COMBINATORICA
Bolyai Society – Springer-Verlag

COMBINATORICA 23 (3) (2003) 503–525

FAST CYCLE CANCELING ALGORITHMS FOR MINIMUM COST SUBMODULAR FLOW*

SATORU IWATA†, S. THOMAS MCCORMICK‡, MAIKO SHIGENO§

Received September 15, 1999

This paper presents two fast cycle canceling algorithms for the submodular flow problem. The first uses an assignment problem whose optimal solution identifies most negative node-disjoint cycles in an auxiliary network. Canceling these cycles lexicographically makes it possible to obtain an optimal submodular flow in $O(n^4 h \log(nC))$ time, which almost matches the current fastest weakly polynomial time for submodular flow (where n is the number of nodes, h is the time for computing an exchange capacity, and C is the maximum absolute value of arc costs). The second algorithm generalizes Goldberg's cycle canceling algorithm for min cost flow to submodular flow to also get a running time of $O(n^4 h \log(nC))$. We show how to modify these algorithms to make them strongly polynomial, with running times of $O(n^6 h \log n)$, which matches the fastest strongly polynomial time bound for submodular flow. We also show how to extend both algorithms to solve submodular flow with separable convex objectives.

1. Introduction

The submodular flow problem, introduced by Edmonds and Giles [4], is one of the most important frameworks of efficiently solvable combinatorial optimization problems. It includes the minimum cost flow, the graph orientation,

Mathematics Subject Classification (2000): 90C27, 90C35, 90B10, 90C25

* An extended abstract of a preliminary version of part of this paper appeared in [22].

† Research supported in part by a Grant-in-Aid of the Ministry of Education, Science, Sports and Culture of Japan.

‡ Research supported by an NSERC Operating Grant. Part of this research was done during a sabbatical leave at Cornell SORIE.

§ Research supported in part by a Grant-in-Aid of the Ministry of Education, Science, Sports and Culture of Japan.

0209–9683/103/$6.00

the polymatroid intersection, and the directed cut covering problems as its special cases. Other frameworks named independent flows [10] and polymatroid network flows [18,28] are equivalent to submodular flows. These three are collectively called *neoflows* in [11]. Section 2 formalizes our notation for submodular flow.

To be able to talk about running times, define n and m as the number of nodes and arcs in an instance of submodular flow, C as the largest absolute value of the arc costs, and U as the largest absolute value of any bound. A number of combinatorial algorithms have been proposed for submodular flow, all of which rely on an oracle for exchange capacities, as generalizations of minimum cost flow algorithms. Define h as the time for computing an exchange capacity. Cunningham and Frank [3] developed a primal-dual algorithm and its cost scaling version, which is the first combinatorial algorithm with weakly polynomial time complexity. This algorithm solves $\mathrm{O}(m \log C)$ maximum submodular flow problems. In [24] we show how to speed this up to an $\mathrm{O}(n^4 h \log C)$ algorithm, the current fastest submodular flow algorithm when C is not too big.

One reasonable strategy for designing submodular flow algorithms is to extend the capacity scaling minimum cost flow algorithm due to Edmonds and Karp [5]. Since rounding a submodular function may destroy its submodularity, we need something more than a straightforward scaling scheme. Iwata [20] shows that adding a small but strictly submodular function before rounding makes it possible to get a weakly polynomial capacity scaling algorithm for the submodular flow problem. This scaling scheme calls a maximum submodular flow subroutine $\mathrm{O}(n^4 h \log U)$ times, which is slower than the Cunningham–Frank algorithm. This algorithm is, however, the first polynomial algorithm for submodular flow with separable convex costs. In [23] we develop a cut canceling algorithm that is the dual of Tight Arc Cycle Canceling algorithm of this paper, in the sense that they both come from extensions of a dual pair of min cost flow algorithms from [33]. The paper [6] shows how to improve the capacity scaling algorithm to an $\mathrm{O}(n^4 h \log U)$ algorithm, which can also be seen as an improvement of the algorithm in [23].

The first strongly polynomial algorithm for the submodular flow problem is due to Frank and Tardos [8]. This is an application of simultaneous Diophantine approximation and substantially generalizes the first strongly polynomial minimum cost flow algorithm of Tardos [34]. A more direct generalization of the Tardos algorithm to the submodular flow problem is described by Fujishige, Röck, and Zimmermann [12]. The latter algorithm combined with the improved primal-dual algorithm in [24] runs in $\mathrm{O}(n^6 h \log n)$ time, which is currently the best strongly polynomial bound.

The strategy of this paper is to extend cycle canceling algorithms for min cost flow (see [33] for a survey) to submodular flow. Cui and Fujishige [2] extended the minimum-mean cycle canceling algorithm of Goldberg and Tarjan [15], but the running time is only pseudopolynomial. Wallacher and Zimmermann [36] improved a pseudopolynomial minimum ratio cycle canceling algorithm of Zimmermann [37] to obtain the time bound $\mathrm{O}(n^8 h \log(nCU))$.

Section 3 develops two fast polynomial cycle canceling algorithms for submodular flow. The first is *Tight Arc Cycle Canceling*, which is an extension of our relaxed most negative disjoint family of cycles canceling algorithm for minimum cost flow [33]. The second is *Negative Arc Cycle Canceling*, which is an extension of Goldberg's cycle canceling algorithm [14].

Both algorithms use Goldberg and Tarjan's generic successive approximation framework [16] where we scale a relaxation parameter. The key idea for the first algorithm is to use an assignment subproblem whose optimal solution corresponds to a most negative family of node-disjoint cycles in an auxiliary network. Given an optimal dual solution of the assignment problem, we extract the tight arcs, and repeatedly cancel cycles of those arcs. Lexicographical ordering makes it possible to obtain an optimal submodular flow in a polynomial number of cancellations. The key idea for the second algorithm is to cancel cycles in the subnetwork of negative reduced cost arcs until it is acyclic. At this point we can make a change to dual variables that removes at least one node from participating in future negative cycles. Both algorithms achieve a running time bound $\mathrm{O}(n^4 h \log(nC))$, nearly matching the fastest current weakly polynomial bound.

Next, Section 4 shows that we can modify these algorithms using the standard technique of canceling a min mean cycle every $\mathrm{O}(\log n)$ scaling phases so that they become strongly polynomial. The running times in this case are $\mathrm{O}(n^6 h \log n)$, matching the fastest current bound.

Finally, Section 5 considers a further extension of our algorithms to submodular flows with separable convex objectives. It is not apparent how to straightforwardly extend the primal-dual algorithm to this case, since it is not clear what the bit scaling of a convex function means. Scaling in the separable convex case fits in much more naturally with the cycle canceling algorithms of this paper since here it is necessary only to scale the relaxation parameter. We show that our algorithms have polynomial convergence for this case.

2. Submodular Flow

Let V be a finite set. A pair of subsets X and Y of V is said to be *crossing* if $X \cap Y \neq \emptyset$, $X - Y \neq \emptyset$, $Y - X \neq \emptyset$, and $X \cup Y \neq V$. A family $\mathcal{F} \subseteq 2^V$ is called a *crossing family* if $X \cap Y \in \mathcal{F}$ and $X \cup Y \in \mathcal{F}$ hold for any crossing pair of X and Y in $\mathcal{F}$.

Let $\mathcal{F} \subseteq 2^V$ be a crossing family with $\emptyset, V \in \mathcal{F}$. A function $f : \mathcal{F} \to \mathbf{R}$ is said to be *submodular* if it satisfies

$$f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y)$$

for any crossing pair of X and Y in $\mathcal{F}$. For a submodular function f on $\mathcal{F}$ with $f(\emptyset) = 0$, the *base polyhedron* $\mathrm{B}(f)$ in $\mathbf{R}^V$ is defined by

$$\mathrm{B}(f) = \{x \mid x(V) = f(V),\ \forall X \in \mathcal{F} : x(X) \leq f(X)\}.$$

A vector in $\mathrm{B}(f)$ is called a *base*. Note that the base polyhedron $\mathrm{B}(f)$ may possibly be empty. The bi-truncation algorithm of Frank–Tardos [9] efficiently finds a base if exists, and proves emptiness otherwise (see also Naitoh–Fujishige [29]).

For any base $x \in \mathrm{B}(f)$ and distinct $v, w \in V$, we define the *exchange capacity* $\widetilde{r}(x, v, w)$ by

$$\widetilde{r}(x, v, w) = \max\{\alpha \mid x + \alpha(\chi_v - \chi_w) \in \mathrm{B}(f)\}.$$

We can equivalently write this as

$$\widetilde{r}(x, v, w) = \min\{f(X) - x(X) \mid v \in X \in \mathcal{F},\ w \notin X\}.$$

Computing an exchange capacity is a submodular function minimization on a distributive lattice, which can be done via the ellipsoid method [17] or by combinatorial methods [21,32] in strongly polynomial time. As is often the case with combinatorial algorithms for the submodular flow problem, we assume in this paper that an oracle for computing exchange capacities is available, with running time h.

The directed graph $D_x = (V, E_x)$ with the arc set $E_x = \{(w, v) \mid \widetilde{r}(x, v, w) > 0\}$ is called the *exchangeability graph*. An arc in E_x is sometimes called a *jumping arc*. The following lemmas are fundamental for developing combinatorial submodular flow algorithms.

Lemma 2.1 **([11, Lemma 4.5]).** *Let $(v_1, w_1), (v_2, w_2), \cdots, (v_q, w_q)$ be node-disjoint arcs in D_x for $x \in \mathrm{B}(f)$ such that $i < j$ implies $(v_i, w_j) \notin E_x$. If $\alpha > 0$ satisfies $\alpha \leq \widetilde{r}(x, w_i, v_i)$ for $i = 1, 2, \cdots, q$, then the vector $y = x + \alpha \sum_{i=1}^{q} (\chi_{w_i} - \chi_{v_i})$ is also in $\mathrm{B}(f)$.* ∎

Lemma 2.2. *Suppose $y \in \mathrm{B}(f)$ is obtained from $x \in \mathrm{B}(f)$ by $y = x + \alpha(\chi_v - \chi_w)$ with $0 < \alpha \le \tilde{r}(x,v,w)$ and that $(s,t) \in E_y$ in spite of $(s,t) \notin E_x$. Then $(w,t) \in E_x$ if $t \neq w$, and $(s,v) \in E_x$ if $s \neq v$.* ∎

Proof. Since $(s,t) \notin E_x$ and $(s,t) \in E_y$, there exists an $X \in \mathcal{F}$ such that $t \in X$, $s \notin X$, $w \in X$, $v \notin X$ and $x(X) = f(X)$. If $t \neq w$ and $(w,t) \notin E_x$, there exists a $Y \in \mathcal{F}$ such that $t \in Y$, $w \notin Y$, $v \notin Y$ and $x(Y) = f(Y)$. Thus $X \cap Y \neq \emptyset$ and $X \cup Y \neq V$. By the submodularity of f, we have $y(X \cap Y) = x(X \cap Y) = f(X \cap Y)$. Since $t \in X \cap Y$ and $s \notin X \cap Y$, this implies $(s,t) \in E_x$, a contradiction. Similarly, if $s \neq v$ and $(s,v) \notin E_x$, there exists a $Z \in \mathcal{F}$ such that $v \in Z$, $s \notin Z$, $w \in Z$ and $x(Z) = f(Z)$. Thus $X \cap Z \neq \emptyset$ and $X \cup Z \neq V$. By the submodularity of f, we have $y(X \cup Z) = x(X \cup Z) = f(X \cup Z)$. Since $t \in X \cup Z$ and $s \notin X \cup Z$, this implies $(s,t) \in E_x$, a contradiction. ∎

Let $G = (V, A)$ be a directed graph with a node set V of cardinality n and an arc set A of cardinality m. We assume no parallel arcs, so that $m = \mathrm{O}(n^2)$. For a node $v \in V$, we denote by $\delta^+ v$ and $\delta^- v$, respectively, the set of arcs leaving v and those entering v. The *boundary* $\partial\varphi$ of a function $\varphi : A \to \mathbf{R}$ on node $v \in V$ is defined by

$$\partial\varphi(v) = \sum_{a \in \delta^+ v} \varphi(a) - \sum_{a \in \delta^- v} \varphi(a),$$

which is the net flow leaving v. We are given *upper* and *lower capacities* (or *bounds*) $u : A \to \mathbf{R}$ and $l : A \to \mathbf{R}$, respectively, *costs* $c : A \to \mathbf{Z}$, and a submodular function $f : \mathcal{F} \to \mathbf{R}$ with $f(\emptyset) = f(V) = 0$. Then the submodular flow problem is:

$$\begin{aligned} \text{Minimize } & \sum_{a \in A} c(a)\varphi(a) \\ \text{subject to } & l(a) \le \varphi(a) \le u(a) \quad (a \in A) \\ & \partial\varphi \in \mathrm{B}(f). \end{aligned}$$

Given a base in $\mathrm{B}(f)$, we can efficiently check the feasibility of this problem [7,35,13]. In particular, a variant of the Fujishige–Zhang algorithm [13,24] runs in $\mathrm{O}(n^3 h)$ time.

Let $\bar{a}$ denote the reversal of a. We associate with a feasible flow $\varphi : A \to \mathbf{R}$ an auxiliary network $\mathcal{N}_\varphi = (G_\varphi, r_\varphi, c)$, where $G_\varphi = (V, A_\varphi)$ is a directed graph with the node set V and the arc set $A_\varphi = F_\varphi \cup B_\varphi \cup E_{\partial\varphi}$ given by

$$\begin{aligned} F_\varphi &= \{a \mid a \in A, \varphi(a) < u(a)\}, \\ B_\varphi &= \{\bar{a} \mid a \in A, \varphi(a) > l(a)\}, \\ E_{\partial\varphi} &= \{(w,v) \mid \tilde{r}(\partial\varphi, v, w) > 0\}. \end{aligned}$$

The arcs in $F_\varphi \cup B_\varphi$ are collectively called *residual* arcs. The residual capacity function $r_\varphi : A_\varphi \to \mathbf{R}$ and cost function $c : A_\varphi \to \mathbf{Z}$ are defined by

$$r_\varphi(a) = \begin{cases} u(a) - \varphi(a) & (a \in F_\varphi), \\ \varphi(\bar{a}) - l(\bar{a}) & (a \in B_\varphi), \\ \tilde{r}(\partial\varphi, v, w) & (a = (w, v) \in E_{\partial\varphi}), \end{cases}$$

and

$$c(a) = \begin{cases} c(a) & (a \in F_\varphi), \\ -c(\bar{a}) & (a \in B_\varphi), \\ 0 & (a \in E_{\partial\varphi}). \end{cases}$$

The cost of a cycle Q in G_φ is $c(Q) = \sum_{a \in Q} c(a)$, and we call Q a *negative cycle* if $c(Q) < 0$. We denote by $\partial^+ a$ and $\partial^- a$, respectively, the tail and head of the arc a. Given a set of *node potentials* $p \in \mathbf{R}^V$, define the *reduced cost* of arc a w.r.t. p as $c_p(a) = c(a) + p(\partial^+ a) - p(\partial^- a)$. If there are no negative cycles in G_φ then we can compute shortest path distances $p \in \mathbf{R}^V$ such that $c_p(a) \geq 0$ for all $a \in A_\varphi$. It is well-known that a flow in a min cost flow network is optimal if and only if there are no negative cost directed cycles in its residual network. A similar statement holds true for submodular flow:

Theorem 2.3 (**[11, Theorem 5.3]**). *A submodular flow φ is optimal if and only if $c(Q) \geq 0$ for all directed cycles Q in G_φ, which is true if and only if there exist node potentials $p \in \mathbf{R}^V$ such that $c_p(a) \geq 0$ for all $a \in A_\varphi$.* ∎

3. Two Fast Cycle-Canceling Algorithms

3.1. Overview

This section presents two fast cycle-canceling algorithm for submodular flow problems. Both algorithms use the Successive Approximation framework of Goldberg–Tarjan [16]. In this framework we relax some aspect of an optimality condition for the problem by a parameter ϵ, here that $c_p(a) \geq 0$ for all $a \in A_\varphi$. Flow φ satisfying the relaxed condition is called *ϵ-optimal*. In these cases we can start with $\epsilon = \mathrm{O}(C)$ (not too large), and once $\epsilon < 1/n$ (reasonably small) we know we are optimal. To make this idea work, it is necessary to invent a Refine subroutine that will take as input a 2ϵ-optimal solution and which will output an ϵ-optimal solution. One call to Refine is an *ϵ-scaling phase*, and now we can halve ϵ and repeat. Thus we will have $\mathrm{O}(\log(nC))$ scaling phases. Since the two algorithms share many ideas, we present them in parallel instead of serially.

The two algorithms use different relaxations. The first notion is a generalization of an idea from the relaxed cycle canceling algorithm from our min cost flow paper [33]: Use an assignment problem with scaled and rounded costs that is a relaxation of the (NP Hard) problem of finding a most negative augmenting cycle to find a most negative *family* of cycles, and cancel the resulting cycles. Since each canceled cycle consists of tight arcs, we call this relaxation *Tight Arc Cycle Canceling*. The second is a generalization of an idea from Goldberg's cycle canceling algorithm for min cost flow [14]. It more straightforwardly relaxes the condition $c_p(a) \geq 0$ to $c_p(a) \geq -\epsilon$ for all $a \in A_\varphi$. We call this relaxation *Negative Arc Cycle Canceling*.

Both algorithms generate a subgraph of G_φ in which we must cancel cycles until the subgraph is acyclic. We use the same Cycle Cancel subroutine in both cases, which will take $O(n^3 h)$ time. At this point the acyclicity of the subgraph implies that we are able to update the dual variables so as to make progress. For Tight Arc Cycle Canceling the value of the assignment problem increases by at least ϵ at each dual update; since the initial assignment value is at least $-2n\epsilon$, we call Cycle Cancel $O(n)$ times per phase, giving us $O(n^4 h)$ work per scaling phase. For Negative Arc Cycle Canceling, each dual update removes at least one node from the set of nodes which are the heads of arcs with reduced cost less than $-\epsilon$, so we again call Cycle Cancel $O(n)$ times per phase, giving us $O(n^4 h)$ work per scaling phase. Thus the total time bound for both algorithms is $O(n^4 h \log(nC))$.

3.2. The Two Relaxations

3.2.1. Tight Arc Cycle Canceling

We construct a bipartite graph $\widehat{G}_\varphi = (V^+, V^-; H_\varphi)$. The node sets V^+ and V^- are copies of V. We denote the two copies of $v \in V$ by $v^+ \in V^+$ and $v^- \in V^-$. The edge set H_φ is given by $H_\varphi = A_\varphi \cup H^\pm$, where $H^\pm = \{(v^+, v^-) \mid v \in V\}$ and $a = (v, w) \in A_\varphi$ is now regarded as an edge (v^+, w^-). We denote by $\mathrm{AP}(\varphi)$ the assignment problem on $\widehat{G}_\varphi$ with edge cost $c(a)$ for $a \in A_\varphi$ and zero for $a \in H^\pm$. A perfect matching in $\widehat{G}_\varphi$ corresponds to a (possibly empty) node-disjoint family of cycles in G_φ. Note that the optimal value of $\mathrm{AP}(\varphi)$ is non-positive, because $H^\pm$ gives a perfect matching of cost zero. When Theorem 2.3 is specialized to this case it becomes:

Theorem 3.1. *A submodular flow φ is optimal if and only if the optimal value of $\mathrm{AP}(\varphi)$ is zero.* ∎

For a parameter $\epsilon > 0$, let $\mathrm{AP}(\varphi,\epsilon)$ denote the assignment problem on $\widehat{G}_\varphi$ obtained from $\mathrm{AP}(\varphi)$ by replacing the edge cost $c(a)$ by $c^\epsilon(a) = \epsilon\lfloor c(a)/\epsilon\rfloor + \epsilon$ for every $a \in A_\varphi$.

The linear programming dual to $\mathrm{AP}(\varphi,\epsilon)$ is

$$\begin{array}{lll} \text{Maximize} & \sum_{v\in V} p(v^+) + \sum_{w\in V} p(w^-) & \\ \text{subject to} & p(v^+) + p(w^-) \le c^\epsilon(a) & (a = (v,w) \in A_\varphi), \\ & p(v^+) + p(v^-) \le 0 & (v \in V), \end{array}$$

which we denote by $\mathrm{DAP}(\varphi,\epsilon)$. Given a feasible solution p for $\mathrm{DAP}(\varphi,\epsilon)$ of negative objective value, we construct a subgraph $G^\circ_\varphi = (V, A^\circ_\varphi)$ of G_φ with the arc set A°_φ that corresponds to the set of tight arcs, i.e.,

$$A^\circ_\varphi = \{a \mid a = (w,v) \in A_\varphi,\ p(w^+) + p(v^-) = c^\epsilon(a)\}.$$

The Tight Arc Cycle Canceling will cancel cycles in G°_φ until the optimal value of $\mathrm{AP}(\varphi,\epsilon)$ is zero. We say that φ is *TA ϵ-optimal* if $\mathrm{AP}(\varphi,\epsilon)$ has no negative-cost assignment, which is a relaxed version of the optimality condition given in Theorem 3.1. If the optimal value of $\mathrm{DAP}(\varphi,\epsilon)$ is zero, the optimal solution p proves the TA ϵ-optimality of φ.

We define $F^\circ_\varphi = A^\circ_\varphi \cap F_\varphi$, $B^\circ_\varphi = A^\circ_\varphi \cap B_\varphi$, and $E^\circ_{\partial\varphi} = A^\circ_\varphi \cap E_{\partial\varphi}$.

3.2.2. Negative Arc Cycle Canceling

Let $G^-_\varphi = (V, A^-_\varphi)$ denote the subgraph of G_φ with the arc set A^-_φ that consists of the negative reduced cost residual arcs and the zero reduced cost jumping arcs in G_φ. Namely, $A^-_\varphi = F^-_\varphi \cup B^-_\varphi \cup E^-_{\partial\varphi}$ where

$$\begin{aligned} F^-_\varphi &= \{a \mid a \in F_\varphi,\ c_p(a) < 0\}, \\ B^-_\varphi &= \{a \mid a \in B_\varphi,\ c_p(a) < 0\}, \\ E^-_{\partial\varphi} &= \{a \mid a = (w,v) \in E_{\partial\varphi},\ p(w) = p(v)\}. \end{aligned}$$

We say that φ is *NA ϵ-optimal* if there is a potential p such that $p(v)$ is an integer multiple of ϵ for every $v \in V$, every residual arc $a \in F_\varphi \cup B_\varphi$ satisfies $c_p(a) \ge -\epsilon$, and $p(w) \ge p(v)$ holds for every jumping arc $(w,v) \in E_{\partial\varphi}$. Negative Arc Cycle Canceling will cancel cycles in G^-_φ until we get an ϵ-optimal submodular flow φ.

3.3. Canceling Admissible Cycles

To avoid duplication, we use the shorthand $G^\ominus_\varphi$ to stand for G°_φ or G^-_φ, and similarly $E^\ominus_{\partial\varphi}$, etc. The proof of the following lemma is similar to [1, Lemma 10.2].

Lemma 3.2. *Any submodular flow is both TA and NA C-optimal, and a submodular flow φ is optimal if φ is TA or NA ϵ-optimal for $\epsilon < 1/n$.* ∎

A pair of arcs is said to be *consecutive* if the head of one arc coincides with the tail of the other. In both cases we need to avoid consecutive jumping arcs.

Lemma 3.3. *There is no consecutive pair of jumping arcs in $E^{\circ}_{\partial\varphi}$.*

Proof. Suppose, to the contrary, $(z,v) \in E^{\circ}_{\partial\varphi}$ and $(v,w) \in E^{\circ}_{\partial\varphi}$. Then p satisfies $p(z^+)+p(v^-)=\epsilon$, $p(v^+)+p(w^-)=\epsilon$, and $p(v^+)+p(v^-)\leq 0$. Therefore we have $p(z^+)+p(w^-)\geq 2\epsilon$. It follows from the feasibility of p for DAP(φ,ϵ) that $(z,w)\notin E_{\partial\varphi}$, which means there exists $X\in\mathcal{D}$ with $z\notin X$, $w\in X$, and $f(X)=\partial\varphi(X)$. This contradicts either $(z,v)\in E_{\partial\varphi}$ or $(v,w)\in E_{\partial\varphi}$, according to $v\in X$ or $v\notin X$. ∎

We call a cycle in G^-_{φ} *legal* if it contains no consecutive jumping arcs. The following lemma implies that if there is no legal cycle then G^-_{φ} is acyclic.

Lemma 3.4. *If $(z,v)\in E^-_{\partial\varphi}$ and $(v,w)\in E^-_{\partial\varphi}$, then $(z,w)\in E^-_{\partial\varphi}$.*

Proof. Since $p(z) = p(v) = p(w)$, it suffices to prove that $(z,w) \in E_{\partial\varphi}$. Suppose to the contrary that $(z,w) \notin E_{\partial\varphi}$. Then there exists $X \in \mathcal{D}$ with $z \notin X$, $w \in X$, and $f(X) = \partial\varphi(X)$. This contradicts either $(z,v) \in E_{\partial\varphi}$ or $(v,w)\in E_{\partial\varphi}$, according to $v\in X$ or $v\notin X$. ∎

We say an AP potential $p \in \mathbf{R}^{V^+\cup V^-}$ is *TA feasible* for $x \in \mathrm{B}(f)$ if it satisfies $p(w^+) + p(z^-) \leq \epsilon$ for every $(w,z) \in E_x$ and $p(v^+) + p(v^-) \leq 0$ for every $v\in V$, i.e., if p is feasible for the jumping arcs and $H^{\pm}$ arcs of DAP. We say a potential $p\in\mathbf{R}^V$ is *NA feasible* for $x\in\mathrm{B}(f)$ if it satisfies $p(w)\geq p(z)$ for every $(w,z)\in E_x$. To avoid special cases in the subsequent lemmas, we use $\overline{E}^{\ominus}_x$ to represent $E^{\ominus}_x$ plus the set of self-loops, i.e., $\overline{E}^{\ominus}_x = E^{\ominus}_x \cup \{(v,v)\,|\,v\in V\}$.

Lemma 3.5. *Suppose p is TA (resp. NA) feasible for $x \in \mathrm{B}(f)$ and let $y = x + \alpha(\chi_v - \chi_w)$ with $0 < \alpha \leq \tilde{r}(x,v,w)$ for $(w,v) \in E^{\circ}_x$ (resp. E^-_x). Then p is TA (resp. NA) feasible for y. If $(s,t) \in E^{\ominus}_y$ in spite of $(s,t) \notin E^{\ominus}_x$, then $(w,t)\in\overline{E}^{\ominus}_x$ and $(s,v)\in\overline{E}^{\ominus}_x$.*

Proof. Suppose that $(s,t)\in E_y$ while $(s,t)\notin E_x$. It follows from Lemma 2.2 that $(w,t)\in E_x$ or $w=t$ and that $(s,v)\in E_x$ or $s=v$.

For the case of TA feasible, since p is feasible for x, we have $p(w^+) + p(t^-) \leq \epsilon$ and $p(s^+) + p(v^-) \leq \epsilon$, which together with $p(w^+) + p(v^-) = \epsilon$ imply $p(s^+) + p(t^-) \leq \epsilon$. Thus p is feasible for y. If we further suppose

$(s,t) \in E^{\circ}_{y}$, we must have $p(w^{+})+p(t^{-})=\epsilon$ and $p(s^{+})+p(v^{-})=\epsilon$, which imply $(w,t)\in E^{\circ}_{x}\subseteq \overline{E}^{\circ}_{x}$ and $(s,v)\in E^{\circ}_{x}\subseteq \overline{E}^{\circ}_{x}$.

For the case of NA feasible, since p is feasible for x, we have $p(w)\geq p(t)$ and $p(s)\geq p(v)$, which together with $p(w)=p(v)$ imply $p(s)\geq p(t)$. Thus p is feasible for y. If we further suppose $(s,t)\in E^{-}_{y}$, we must have $p(w)=p(t)$ and $p(s)=p(v)$, which imply $(w,t)\in \overline{E}^{-}_{x}$ and $(s,v)\in \overline{E}^{-}_{x}$. ∎

A cycle Q in G°_{φ}, or a legal cycle in G^{-}_{φ}, is called *admissible* if it allows a numbering (w_1,v_1), (w_2,v_2), $\cdots,(w_q,v_q)$ of the arcs in $Q\cap E^{\ominus}_{\partial\varphi}$ such that $i<j$ implies $(w_i,v_j)\notin E^{\ominus}_{\partial\varphi}$. We cancel an admissible cycle Q by computing the *step length* $\alpha=\min\{r_{\varphi}(a)\,|\,a\in Q\}$ and modifying φ as

$$\varphi'(a)=\begin{cases}\varphi(a)+\alpha & (a\in Q\cap F^{\ominus}_{\varphi})\\ \varphi(a)-\alpha & (\bar{a}\in Q\cap B^{\ominus}_{\varphi})\\ \varphi(a) & (\text{otherwise}).\end{cases}$$

Lemma 3.6. *If φ' is obtained from φ by canceling an admissible cycle Q in $G^{\ominus}_{\varphi}$, then $F^{\ominus}_{\varphi'}\subseteq F^{\ominus}_{\varphi}$ and $B^{\ominus}_{\varphi'}\subseteq B^{\ominus}_{\varphi}$ hold.*

Proof. For the G°_{φ} case, suppose to the contrary that $a=(w,v)\notin F^{\circ}_{\varphi}\cup B^{\circ}_{\varphi}$ and $a\in F^{\circ}_{\varphi'}\cup B^{\circ}_{\varphi'}$. Then $\bar{a}\in Q\subseteq A^{\circ}_{\varphi}$, which implies $p(v^{+})+p(w^{-})=\epsilon\lfloor c(\bar{a})/\epsilon\rfloor+\epsilon$. Since $p(w^{+})+p(w^{-})\leq 0$ and $p(v^{+})+p(v^{-})\leq 0$, we have $p(w^{+})+p(v^{-})\leq \epsilon\lceil c(a)/\epsilon\rceil-\epsilon<\epsilon\lfloor c(a))/\epsilon\rfloor+\epsilon$, a contradiction to $a\in F^{\circ}_{\varphi'}\cup B^{\circ}_{\varphi'}$.

For the G^{-}_{φ} case, suppose to the contrary that $a\notin F^{-}_{\varphi}\cup B^{-}_{\varphi}$ and $a\in F^{-}_{\varphi'}\cup B^{-}_{\varphi'}$. Then $\bar{a}\in Q\subseteq A^{-}_{\varphi}$, which implies $c_p(\bar{a})<0$, and hence $c_p(a)>0$, a contradiction to $a\in F^{-}_{\varphi'}\cup B^{-}_{\varphi'}$. ∎

Lemma 3.7. *Suppose φ' is obtained from φ by canceling an admissible cycle Q in G°_{φ} (resp. G^{-}_{φ}). Then φ' is a submodular flow, and p remains TA feasible (resp. NA feasible).*

Proof. Since Q is admissible, there is a numbering (w_1,v_1), (w_2,v_2), $\ldots$, (w_q,v_q) of the arcs in $Q\cap E^{\ominus}_{\partial\varphi}$ such that $i<j$ implies $(w_i,v_j)\notin E^{\ominus}_{\partial\varphi}$. These arcs in $Q\cap E^{\circ}_{\partial\varphi}$ (resp. $Q\cap E^{-}_{\partial\varphi}$) are node-disjoint by Lemma 3.3 (resp. by legality).

For the TA feasible case, note that $p(w^{+})+p(v^{-})=\epsilon$ holds for every arc $(w,v)\in E^{\circ}_{\partial\varphi}$. If $p(w_i^{+})>p(w_j^{+})$, we have $p(w_i^{+})+p(v_j^{-})>\epsilon$, which implies $(w_i,v_j)\notin E_{\partial\varphi}$ by the TA feasibility of p. Therefore we can renumber those arcs so that $i<j$ implies $p(w_i^{+})\geq p(w_j^{+})$ in addition to $(w_i,v_j)\notin E^{\circ}_{\partial\varphi}$.

If $p(w_i^+) > p(w_j^+)$, we have shown $(w_i, v_j) \notin E_{\partial\varphi}$. On the other hand, if $p(w_i^+) = p(w_j^+)$ and $(w_i, v_j) \notin E^\circ_{\partial\varphi}$, then we have $(w_i, v_j) \notin E_{\partial\varphi}$ from the definition of $E^\circ_{\partial\varphi}$. Thus $i < j$ implies $(w_i, v_j) \notin E_{\partial\varphi}$.

Therefore $y_k = \partial\varphi + \alpha\sum_{i=1}^k(\chi_{v_i} - \chi_{w_i}) \in \mathrm{B}(f)$ holds for $k = 1, \cdots, q$ by Lemma 2.1. Hence φ' is a submodular flow. It follows from repeated applications of Lemma 3.5 that p is TA feasible for $\partial\varphi'$, which together with the proof of Lemma 3.6 implies that p is feasible to $\mathrm{DAP}(\varphi', \epsilon)$.

For the NA feasible case, note that $p(w) = p(v)$ holds for every arc $(w, v) \in E^-_{\partial\varphi}$. If $p(w_i) < p(w_j)$, we have $p(w_i) < p(v_j)$, which implies $(w_i, v_j) \notin E_{\partial\varphi}$ by the NA feasibility of p. Therefore we may renumber those arcs so that $i < j$ implies $p(w_i) \leq p(w_j)$ in addition to $(w_i, v_j) \notin E^-_{\partial\varphi}$.

If $p(w_i) < p(w_j)$, we have shown $(w_i, v_j) \notin E_{\partial\varphi}$. On the other hand, if $p(w_i) = p(w_j)$ and $(w_i, v_j) \notin E^-_{\partial\varphi}$, then we have $(w_i, v_j) \notin E_{\partial\varphi}$ from the definition of $E^-_{\partial\varphi}$. Thus $i < j$ implies $(w_i, v_j) \notin E_{\partial\varphi}$.

Therefore $y_k = \partial\varphi + \alpha\sum_{i=1}^k(\chi_{v_i} - \chi_{w_i}) \in \mathrm{B}(f)$ holds for $k = 1, \cdots, q$ by Lemma 2.1. Hence φ' is a submodular flow. It follows from repeated applications of Lemma 3.5 that p is NA feasible for $\partial\varphi'$, which together with the proof of Lemma 3.6 implies that p is feasible for $G^-_{\varphi'}$. ∎

3.4. Eligible Cycles

Following an idea of Cui and Fujishige [2], suppose that we have a fixed numbering of the nodes, i.e., $V = \{1, \cdots, n\}$. This induces the numbering of the arcs of $A^\ominus_\varphi$ defined by

$$\eta(a) = \begin{cases} 0 & (a \in F^\ominus_\varphi \cup B^\ominus_\varphi) \\ \partial^- a & (a \in E^\ominus_{\partial\varphi}). \end{cases}$$

This in turn induces a different numbering of the nodes defined by

$$\rho_\varphi(v) = \min\{\eta(a) \mid \partial^+ a = v \text{ and } a \text{ belongs to a (legal) cycle in } G^\ominus_\varphi\},$$

where we put $\rho_\varphi(v) = n+1$ if there is no cycle through v. Note that $\rho_\varphi(v)$ changes as the algorithm proceeds, while $\eta(a)$ does not. A cycle Q in $G^\ominus_\varphi$ is called an *eligible* cycle if $\rho_\varphi(\partial^+ a) = \eta(a)$ holds for every arc $a \in Q$. An eligible cycle can be found efficiently by depth-first search. For a cycle Q and nodes v, w on Q, we denote the path on Q from v to w by $Q[v, w]$.

Lemma 3.8. *An eligible cycle Q in $G^\ominus_\varphi$ is admissible.*

Proof. Suppose to the contrary that Q is not admissible. Then there exists $\{(w_j,v_j)\,|\,j=1,\cdots,q\}\subseteq Q\cap E^{\ominus}_{\partial\varphi}$ such that $(w_j,v_{j+1})\in E^{\ominus}_{\partial\varphi}$, where $v_{q+1}=v_1$. Adding (w_j,v_{j+1}) to $Q[v_{j+1},w_j]$, we obtain a cycle for each $j=1,\cdots,q$. Then from the eligibility of Q, we have $v_j<v_{j+1}$ for $j=1,\cdots,q$, which implies $v_1<v_1$, a contradiction. ∎

Thus the definition of eligible cycles provides a systematic way of selecting admissible cycles to be canceled. We now show that this selection rule leads to a polynomial bound on the number of cancellations. The following lemma will be used in the proof of Theorem 3.10, which is the key in the complexity analysis of our cycle canceling algorithm.

Lemma 3.9. *Suppose φ' is obtained from φ by canceling an eligible cycle Q in $G^{\ominus}_{\varphi}$. If $(s,t)\in E^{\ominus}_{\partial\varphi'}$ in spite of $(s,t)\notin E^{\ominus}_{\partial\varphi}$, there exist (w_b,v_b) and (w_d,v_d) in $Q\cap E^{\ominus}_{\partial\varphi}$ such that $(w_b,t)\in\overline{E}^{\ominus}_{\partial\varphi}$ and $(s,v_d)\in\overline{E}^{\ominus}_{\partial\varphi}$ with $w_b=t$, $v_d=s$, or $v_d\leq v_b$.*

Proof. According to Lemmas 3.7 and 3.8, there is a numbering (w_1,v_1), (w_2,v_2), $\ldots$, (w_q,v_q) of the arcs in $Q\cap E^{\ominus}_{\partial\varphi}$ such that $y_k=\partial\varphi+\alpha\sum_{i=1}^{k}(\chi_{v_i}-\chi_{w_i})\in\mathrm{B}(f)$ for $k=1,2,\cdots,q$. We will show by induction on k that if $(s,t)\in E^{\ominus}_{y_k}$ in spite of $(s,t)\notin E^{\ominus}_{\partial\varphi}$ then there exist b and d such that $(w_b,t)\in\overline{E}^{\ominus}_{\partial\varphi}$ and $(s,v_d)\in\overline{E}^{\ominus}_{\partial\varphi}$ with $w_b=t$, $v_d=s$, or $v_d\leq v_b$.

Suppose $(s,t)\in E^{\ominus}_{y_k}$ in spite of $(s,t)\notin E^{\ominus}_{\partial\varphi}$. If $(s,t)\in E^{\ominus}_{x}$ for $x=y_{k-1}$, the statement holds by the inductive assumption. Hence we consider the case of $(s,t)\notin E^{\ominus}_{x}$. Then by Lemma 3.5 we have $(s,v_k)\in\overline{E}^{\ominus}_{x}$ and $(w_k,t)\in\overline{E}^{\ominus}_{x}$.

We now suppose that $s\neq v_i$ for any i and claim that there exists an arc $(w_d,v_d)\in Q\cap E^{\ominus}_{\partial\varphi}$ such that $(s,v_d)\in E^{\ominus}_{\partial\varphi}$ and $v_d\leq v_k$. If $(s,v_k)\in E^{\ominus}_{\partial\varphi}$, the claim is obvious. If $(s,v_k)\notin E^{\ominus}_{\partial\varphi}$, there exist d and j such that $(s,v_d)\in E^{\ominus}_{\partial\varphi}$ and $(w_j,v_k)\in E^{\ominus}_{\partial\varphi}$ with $v_d\leq v_j$ by the inductive assumption. From the eligibility of Q, we have $v_j\leq v_k$, and hence $v_d\leq v_k$.

Similarly, if $t\neq w_i$ for any i, there exists an arc $(w_b,v_b)\in Q\cap E^{\ominus}_{\partial\varphi}$ such that $(w_b,t)\in E^{\ominus}_{\partial\varphi}$ and $v_k\leq v_b$. Therefore, we have $(w_b,t)\in\overline{E}^{\ominus}_{\partial\varphi}$ and $(s,v_d)\in\overline{E}^{\ominus}_{\partial\varphi}$ with $w_b=t$, $v_d=s$, or $v_d\leq v_k\leq v_b$. ∎

Theorem 3.10. *Suppose φ' is obtained from φ by canceling an eligible cycle Q in $G^{\ominus}_{\varphi}$. Then $\rho_{\varphi'}(w)\geq\rho_{\varphi}(w)$ for every node $w\in V$. Moreover, there exists a residual arc $a\in F^{\ominus}_{\varphi}\cup B^{\ominus}_{\varphi}$ that disappears in $G_{\varphi'}$ or a node $u\in V$ such that $\rho_{\varphi'}(u)>\rho_{\varphi}(u)$.*

Proof. Suppose $\rho_{\varphi'}(s_0)=\eta(a_0)$ with $\partial^+a_0=s_0$ and that a_0 belongs to a cycle Q' in $G^{\ominus}_{\varphi'}$. We now prove that $\eta(a_0)\geq\rho_{\varphi}(s_0)$. If all the arcs in Q' are

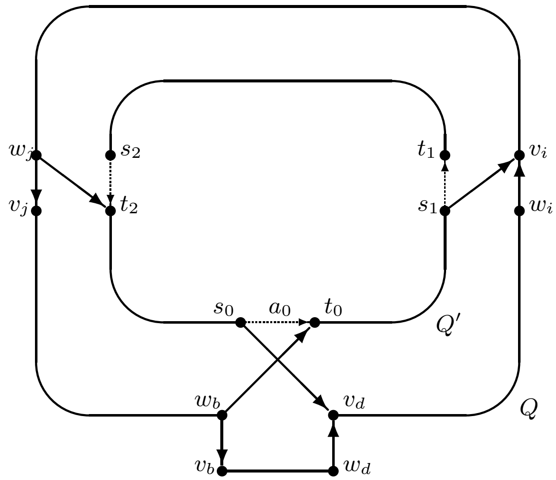

Fig. 1. *An eligible cycle Q in $G_\varphi^\ominus$ and a cycle Q' in $G_{\varphi'}^\ominus$ for the proof of Theorem 3.10. Solid arrows represent jumping arcs w.r.t. φ, and the three dotted arcs represent jumping arcs that appeared when we changed to φ', but which did not exist w.r.t. φ.*

in $A_\varphi^\ominus$, then this inequality is clear from the eligibility of Q'. Hence we may assume that $a \notin A_\varphi^\ominus$ for some $a \in Q'$. Note that $a \notin A_\varphi^\ominus$ and $a \in A_{\varphi'}^\ominus$ imply $a \in E_{\partial\varphi'}^\ominus$ by Lemma 3.6.

If $a_0 \notin A_\varphi^\ominus$ but the other arcs in Q' are in $A_\varphi^\ominus$, then by Lemma 3.9 there exist arcs (w_b, v_b) and (w_d, v_d) in $Q \cap E_{\partial\varphi}^\ominus$ such that $(w_b, t_0) \in E_{\partial\varphi}^\ominus$ and $(s_0, v_d) \in E_{\partial\varphi}^\ominus$ with $v_d \leq v_b$. Connecting $Q[v_d, w_b]$ and $Q'[t_0, s_0]$ by (w_b, t_0) and (s_0, v_d) yields a closed path, which includes a cycle in $G_\varphi^\ominus$ containing (w_b, t_0). Since the first arc in $Q'[t_0, s_0]$ is a residual arc, $t_0 = w_b$ contradicts to $\rho_\varphi(w_b) = v_b$. Thus we have $t_0 \neq w_b$, and it follows from the eligibility of Q that $v_b \leq t_0$. Since the first arc of $Q[v_d, w_b]$ is also a residual arc, $s = v_d$ implies $\rho_\varphi(s_0) = 0 \leq \eta(a_0)$. If $s \neq v_d$, we have $\rho_\varphi(s_0) \leq v_d \leq v_b \leq t_0 = \eta(a_0)$.

We now assume that some arc in $Q'[t_0, s_0]$ is not in $A_\varphi^\ominus$. Traverse the cycle Q' from $t_0 = \partial^- a_0$ and let $(s_1, t_1) \in Q'$ be the first arc that is not in $A_\varphi^\ominus$. Then by Lemma 3.9 there exists an arc $(w_i, v_i) \in Q$ such that $(s_1, v_i) \in A_\varphi^\ominus$. Similarly, traverse the cycle Q' backward from s_0 and let $(s_2, t_2) \in Q'$ be the first arc that is not in $A_\varphi^\ominus$. There exists an arc $(w_j, v_j) \in Q$ such that $(w_j, t_2) \in A_\varphi^\ominus$ by Lemma 3.9.

If $a_0 \in A_\varphi^\ominus$, connecting $Q'[t_2, s_1]$ and $Q[v_i, w_j]$ by (s_1, v_i) and (w_j, t_2) yields a closed path, which includes a cycle in $G_\varphi^\ominus$ containing a_0. Then the eligibility of Q implies $\rho_\varphi(s_0) \leq \eta(a_0)$.

If $a_0 \notin A_\varphi^\ominus$, it follows from Lemma 3.9 that there exist arcs (w_b, v_b) and (w_d, v_d) in $Q \cap E_{\partial\varphi}^\ominus$ such that $(w_b, t_0) \in E_{\partial\varphi}^\ominus$ and $(s_0, v_d) \in E_{\partial\varphi}^\ominus$ with $s_0 = v_d$, $t_0 = w_b$, or $v_d \le v_b$, see Figure 1. Then connecting $Q[v_i, w_b]$ and $Q'[t_0, s_1]$ by (w_b, t_0) and (s_1, v_i) yields a closed path, which includes a cycle in $G_\varphi^\ominus$ containing (w_b, t_0). Since the first arc in $Q'[t_0, s_1]$ is a residual arc, we have $t_0 \neq w_b$. Hence we have $v_b \le t_0$ from the eligibility of Q. On the other hand, connecting $Q[v_d, w_j]$ and $Q'[t_2, s_0]$ by (s_0, v_d) and (w_j, t_2) similarly brings a cycle in $G_\varphi^\ominus$ containing (s_0, v_d). Since the first arc of $Q[v_d, w_j]$ is also a residual arc, $s = v_d$ implies $\rho_\varphi(s_0) = 0 \le \eta(a_0)$. If $s \neq v_d$, we have $\rho_\varphi(s_0) \le v_d$, which together with $v_d \le v_b$ and $v_b \le t_0$ implies $\rho_\varphi(s_0) \le t_0$.

When we cancel Q, there is an arc $a \in Q$ that disappears in $G_{\varphi'}^\ominus$. If a is a jumping arc, $w = \partial^+ a$ satisfies $\rho_{\varphi'}(w) > \rho_\varphi(w)$. ∎

We are now ready to present an outline of our cycle canceling algorithms. The subroutine Cycle Cancel repeatedly finds and cancels eligible cycles until $G_\varphi^\ominus$ becomes acyclic. Theorem 3.10 says that we need at most $\mathrm{O}(n^2 + m) = \mathrm{O}(n^2)$ cancellations. Cycle Cancel does not construct $G_\varphi^\ominus$ explicitly, but instead computes exchange capacities using a depth-first search to find an eligible cycle. Once a node $v \in V$ turns out not to have a cycle through v, i.e., $\rho_\varphi(v) = n+1$, Theorem 3.10 says that we can ignore v in the subsequent depth-first searches until $G_\varphi^\ominus$ becomes acyclic. Cycle Cancel finds such a node or an eligible cycle after computing exchange capacities $\mathrm{O}(n)$ times. Thus Cycle Cancel takes $\mathrm{O}(n^3 h)$ time. Cycle Cancel is repeatedly called in the procedure Refine which obtains an ϵ-optimal submodular flow starting from a 2ϵ-optimal submodular flow. The following subsections describe the procedure Refine in the TA and the NA cases.

3.4.1. Refine for Tight Arc Cycle Canceling

This subsection describes the procedure Refine that transforms a TA 2ϵ-optimal submodular flow into a TA ϵ-optimal submodular flow. Given an initial feasible flow φ, we solve $\mathrm{AP}(\varphi, \epsilon)$ to obtain an optimal dual solution p whose components are integer multiples of ϵ. If the optimal value is zero, then φ is a TA ϵ-optimal submodular flow. Otherwise, we call Cycle Cancel.

When G_φ° becomes acyclic, there is no perfect matching that consists of tight arcs in $\widehat{G}_\varphi$, which means p is no longer optimal to $\mathrm{DAP}(\varphi, \epsilon)$. Since the weight of every edge in G_φ is an integer multiple of ϵ, the optimal value of $\mathrm{AP}(\varphi, \epsilon)$ has increased by at least ϵ. We repeat this process until $\mathrm{AP}(\varphi, \epsilon)$ has an optimal value zero, and then φ is a TA ϵ-optimal submodular flow.

Refine **for Tight Arc Cycle Canceling**

Step 1: Solve $\mathrm{AP}(\varphi,\epsilon)$ to obtain an optimal solution p for $\mathrm{DAP}(\varphi,\epsilon)$.

Step 2: While $\mathrm{AP}(\varphi,\epsilon)$ has a negative-cost assignment, do the following:

- **(2–1)** Call Cycle Cancel.
- **(2–2)** Update p into an optimal solution for $\mathrm{DAP}(\varphi,\epsilon)$.

Lemma 3.11. *The initial optimal value of* $\mathrm{AP}(\varphi,\epsilon)$ *is at least* $-2n\epsilon$*.*

Proof. Since $c^{\epsilon}(a) \geq c^{2\epsilon}(a) - 2\epsilon$, the initial reduced cost of any edge of $\mathrm{AP}(\varphi,\epsilon)$ w.r.t. an optimal dual solution p^* to the final assignment problem of the 2ϵ-scaling phase is at least -2ϵ. Since the objective value of p^* is zero, the cost of any assignment is the same if we replace cost by reduced cost. Thus the cost of any assignment is at least $-2n\epsilon$. ∎

Each time G°_{φ} becomes acyclic, the optimal value of $\mathrm{AP}(\varphi,\epsilon)$ increases by at least ϵ, so G°_{φ} becomes acyclic $\mathrm{O}(n)$ times per scaling phase. We must re-solve $\mathrm{AP}(\varphi,\epsilon)$ to get a new p, but this can be done via a Dijkstra shortest path update, which also happens $\mathrm{O}(n)$ times per phase, so this is not a bottleneck. It requires one call to Cycle Cancel to make G°_{φ} acyclic, requiring $\mathrm{O}(n^3h)$ time. Thus this Refine takes $\mathrm{O}(n^4h)$ time.

3.4.2. Refine for Negative Arc Cycle Canceling

This subsection describes the procedure Refine that transforms an NA 2ϵ-optimal submodular flow φ and a potential p that proves 2ϵ-optimality of φ into an NA ϵ-optimal submodular flow and a potential that proves NA ϵ-optimality of φ. Given such a pair of φ and p, we call an arc $a \in A_{\varphi}$ *improvable* if $c_p(a) < -\epsilon$. A node $v \in V$ is called *improvable* if v is the head of an improvable arc.

Given such an initial NA 2ϵ-optimal submodular flow φ and a potential p, we construct G^{-}_{φ} and call Cycle Cancel. These cycle cancellations do not produce any new improvable arcs. Then select an improvable node v, find the set W of nodes reachable from v in G^{-}_{φ}, and put $p(w) := p(w) - \epsilon$ for $w \in W$. This Cut-Relabel(v) operation does not violate the NA feasibility of p nor produce a new improvable arc. Furthermore, it eliminates all the improvable arcs entering v. That is, v is no longer an improvable node. We repeat this process until G^{-}_{φ} has no more improvable nodes. If G^{-}_{φ} has no improvable nodes, then it is obvious that φ is an NA ϵ-optimal submodular flow.

Refine **for Negative Arc Cycle Canceling**

Step 1: Call Cycle Cancel.

Step 2: While there still exist improvable nodes, do the following:

(2–1) Select an improvable node v and call Cut-Relabel(v).

(2–2) Call Cycle Cancel.

Since one iteration eliminates one improvable node, it is clear that we need at most n iterations. Thus this Refine also takes $\mathrm{O}(n^4 h)$ time.

3.5. Successive Approximation

We now apply the successive approximation technique of Goldberg and Tarjan [16] to our cycle canceling algorithms. The algorithms start with $\epsilon = 2^{\lceil \log_2 C \rceil}$ and scale the parameter ϵ. Each scaling phase uses Refine to find an ϵ-optimal submodular flow. When the algorithm terminates, after performing $\mathrm{O}(\log(nC))$ scaling phases, it obtains a submodular flow φ that is of minimum cost due to Lemma 3.2.

Algorithm Successive Approximation

Step 0: Find a feasible submodular flow φ.

Step 1: Compute $\epsilon := 2^{\lceil \log_2 C \rceil}$.

Step 2: While $\epsilon \geq 1/n$, do the following.

(2–1) Put $\epsilon := \epsilon/2$.

(2–2) Find an ϵ-optimal submodular flow φ^* by Refine starting from φ.

(2–3) Put $\varphi := \varphi^*$.

We now compute the running time of Successive Approximation. By Lemma 3.2 there are $\mathrm{O}(\log(nC))$ scaling phases. We have already remarked that both versions of Refine take $\mathrm{O}(n^4 h)$ time. Thus we have the following theorem.

Theorem 3.12. *Both Tight Arc Cycle Canceling and Negative Arc Cycle Canceling run in* $\mathrm{O}(n^4 h \log(nC))$ *time.* ∎

4. Strongly Polynomial Bounds

It is generally considered that a strongly polynomial algorithm cannot use the floor or ceiling operation, unless the magnitude of the number is polynomially bounded. This is because computing the floor of a ratio of real numbers can generally be accomplished only through something like binary search. This is bad since the running time of binary search-type algorithms definitely depends on the size of the data. Similarly, we can't compute $\lceil \log_2 C \rceil$ in strongly polynomial time since this would involve counting the $\Theta(\log C)$ bits of C. Step 1 of Successive Approximation computes $\epsilon := 2^{\lceil \log_2 C \rceil}$; since we are using real arithmetic, we can just replace this by $\epsilon := C$.

Our Tight Arc Cycle Canceling uses floors in the definition of c^ϵ, so it is tricky to extend it to be strongly polynomial. The min cost flow versions of the Tight Arc Cycle Canceling [33] include variants which do not need to round costs. Unfortunately, we have not been able to extend these variants to submodular flow. The main hurdle seems to be that the argument showing that p remains TA feasible in Lemma 3.5 does not have any slack in it, which appears to be necessary to make a more relaxed variant of the algorithm work for this case.

Happily, a simple observation allows us to get a strongly polynomial version of the Tight Arc Cycle Canceling despite using floors. The proof of Lemma 3.11 noted that we can compute using reduced costs in $\mathrm{AP}(\varphi, \epsilon)$ w.r.t. p^* (the optimal node prices in DAP at the end of the previous phase). For edge $a = (v, w)$ this AP reduced cost is $c^\epsilon(a) - p^*(v^+) - p^*(w^-)$. Note that if $c(a) - p^*(v^+) - p^*(w^-) \geq 2n\epsilon$, then $c^\epsilon(a) - p^*(v^+) - p^*(w^-) \geq 2n\epsilon$, implying that edge a cannot appear in any optimal solution of $\mathrm{AP}(\varphi, \epsilon)$. We can identify and delete such edges without harming strong polynomiality in $\mathrm{O}(m)$ time. For the remaining edges, we can compute $c^\epsilon(a) - p^*(v^+) - p^*(w^-)$ using binary search in $\mathrm{O}(\log n)$ time each. This needs to be done only once in each scaling phase, so it is not a bottleneck.

Our Negative Arc Cycle Canceling does not use rounding, so it does not need this change.

Now we can apply the standard strongly polynomial machinery. We quote the proximity lemma we need, which applies to both notions of ϵ-optimality.

Lemma 4.1 **([12, Theorem 3]).** *Let φ^* be any optimal submodular flow, and suppose that p proves that φ is ϵ-optimal. If $c_p(a) > n\epsilon$, then $\varphi^*(a) = l(a)$. If $c_p(a) < -n\epsilon$, then $\varphi^*(a) = u(a)$. If $p(w) > p(v) + n\epsilon$, then $\widetilde{r}(\partial\varphi^*, v, w) = 0$.* ∎

Note that G_φ has $m' = m + n^2 = \mathrm{O}(n^2)$ arcs. A *min mean cycle* is a directed cycle in N_φ that minimizes $c(Q)/|Q|$ over all directed cycles in N_φ. A min mean cycle with a minimum number of arcs can be computed in $\mathrm{O}(m'n) = \mathrm{O}(n^3)$ time [26]. Cui and Fujishige [2, Theorem 3.2] show that such a min mean cycle is admissible, so by Lemma 3.7 we can feasibly cancel it. If Q^* is a min mean cycle with value $\mu^* = c(Q^*)/|Q^*|$, then any of the min mean cycle algorithms produces dual variables p proving that φ is μ^*-optimal. We now divide the scaling phases into *groups*, where each group consists of $\beta = \lceil \log_2 n+1 \rceil$ consecutive scaling phases. We then replace the first cycle cancel in each group by canceling an admissible min mean cycle Q^*.

Our choice of β ensures that the value of ϵ at the end of a group is smaller than $1/n$ of the min mean cycle value μ^*, which is the value of ϵ at the beginning of the group. From this it follows that at least one arc a of Q^* must satisfy one of the hypotheses of Lemma 4.1, so that we can fix $\varphi(a)$ to one of its bounds. That is, if $a \in F_\varphi$ we fix $\varphi(a) = u(a)$, if $a \in B_\varphi$ we fix $\varphi(a) = l(a)$, and if $a \in E_{\partial\varphi}$ we "fix" the flow in this jumping arc to zero by forbidding it to be used hereafter. It can be shown as in [33] that the flow on a was changed by canceling Q^*, so that $\varphi(a)$ is newly fixed to a bound. After $\mathrm{O}(m')$ groups all $\varphi(a)$'s are fixed, so we must be optimal. This yields

Theorem 4.2. *For both Tight Arc Cycle Canceling and Negative Arc Cycle Canceling, performing scaling phases by groups with a min mean cycle cancellation at the beginning of each group takes* $\mathrm{O}(n^6 h \log n)$ *time overall.*

Proof. Each group takes $\mathrm{O}(\log n)$ scaling phases, each of which costs $\mathrm{O}(n^4 h)$ time, and we need $\mathrm{O}(m')$ groups to reach optimality. The cost of the min mean cycle computations is ($\mathrm{O}(m')$ computations) $\times$ ($\mathrm{O}(n^3)$ time/computation), or $\mathrm{O}(n^5)$ time, which is not a bottleneck. ∎

5. Separable Convex Objective Functions

Rockafellar's book *Network Flows and Monotropic Optimization* [30] argues that it is not really harder to optimize with a separable convex objective function than it is to optimize with a linear objective function. Papers such as Hochbaum and Shanthikumar [19], Karzanov and McCormick [27], and McCormick, Shigeno, and Iwata [25] have shown that this is indeed the case for minimum cost flow algorithms based on capacity scaling, min mean cycle canceling, and relaxed most negative cycle canceling, respectively (in all three cases, even over totally unimodular constraints). Iwata [20] shows that capacity scaling further extends to separable convex cost submodular flow, yielding the only previous polynomial algorithm for separable convex submodular flow we know of. The concurrent paper [23] shows a cut canceling algorithm for submodular flow also extends to separable convex costs. This section sketches how to extend the cycle canceling algorithms of this paper to separable convex cost submodular flow.

In the separable convex case each arc a has an objective function $g_a(\varphi(a))$ which is convex, and we want to minimize $\sum_a g_a(\varphi(a))$. Note that each g_a is allowed to take on the value $+\infty$, so that it is possible to incorporate any lower or upper bounds on $\varphi(a)$ into g_a by setting $g_a(\varphi(a)) = +\infty$ for $\varphi(a) < l(a)$ or $\varphi(a) > u(a)$. The convexity of g_a ensures the existence of the *right derivative*, denoted by $g_a^{\vdash}(\varphi(a))$, and the *left derivative*, $g_a^{\dashv}(\varphi(a))$, at every $\varphi \in \mathbf{R}$. Note that $g_a^{\dashv}(\varphi(a)) < g_a^{\vdash}(\varphi(a))$ is possible at a *breakpoint*. We assume that we have a routine that, given any $\alpha \in \mathbf{R}$ and $a \in A$, can compute $l_a(\alpha) = \inf\{\varphi \mid g_a^{\dashv}(\varphi) \geq \alpha\}$ and $u_a(\alpha) = \sup\{\varphi \mid g_a^{\vdash}(\varphi) \leq \alpha\}$ in O(1) time. Note that if g_a is differentiable with derivative g'_a, then $l_a(\alpha) = u_a(\alpha) = (g'_a)^{-1}(\alpha)$, i.e., the inverse of the derivative.

Given a potential $p \in \mathbf{R}^V$, define the *tension*, or *potential difference* for each arc $a = (v, w)$ w.r.t. p as $\tau_p(a) = p(w) - p(v)$. A characterization of optimality here is

Theorem 5.1 ([30]). *A submodular flow φ is optimal if and only there exists a potential $p \in \mathbf{R}^V$ such that $g_a^{\dashv}(\varphi(a)) \leq \tau_p(a) \leq g_a^{\vdash}(\varphi(a))$ for all $a \in A$.* ∎

The main effect from this more general objective function is that it changes residual costs c_φ (since costs now do depend on φ, we start subscripting c by φ) and residual capacities r_φ. The forward residual cost of arc a is $c_\varphi^{\vdash}(a) = g_a^{\vdash}(\varphi(a))$ and the backward residual cost of arc a is $c_\varphi^{\dashv}(a) = -g_a^{\dashv}(\varphi(a))$. To define residual capacities we will need to carry along explicit node potentials p in our algorithms. Both algorithms will keep $p(v)$ as an integer multiple of ϵ for all $v \in V$ in the ϵ-scaling phase, so that τ_p will also be an integer multiple of ϵ. The Tight Arc Cycle Canceling will maintain the invariant that $\tau_p(a) \leq g_a^{\vdash}(\varphi(a))$ and $g_a^{\dashv}(\varphi(a)) \leq \tau_p(a) + \epsilon$, and the Negative Arc Cycle Canceling will maintain the invariant that $g_a^{\dashv}(\varphi(a)) - \epsilon \leq \tau_p(a) \leq g_a^{\vdash}(\varphi(a)) + \epsilon$.

For the Tight Arc Cycle Canceling, the forward residual capacity of $a \in A$ is $u_a(\tau_p + \epsilon) - \varphi(a)$ and the backward residual capacity of $\bar{a}$ is $\varphi(a) - l_a(\tau_p)$, see Figure 2. For the Negative Arc Cycle Canceling, the forward residual capacity of $a \in A$ is $u_a(\tau_p + \epsilon) - \varphi(a)$ and the backward residual capacity of $\bar{a}$ is $\varphi(a) - l_a(\tau_p - \epsilon)$, see Figure 3. From these we can define the residual graph G_φ as before.

Now we run the algorithms largely unchanged. As φ is changed by Cycle Cancel the residual costs will change, but it can be checked that the invariants will be preserved. It can also be checked that the invariants are preserved when p changes. The analogue of Lemma 3.11 for the Tight Arc

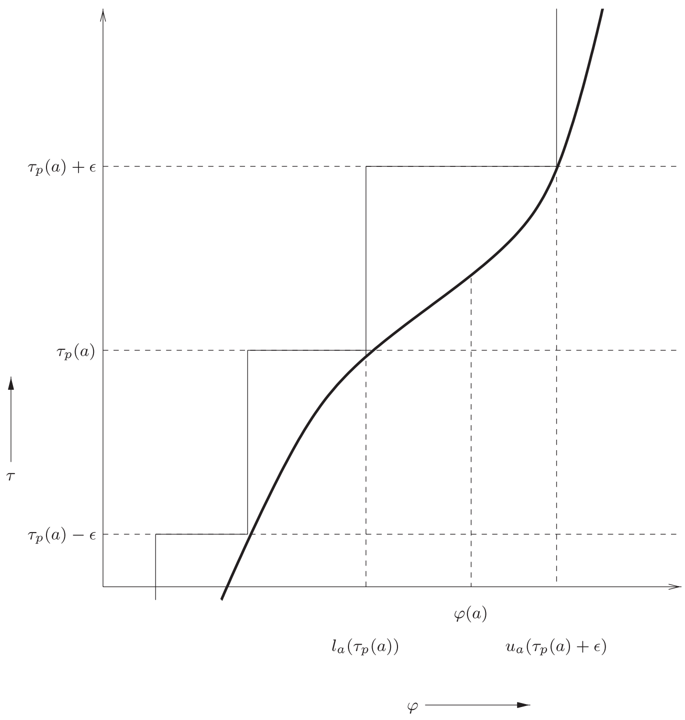

Fig. 2. *The heavy line is the graph of the derivative of $g_a(\varphi(a))$, a generalized kilter line. The thin solid line is the rounded TA ϵ-approximation of the kilter line. The horizontal distances between $l_a(\tau_p(a))$ and $\varphi(a)$, and between $\varphi(a)$ and $u_a(\tau_p(a)+\epsilon)$, are the residual capacities.*

Cycle Canceling is still true in this more general case. Thus in both cases Refine still runs in $O(n^4h)$ time.

With general separable convex objective functions it is impossible to get any a priori bound on how large we must choose the initial value of ϵ in Successive Approximation to ensure that our initial solution is ϵ-optimal. Thus we use B to stand for this unknown initial value. Similarly, there is no way to ensure exact optimality no matter how many scaling phases we run, so we use ι to stand for the final value of ϵ we use. Thus Successive Approximation makes $O(B/\iota)$ calls to Refine, and we get the following

Theorem 5.2. *For both Tight Arc Cycle Canceling and Negative Arc Cycle Canceling, the running time in the separable convex case is* $O(n^4h\log(B/\iota))$. ∎

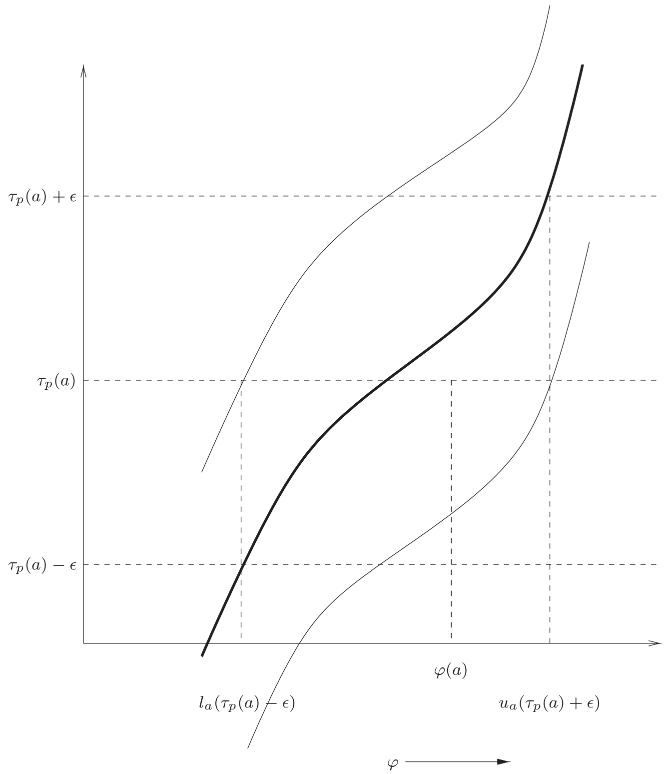

Fig. 3. *The heavy line is the graph of the derivative of $g_a(\varphi(a))$, a generalized kilter line. The two thin solid lines are the bounds on the NA ϵ-approximation of the kilter line. The horizontal distances between $l_a(\tau_p(a)-\epsilon)$ and $\varphi(a)$, and between $\varphi(a)$ and $u_a(\tau_p(a)+\epsilon)$, are the residual capacities.*

The interested reader can consult, e.g., [27] for more details on what guarantees are possible for ϵ-optimality implying that φ is close to some optimal φ^*, to see how the algorithms can compute an exact solution in polynomial time when all the g_a are piecewise quadratic, and to see how the algorithms are strongly polynomial when all the g_a are piecewise linear.

Acknowledgment

We would like to thank an anonymous referee of [33] for suggesting that Goldberg's algorithm might extend to submodular flow.

References

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin: *Network Flows — Theory, Algorithms, and Applications*, Prentice Hall (1993).

[2] W. Cui and S. Fujishige: A primal algorithm for the submodular flow problem with minimum-mean cycle selection, *J. Oper. Res. Soc. Japan*, **31** (1988), 431–440.

[3] W. H. Cunningham and A. Frank: A primal-dual algorithm for submodular flows, *Math. Oper. Res.*, **10** (1985), 251–262.

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

[5] J. Edmonds and R. M. Karp: Theoretical improvements in algorithmic efficiency for network flow problems, *J. ACM*, **19** (1972), 248–264.

[6] L. Fleischer, S. Iwata, and S. T. McCormick: A faster capacity-scaling algorithm for minimum cost submodular flow, *Math. Programming*, **92** (2002), 119–139.

[7] A. Frank: Finding feasible vectors of Edmonds–Giles polyhedra, *J. Combinatorial Theory*, Ser. B, **36** (1984), 221–239.

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

[9] A. Frank and É. Tardos: Generalized polymatroids and submodular flows, *Math. Programming*, **42** (1988), 489–563.

[10] S. Fujishige: Algorithms for solving the independent-flow problems, *J. Oper. Res. Soc. Japan*, **21** (1978), 189–204.

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

[12] S. Fujishige, H. Röck, and U. Zimmermann: A strongly polynomial algorithm for minimum cost submodular flow problems, *Math. Oper. Res.*, **14** (1989), 60–69.

[13] S. Fujishige and X. Zhang: New algorithms for the intersection problem of submodular systems, *Japan J. Indust. Appl. Math.*, **9** (1992), 369–382.

[14] A. V. Goldberg: Scaling algorithms for the shortest paths problem, *SIAM J. Comput.*, **24** (1995), 494–504.

[15] A. V. Goldberg and R. E. Tarjan: Finding minimum-cost circulations by canceling negative cycles, *J. ACM*, **36** (1989), 873–886.

[16] A. V. Goldberg and R. E. Tarjan: Finding minimum-cost circulations by successive approximation, *Math. Oper. Res.*, **15** (1990), 430–466.

[17] M. Grötschel, L. Lovász, and A. Schrijver: *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, 1988.

[18] R. Hassin: Minimum cost flow with set-constraints, *Networks*, **12** (1982), 1–21.

[19] D. S. Hochbaum and J. G. Shanthikumar: Convex separable optimization is not much harder than linear optimization, *J. ACM*, **37** (1990), 843–862.

[20] S. Iwata: A capacity scaling algorithm for convex cost submodular flows, *Math. Programming*, **76** (1997), 299–308.

[21] S. Iwata, L. Fleischer, and S. Fujishige: A combinatorial strongly polynomial-time algorithm for minimizing submodular functions, *J. ACM*, **48** (2001), 761–777.

[22] S. Iwata, S. T. McCormick, and M. Shigeno: A faster algorithm for minimum cost submodular flows, *Proceedings of the Ninth Annual ACM/SIAM Symposium on Discrete Algorithms* (1998), 167–174.

[23] S. Iwata, S. T. McCormick, and M. Shigeno: A strongly polynomial cut canceling algorithm for the submodular flow problem, *Proceedings of the Seventh MPS Conference on Integer Programming and Combinatorial Optimization* (1999), G. Cornuéjols, R. E. Burkard, and G. J. Woeginger, eds., 259–272.

[24] S. Iwata, S. T. McCormick, and M. Shigeno: A fast cost scaling algorithm for submodular flow, *Inform. Process. Lett.*, **74** (2000), 123–128.
[25] S. Iwata, S. T. McCormick, and M. Shigeno: A relaxed cycle-canceling approach to separable convex optimization in unimodular linear spaces, in preparation.
[26] R. M. Karp: A characterization of the minimum cycle mean in a digraph, *Discrete Math.*, **23** (1978), 309–311.
[27] A. V. Karzanov and S. T. McCormick: Polynomial methods for separable convex optimization in totally unimodular linear spaces with applications to circulations and co-circulations in networks, *SIAM J. Comput.*, **26** (1997), 1245–1275.
[28] E. L. Lawler and C. U. Martel: Computing maximal polymatroidal network flows, *Math. Oper. Res.*, **7** (1982), 334–347.
[29] T. Naitoh and S. Fujishige: A note on the Frank–Tardos bitruncation algorithm for crossing-submodular functions, *Math. Programming*, **53** (1992), 361–363.
[30] R. T. Rockafellar: *Network Flows and Monotropic Optimization.* Wiley Interscience, New York, NY (1984).
[31] P. Schönsleben: *Ganzzahlige Polymatroid-Intersektions-Algorithmen*, ETH Zürich (1980).
[32] A. Schrijver: A combinatorial algorithm minimizing submodular functions in strongly polynomial time, *J. Combinatorial Theory*, Ser. B, **80** (2000), 346–355.
[33] M. Shigeno, S. Iwata, and S. T. McCormick: Relaxed most negative cycle and most positive cut canceling algorithms for minimum cost flow, *Math. Oper. Res.*, **25** (2000), 76–104.
[34] É. Tardos: A strongly polynomial minimum cost circulation algorithm, *Combinatorica*, **5** (1985), 247–256.
[35] É. Tardos, C. A. Tovey, and M. A. Trick: Layered augmenting path algorithms, *Math. Oper. Res.*, **11** (1986), 362–370.
[36] C. Wallacher and U. Zimmermann: A polynomial cycle canceling algorithm for submodular flows, *Math. Programming*, **86** (1999), 1–15.
[37] U. Zimmermann: Negative circuits for flows and submodular flows, *Discrete Appl. Math.*, **36** (1992), 179–189.

Satoru Iwata

Department of Mathematical Engineering and Information Physics, University of Tokyo, Tokyo 113-8656, Japan

iwata@sr3.t.u-tokyo.ac.jp

S. Thomas McCormick

Faculty of Commerce and Business Administration, University of British Columbia, Vancouver, BC V6T 1Z2 Canada

stmv@adk.commerce.ubc.ca

Maiko Shigeno

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

maiko@shako.sk.tsukuba.ac.jp