat{p}(\partial^- a) \\ &= \max\{\hat{p}(\partial^- a) - \hat{p}(\partial^+ a), \gamma(a)\} + \hat{p}(\partial^+ a) - \hat{p}(\partial^- a) \\ &\geq 0. \end{aligned} \tag{6.66}$$

For each $a \in B_\varphi$ ($\varphi(\bar{a}) = 1$) we have

$$\begin{aligned}
&\hat{\gamma}_\Delta(a) + \hat{p}(\partial^+ a) - \hat{p}(\partial^- a) \\
&= -\hat{\gamma}(\bar{a}) + \hat{p}(\partial^- \bar{a}) - \hat{p}(\partial^+ \bar{a}) \\
&= -\min\{\hat{p}(\partial^- \bar{a}) - \hat{p}(\partial^+ \bar{a}), -\gamma(\bar{a})\} + \hat{p}(\partial^- \bar{a}) - \hat{p}(\partial^+ \bar{a}) \\
&\geq 0.
\end{aligned} \tag{6.67}$$

For the second part of this lemma, we note that the relabeling operations keep $p(v)/\varepsilon$ integral for each $v \in V$. This is also true for $\hat{p}$. For each $a \in A_\varphi$ ($\varphi(a) = 0$) we have $\hat{\gamma}(a) = \tilde{\gamma}(a)$ when

$$\max\{\hat{p}(\partial^- a) - \hat{p}(\partial^+ a), \gamma(a)\} = \hat{p}(\partial^- a) - \hat{p}(\partial^+ a). \tag{6.68}$$

If we have

$$\max\{\hat{p}(\partial^- a) - \hat{p}(\partial^+ a), \gamma(a)\} = \gamma(a), \tag{6.69}$$

then from (6.66) we have

$$\begin{aligned}
&\hat{\gamma}_\Delta(a) + \hat{p}(\partial^+ a) - \hat{p}(\partial^- a) \\
&= \lfloor \tilde{\gamma}(a)/\varepsilon \rfloor \varepsilon + \hat{p}(\partial^+ a) - \hat{p}(\partial^- a) \\
&= \lfloor (\tilde{\gamma}(a) + \hat{p}(\partial^+ a) - \hat{p}(\partial^- a))/\varepsilon \rfloor \varepsilon \\
&\geq 0.
\end{aligned} \tag{6.70}$$

Therefore, for each $a \in A_\varphi$ ($\varphi(a) = 0$) we have

$$\hat{\gamma}_\Delta(a) + \hat{p}(\partial^+ a) - \hat{p}(\partial^- a) \geq 0. \tag{6.71}$$

Similarly, for each $a \in B_\varphi$ ($\varphi(\bar{a}) = 1$) we have

$$\hat{\gamma}_\Delta(a) + \hat{p}(\partial^+ a) - \hat{p}(\partial^- a) \geq 0. \tag{6.72}$$

□

Procedure SuccessiveShortestPath(ε, p, Δ)
Input: A potential p and an ε-optimal submodular pseudoflow $\Delta = (\varphi, z)$ with respect to p.
Output: A potential p and a 2ε-optimal submodular flow Δ with respect to p.
Step 1: Put

$$\hat{p}(v) = \begin{cases} p(v) + \varepsilon & \text{for } v \in D^+_{\Delta,p} \\ p(v) & \text{for } v \notin D^+_{\Delta,p} \end{cases} \tag{6.73}$$

for each $v \in V$, and compute $\hat{\gamma}(a)$ for each $a \in A$ from $\tilde{\gamma}$ as defined in Lemma 6.8. Similarly as (3.29) we define $\hat{\gamma}_\Delta : A_\Delta \to \mathbf{R}$ in terms of $\hat{\gamma}$ instead of γ.

Step 2: For each $a \in A_\Delta$ let $l(a) = \hat{\gamma}_\Delta(a) + \hat{p}(\partial^+ a) - \hat{p}(\partial^- a)$ be the length of arc a. For each $v \in V$ let $\tilde{p}(v)$ be the length of a shortest path from the vertex set $S^+ = \{v \in V \mid z(v) > \partial\varphi(v)\}$ to vertex v in V. (Here, for simplicity we assume that all $\tilde{p}(v)$ $(v \in V)$ are well defined and take on finite values.) If there exists no vertex in $S^- = \{v \in V \mid z(v) < \partial\varphi(v)\}$ which is reachable from S^+, stop (there is no feasible submodular flow in $\mathcal{N}$). Otherwise go to Step 3.
Step 3: Find a shortest directed path P in $\mathcal{N}_\Delta$ from S^+ to S^- and let $w \in S^-$ be the terminal vertex of P: if more than one such path exists, choose one which consists of the fewest number of arcs. Denote the arc set of P by A_P. Put

$$\varphi(a) \leftarrow \begin{cases} 1 & \text{if } a \in A_P \\ 0 & \text{if } \bar{a} \in A_P \\ \varphi(a) & \text{otherwise} \end{cases} \tag{6.74}$$

for each $a \in A$. Also, put

$$z \leftarrow z + \sum_{a \in A_P \cap C_z} (\chi_{\partial^- a} - \chi_{\partial^+ a}) \tag{6.75}$$

and $\hat{p} \leftarrow \hat{p} + \tilde{p}$.
Step 4: If $S^+ = \emptyset$, then put $p \leftarrow \hat{p}$ and stop. Otherwise go to Step 2.
(End)

Note that at the end of Step 3 $\hat{p}(v)/\varepsilon$ is still an integer for any $v \in V$. The rest of this section is devoted to the proof of the validity of procedure SuccessiveShortestPath. The argument is similar to that of M. Iri and N. Tomizawa [18] (also see [11, Section 5.5]). It should also be noted that the validity of the infeasibility check in Step 2 can be shown by a standard proof technique as in [11, Section 5.3].

Lemma 6.9: *In Step 2, $\Delta = (\varphi, z)$ is a 0-optimal submodular pseudoflow with respect to the potential $\hat{p} + \tilde{p}$ and cost function $\hat{\gamma}$.*

Proof: By the definition of $\tilde{p}$ we have $\tilde{p}(\partial^- a) \le \tilde{p}(\partial^+ a) + l(a)$ for each $a \in A_\Delta$, i.e., $\hat{\gamma}_\Delta(a) + \tilde{p}(\partial^+ a) + \hat{p}(\partial^+ a) - (\tilde{p}(\partial^- a) + \hat{p}(\partial^- a)) \ge 0$. □

Lemma 6.10: *After an execution of Step 3 we have $z \in \mathrm{B}(f)$.*

Proof: From the definition of P we have $\tilde{p}(\partial^- a) = \tilde{p}(\partial^+ a) + l(a)$ for any $a \in A_P$. It follows that

$$\hat{p}(\partial^- a) - \hat{p}(\partial^+ a) = \hat{\gamma}_\Delta(a) \qquad (a \in A_P), \tag{6.76}$$

where $\hat{p}$ is the potential at the end of Step 3. Denote the submodular pseudoflow Δ obtained at the beginning of Step 3 by $\Delta_0 = (\varphi_0, z_0)$. Suppose that the arc set $A_P \cap C_{z_0}$ is given by $\{a_1, \cdots, a_q\}$ with $a_i = (u_i, v_i)$ $(i = 1, \cdots, q)$. Since $\hat{\gamma}_\Delta(a) = 0$

for $a \in A_P \cap C_{z_0}$, we have $\hat{p}(u_i) = \hat{p}(v_i)$ $(i = 1, \cdots, q)$ from (6.76). Also by definition, at the end of Step 3

$$z = z_0 + \sum_{i=1}^{q} (\chi_{v_i} - \chi_{u_i}). \tag{6.77}$$

Without loss of generality let u_i's and v_i's be numbered in such a way that

$$\hat{p}(u_i) = \hat{p}(v_i) \leq \hat{p}(u_j) = \hat{p}(v_j) \quad (1 \leq i < j \leq q), \tag{6.78}$$

and that if $\hat{p}(u_i) = \hat{p}(v_i) = \hat{p}(u_j) = \hat{p}(v_j)$ $(i < j)$, then a_i lies nearer to the initial vertex of path P than a_j along P. From these assumptions it is seen that there exists no arc (u_i, v_j) in C_{z_0} with $1 \leq i < j \leq q$, due to the 0-optimality and the way of selecting P. Hence, by Lemma 2.2 we have $z \in \mathrm{B}(f)$. □

Lemma 6.11: *After an execution of Step 3 Δ becomes a 0-optimal submodular flow with respect to the current potential $\hat{p}$ and cost function $\hat{\gamma}$.*

Proof: The notations are the same as in the proof of Lemma 6.10. We prove that for each $a \in A_\Delta - A_{\Delta_0}$ we have $\hat{\gamma}_\Delta(a) + \hat{p}(\partial^+ a) - \hat{p}(\partial^- a) \geq 0$. Here,

$$A_\Delta - A_{\Delta_0} = ((A_\varphi \cup B_\varphi) - (A_{\varphi_0} \cup B_{\varphi_0})) \cup (C_z - C_{z_0}). \tag{6.79}$$

For any $a \in (A_\varphi \cup B_\varphi) - (A_{\varphi_0} \cup B_{\varphi_0})$ we have $\bar{a} \in A_P$. From (6.76) we get $\hat{\gamma}_\Delta(a) + \hat{p}(\partial^+ a) - \hat{p}(\partial^- a) = 0$.

Next, consider arcs in $C_z - C_{z_0}$. Define

$$z_t = z_0 + \sum_{i=1}^{t} (\chi_{v_i} - \chi_{u_i}) \qquad (t = 1, \cdots, q). \tag{6.80}$$

Then, from Lemma 2.2 $z_t = z_{t-1} + \chi_{v_t} - \chi_{u_t}$ is in $\mathrm{B}(f)$ for each $t = 1, \cdots, q$. Note that $z_q = z$. We prove by induction on $t = 0, \cdots, q$ that for each $t = 0, \cdots, q$ and $a \in C_{z_t}$ we have $\hat{p}(\partial^+ a) - \hat{p}(\partial^- a) \geq 0$. This is true for $t = 0$ due to Lemma 6.9. Suppose that it is true for $t = k - 1$ $(1 \leq k \leq q)$. For $t = k$, let $a = (w, s) \in C_{z_t} - C_{z_{t-1}}$. From Lemma 2.3 we have

(i) $u_t = s$ or $(u_t, s) \in C_{z_{t-1}}$,

(ii) $v_t = w$ or $(w, v_t) \in C_{z_{t-1}}$.

Therefore, $\hat{p}(u_t) \geq \hat{p}(s)$ and $\hat{p}(w) \geq \hat{p}(v_t)$. It follows that $\hat{p}(w) \geq \hat{p}(s)$ since $\hat{p}(u_t) = \hat{p}(v_t)$. Hence, the induction hypothesis is true for $t = k$, which is the required conclusion. □

From Lemma 6.11, the arc length $l(a)$ defined in Step 2 is nonnegative for each $a \in A_\Delta$. Consequently, $\hat{p}(v)$ is well defined and can be computed efficiently by Dijkstra's algorithm.

Lemma 6.12: *The output Δ of procedure SuccessiveShortestPath is a 2ε-optimal submodular flow with respect to the corresponding p and γ.*

Proof: By the definitions given in Lemma 6.8 we have $|\hat{\gamma}_\Delta(a) - \gamma_\Delta(a)| \leq 2\varepsilon$ for all $a \in A_\Delta$. Hence, the present lemma follows from Lemma 6.11. □

Let p' be the input of procedure PushRelabel, and p and $\Delta = (\varphi, z)$ be the outputs. Define $x_0(v) = -|\delta^- v|$ for each $v \in V$. If Problem (P_s) has a feasible solution φ, then $x_0 \leq \partial\varphi \in B(f)$. Let $(2^V, f_{x_0})$ be the contraction of $(\mathcal{D}, f)$ by the vector x_0. Replacing f by f_{x_0} in Problem (P_s) does not change the set of all the feasible submodular flows. Since $-x_0(V) = \sum_{v \in V} |\delta^- v| = m$, putting $d = -x_0$ and $C = 1$, we have from (5.49)

$$\sum_{v \in S^+} (p'(v) - p(v))(z(v) - \partial\varphi(v)) \leq 9\varepsilon m. \tag{6.81}$$

Equation (6.81) implies that

$$L \sum_{v \in S^+} (z(v) - \partial\varphi(v)) \leq 9m. \tag{6.82}$$

In each iteration of Steps 2 and 3 in procedure SuccessiveShortestPath, the value of $\sum_{v \in S^+}(z(v) - \partial\varphi(v))$ is reduced by one. If we choose $L = O(\sqrt{m})$, then the number of such iterations is $O(\sqrt{m})$ from equation (6.82). The computation of Dijkstra's shortest path algorithm for finding $\tilde{p}$ and the required shortest directed path requires $O(n^2)$ times. Hence, SuccessiveShortestPath requires $O(\sqrt{m}n^2)$ time.

Consequently, we have from Lemmas 6.5 and 6.7

Theorem 6.13: *If we choose $L = O(\sqrt{m})$, then the complexity of the 0-1 submodular flow algorithm is $O(\sqrt{m}n^2 \log(n\Gamma))$ with oracles for the dependence function and the exchange capacity of the given submodular system.* □

References

[1] W. Cui and S. Fujishige: A primal algorithm for the submodular flow problem with minimum-mean cycle selection. *Journal of the Operations Research Society of Japan* **31** (1988) 431-440.

[2] W. H. Cunningham and A. Frank: A primal-dual algorithm for submodular flows. *Mathematics of Operations Research* **10** (1985) 251-262.

[3] J. Edmonds: Submodular functions, matroids, and certain polyhedra. *Proceedings of the Calgary International Conference on Combinatorial Structures and Their Applications* (R. Guy, H. Hanani, N. Sauer and J. Schönheim, eds., Gordon and Breach, New York, 1970), 69-87.

[4] J. Edmonds and R. Giles: A min-max relation for submodular functions on graphs. *Annals of Discrete Mathematics* **1** (1977) 185-204.

[5] A. Frank: How to make a digraph strongly connected. *Combinatorica* 1 (1981) 145-153.

[6] A. Frank: An algorithm for submodular functions on graphs. *Annals of Discrete Mathematics* **16** (1982) 189-212.

[7] A. Frank: Augmenting graphs to meet edge-connectivity requirements. *SIAM Journal on Discrete Mathematics* **5** (1992) 25-53.

[8] A. Frank and É. Tardos: An application of the simultaneous approximation in combinatorial optimization. *Combinatorica* 7 (1987) 49-65.

[9] S. Fujishige: Algorithms for solving the independent-flow problems. *Journal of the Operations Research Society of Japan* **21** (1978) 189-204.

[10] S. Fujishige: A capacity-rounding algorithm for the minimum cost circulation problem– A dual framework of the Tardos algorithm. *Mathematical Programming* **35** (1986) 298-308.

[11] S. Fujishige: *Submodular Functions and Optimization* (North-Holland, Amsterdam, 1991).

[12] S. Fujishige, H. Röck and U. Zimmermann: A strongly polynomial algorithm for minimum cost submodular flow problems. *Mathematics of Operations Research* **14** (1989) 60-69.

[13] S. Fujishige and X. Zhang: New algorithms for the intersection problem of submodular systems. *Japan Journal of Industrial and Applied Mathematics* **9** (1992) 369-382.

[14] S. Fujishige and X. Zhang: An efficient cost scaling algorithm for the independent assignment problem. *Journal of the Operations Research Society of Japan* **38** (1995) 124-136.

[15] H. N. Gabow: A framework for cost-scaling algorithms for submodular flow problems. Technical Report No. CU-CS-661-93, Computer Science Department, University of Colorado, (1993).

[16] D. Gale: A theorem of flows in networks. *Pacific Journal of Mathematics* **7** (1957) 1073-1082.

[17] A. V. Goldberg and R. E. Tarjan: Finding minimum-cost circulations by successive approximation. *Mathematics of Operations Research* **15** (1990) 430-466.

[18] M. Iri and N. Tomizawa: An algorithm for finding an optimal "independent assignment". *Journal of the Operations Research Society of Japan* 19 (1976) 32-57.

[19] C. Lucchesi and D. H. Younger: A minimax relation for directed graphs. *Journal of the London Mathematical Society* 17 (1978) 369-374.

[20] M. Shigeno and S. Iwata: A cost-scaling algorithm for 0-1 submodular flows. *Discrete Applied Mathematics* (to appear).

[21] C. Wallacher and U. Zimmermann: A polynomial cycle canceling algorithm for submodular flows. Preprint (1994).

[22] X. Zhang: Algorithms for network flow problems with matroidal and submodular constraints. Ph.D. Dissertation, Institute of Socio-Economic Planning, University of Tsukuba (1994).

[23] U. Zimmermann: Negative circuits for flows and submodular flows. *Discrete Applied Mathematics* 36 (1992) 179-189.