Capacity Scaling Algorithm for Scalable M-convex Submodular Flow Problems

Satoko MORIGUCHI [*] Kazuo MUROTA [†]

December 18, 2002

Abstract. An M-convex function is a nonlinear discrete function defined on integer points introduced by Murota in 1996, and the M-convex submodular flow problem is one of the most general frameworks of efficiently solvable combinatorial optimization problems. It includes the minimum cost flow and the submodular flow problems as its special cases. In this paper, we first devise a successive shortest path algorithm for the M-convex submodular flow problem. We then propose an efficient algorithm based on a capacity scaling framework for the scalable M-convex submodular flow problem. Here an M-convex function $f(x)$ is said to be scalable if $f^{\alpha}(x) := f(\alpha x)$ is also M-convex for any positive integer α.

Key words. discrete optimization; discrete convex function; submodular flow; algorithm

[*] Graduate School of Information Sciences and Engineering, Tokyo Institute of Technology, Tokyo 152-8552, Japan. E-mail: Satoko.Moriguchi@is.titech.ac.jp

[†] Graduate School of Information Science and Technology, University of Tokyo, Tokyo 113-8656, Japan. E-mail: murota@mist.i.u-tokyo.ac.jp

1 Introduction

The M-convex submodular flow problem, introduced by Murota [16], is one of the most general frameworks of efficiently solvable combinatorial optimization problems. It includes the minimum cost flow and the submodular flow problems [3, 7] as its special cases, and has an application to mathematical economics [19]. The submodular flow problem with an M-convex function admits nice optimality criteria in terms of potentials and negative cycles like the minimum cost flow problem. The concept of M-convex functions was proposed by Murota [14, 15] as a natural extension of the concept of valuated matroids [2] and plays a central role in the theory of discrete convex analysis [17, 18].

A number of combinatorial algorithms to solve the submodular flow problem have been proposed as extensions of minimum cost flow algorithms. The first such polynomial-time algorithm is due to Cunningham and Frank [1]. It generalizes a cost-scaling primal-dual method.

It is quite natural to adopt a capacity scaling approach, i.e., to scale capacities as well as demands, which are given by a submodular function. A straightforward attempt to scale a submodular function, however, destroys the submodularity. Extending the Edmonds–Karp [4] algorithm for minimum-cost flow problem, Iwata [9] devised the first capacity scaling algorithm for the submodular flow problem. Introducing a variant of the Dijkstra shortest path algorithm modified to deal with exchange capacity arcs, Fleischer, Iwata and McCormick [5] have improved this algorithm.

In designing combinatorial algorithms for the M-convex submodular flow problem, it is quite natural to attempt extensions of the existing methods for the submodular flow problem. In fact, submodular flow algorithms such as the cycle-canceling and primal-dual methods are successfully extended to solve the M-convex submodular flow problem. Feasibility of the M-convex submodular flow problem can also be checked by the algorithm of Frank [6] for the submodular flow problem. In spite of the fact that an M-convex function is not closed under the scaling operation, Iwata and Shigeno [10] have devised a polynomial-time algorithm based on a new scaling framework, i.e., the conjugate scaling algorithm.

The objective of this paper is to propose an efficient algorithm based on a capacity scaling framework. Our capacity scaling algorithm uses the proximity theorem for M-convex functions. This is compared to that the conjugate scaling algorithm of Iwata–Shigeno [10] is based on the proximity theorem for L-convex functions.

In this paper, we first devise a successive shortest path algorithm for the M-convex submodular flow problem. This is closely related to the primal-dual algorithm of Murota [13] for the valuated matroid intersection problem. We then propose an efficient algorithm based on a capacity scaling framework for the scalable M-convex submodular flow problem. Here an M-convex function $f(x)$ is said to be scalable if $f^{\alpha}(x) := f(\alpha x)$ is also M-convex for any positive integer α. It is worth mentioning that a number of fundamental M-convex functions are scalable, including linear, separable, quadratic, and laminar M-convex functions [11, 17, 18]. We embed the successive shortest path algorithm in a scaling framework to propose capacity scaling algorithm.

2 M-convex Submodular Flow

2.1 M-convex Function

Let V be a finite set. A function $f : \mathbf{Z}^V \to \mathbf{R} \cup \{+\infty\}$ is said to be M-convex if it satisfies

(M-EXC) $\forall x, y \in \mathrm{dom}\, f$, $\forall u \in \mathrm{supp}^+(x-y)$, $\exists v \in \mathrm{supp}^-(x-y)$ such that

$$f(x) + f(y) \geq f(x - \chi_u + \chi_v) + f(y + \chi_u - \chi_v),$$

where $\chi_w \in \{0,1\}^V$ is the characteristic vector of $w \in V$ and

$$\mathrm{dom}\, f = \{x \in \mathbf{Z}^V \mid f(x) < +\infty\},$$
$$\mathrm{supp}^+(x-y) = \{w \in V \mid x(w) > y(w)\}, \;\; \mathrm{supp}^-(x-y) = \{w \in V \mid x(w) < y(w)\}.$$

It is easy to see that $B = \mathrm{dom}\, f$ satisfies the following property:

(B-EXC) $\forall x, y \in B$, $\forall u \in \mathrm{supp}^+(x-y)$, $\exists v \in \mathrm{supp}^-(x-y)$ such that

$$x - \chi_u + \chi_v \in B, \;\; y + \chi_u - \chi_v \in B.$$

Note that (B-EXC) implies $\sum_{v \in V} x(v) = \sum_{v \in V} y(v)$ for any $x, y \in B$. A nonempty set $B \subseteq \mathbf{Z}^V$ with (B-EXC) is called an M-convex set.

Modification of a function by a linear function is a fundamental operation. For an M-convex function f and a vector p, we denote by $f[-p]$ the function defined by

$$f[-p](x) = f(x) - \langle p, x \rangle \qquad (x \in \mathbf{Z}^V).$$

This is M-convex for M-convex f.

For $f : \mathbf{Z}^V \to \mathbf{R} \cup \{+\infty\}$ and a positive integer α, define a function $f^\alpha : \mathbf{Z}^V \to \mathbf{R} \cup \{+\infty\}$ by

$$f^\alpha(x) = f(\alpha x) \qquad (x \in \mathbf{Z}^V).$$

This operation is called *scaling* by a factor of α. Even if f is an M-convex function, f^α is not necessarily M-convex in general. An M-convex function f is called *scalable* if f^α is also M-convex for any positive integer α. We can identify a number of subclasses of scalable M-convex functions such as linear functions, separable convex functions, quadratic M-convex functions and laminar convex functions [11, 17, 18].

Let α be a positive integer, and $x_\alpha \in \mathrm{dom}\, f$. We call x_α an α-local minimum of f if it satisfies

$$f(x_\alpha) \leq f(x_\alpha - \alpha(\chi_u - \chi_v)) \quad (\forall u, v \in V).$$

The following is a "proximity theorem," showing that a global minimizer of an M-convex function exists in the neighborhood of an α-local minimum.

Theorem 2.1 ([11]). *Let $f : \mathbf{Z}^V \to \mathbf{R} \cup \{+\infty\}$ be an M-convex function and α be any positive integer. Suppose that $x_\alpha \in \mathrm{dom}\, f$ satisfies $f(x_\alpha) \leq f(x_\alpha - \alpha(\chi_u - \chi_v))$ for all $u, v \in V$. Then, there exists some $x_* \in \arg\min f$ such that*

$$|x_\alpha(v) - x_*(v)| \leq (n-1)(\alpha - 1) \qquad (v \in V).$$

A special case of Theorem 2.1 with $\alpha = 1$ yields the following local characterization of global minimality.

Theorem 2.2 ([14]). *Let $f : \mathbf{Z}^V \to \mathbf{R} \cup \{+\infty\}$ be an M-convex function and $x_* \in \mathrm{dom}\, f$. Then $x_* \in \arg\min f$ if and only if $f(x_*) \leq f(x_* - \chi_u + \chi_v)$ for all $u, v \in V$.*

2.2 Base Polyhedra and Exchange Capacity

Associated with a submodular function $\rho : 2^V \to \mathbf{Z}$ the base polyhedron $\mathbf{B}(\rho)$ is defined as follows:

$$\mathbf{B}(\rho) = \{x \mid x \in \mathbf{R}^V, x(V) = \rho(V), \forall X \subseteq V : x(X) \leq \rho(X)\}.$$

An M-convex set is the same as the set of integer points of the base polyhedron. Namely, a bounded set $B \subseteq \mathbf{Z}^V$ is M-convex if and only if $B = \mathbf{B}(\rho) \cap \mathbf{Z}^V$ for some $\rho : 2^V \to \mathbf{Z}$.

For any base $x \in B$ and $u, v \in V$ with $u \neq v$, put

$$\tilde{c}(x, v, u) = \max\{\alpha \mid \alpha \in \mathbf{Z}, x - \alpha(\chi_u - \chi_v) \in B\},$$

which is called the exchange capacity.

The exchangeability graph is a directed graph with the vertex set V and the arc set $C_x = \{(u,v) \mid \tilde{c}(x,v,u) > 0\}$. The exchangeability graph is transitive. Namely, $(u,v) \in C_x$ and $(v,w) \in C_x$ imply $(u,w) \in C_x$.

Lemma 2.3 ([8], No-short cut lemma). *Suppose that B is an M-convex set, $x \in B$ and that $u_1, v_1, u_2, v_2, \ldots, u_r, v_r$ are distinct. If $x + \chi_{v_i} - \chi_{u_i} \in B$ for $i = 1, \ldots, r$ and $x + \chi_{v_j} - \chi_{u_i} \notin B$ for any $i < j$, then $y = x + \sum_{i=1}^{r}(\chi_{v_i} - \chi_{u_i}) \in B$.*

Lemma 2.4. *Suppose that $x \in \arg\min f[-p]$ and that $u_1, v_1, u_2, v_2, \ldots, u_r, v_r$ are distinct. If $x + \chi_{v_i} - \chi_{u_i} \in \arg\min f[-p]$ for $i = 1, \ldots, r$ and $x + \chi_{v_j} - \chi_{u_i} \notin \arg\min f[-p]$ for any $i < j$, then $y = x + \sum_{i=1}^{r}(\chi_{v_i} - \chi_{u_i}) \in \arg\min f[-p]$.*

Proof. Since $\arg\min f[-p]$ is an M-convex set, the assertion follows from the no-short cut lemma 2.3 applied to $\arg\min f[-p]$. □

2.3 M-convex Submodular Flow Problem

Let $G = (V, A)$ be a directed graph with upper and lower capacity bounds $\underline{c}, \overline{c} \in \mathbf{Z}^A$ and the cost function $\gamma \in \mathbf{R}^A$. For each vertex $v \in V$, let $\delta^+ v$ (resp., $\delta^- v$) denote the set of arcs leaving (resp., entering) v. For each arc $a \in A$, $\partial^+ a$ designates the initial vertex of a, and $\partial^- a$ the terminal vertex of a. The boundary of flow ξ is defined to be

$$\partial\xi(v) = \sum\{\xi(a) \mid a \in \delta^+ v\} - \sum\{\xi(a) \mid a \in \delta^- v\} \qquad (v \in V),$$

which represents the net flow leaving vertex v. Suppose that $B \subseteq \mathbf{Z}^V$ is a bounded M-convex set. Then the integer-flow version of the submodular flow problem is formulated as follows [3].

Submodular flow problem MSFP_1 (linear arc cost, integer-flow)

$$\begin{aligned}
\text{minimize} \quad & \Gamma_1(\xi) = \sum_{a \in A} \gamma(a)\xi(a) && (2.1)\\
\text{subject to} \quad & \underline{c}(a) \leq \xi(a) \leq \overline{c}(a) \quad (a \in A), && (2.2)\\
& \partial\xi \in B, && (2.3)\\
& \xi(a) \in \mathbf{Z} \quad (a \in A). && (2.4)
\end{aligned}$$

The submodular flow problem is a well-behaved combinatorial problem that has nice properties such as optimality criterion in terms of potentials (dual variables), optimality criterion in terms of negative cycles, integrality of optimal solutions, and efficient algorithms.

A generalization of the submodular flow problem is obtained by introducing a cost function for the flow boundary $\partial\xi$ rather than merely imposing the feasibility constraint $\partial\xi \in B$. Namely, with a function $f : \mathbf{Z}^V \to \mathbf{R} \cup \{+\infty\}$ we add a new term $f(\partial\xi)$ to the objective function, thereby imposing feasibility constraint $\partial\xi \in B = \mathrm{dom}\, f$ implicitly. The aforementioned nice properties are maintained if f is an M-convex function. Such problem described by an M-convex function f is called the M-convex submodular flow problem [16].

M-convex submodular flow problem MSFP_2 (linear arc cost, integer-flow)

$$\begin{aligned}
\text{minimize} \quad & \Gamma_2(\xi) = \sum_{a \in A} \gamma(a)\xi(a) + f(\partial\xi) && (2.5)\\
\text{subject to} \quad & \underline{c}(a) \leq \xi(a) \leq \overline{c}(a) \quad (a \in A), && (2.6)\\
& \partial\xi \in \mathrm{dom}\, f, && (2.7)\\
& \xi(a) \in \mathbf{Z} \quad (a \in A). && (2.8)
\end{aligned}$$

Note that the M-convex submodular flow problem with a $\{0, +\infty\}$-valued f reduces to the submodular flow problem MSFP_1.

Throughout this paper we assume that $\mathrm{dom}\, f$ is bounded and

$$\mathrm{dom}\, f \subseteq \{x \in \mathbf{Z}^V \mid x(V) = 0\}$$

since $\partial\xi(V) = 0$ for any flow ξ and $\partial\xi \in \mathrm{dom}\, f$ is imposed.

The following optimality condition is known for the M-convex submodular flow problem MSFP$_2$. By a potential we mean a function $p : V \to \mathbf{R}$ (or a vector $p \in \mathbf{R}^V$) on the vertex set.

Theorem 2.5 ([16, 17, 18], Optimality condition by the potential). *A flow $\xi : A \to \mathbf{Z}$ satisfying (2.6) and (2.7) is optimal if and only if there exists a potential $p : V \to \mathbf{R}$ such that*

(i)

$$\gamma_p(a) > 0 \implies \xi(a) = \underline{c}(a), \tag{2.9}$$

$$\gamma_p(a) < 0 \implies \xi(a) = \overline{c}(a) \tag{2.10}$$

in terms of the reduced cost $\gamma_p : A \to \mathbf{R}$ defined by $\gamma_p(a) = \gamma(a) + p(\partial^+ a) - p(\partial^- a) \quad (a \in A)$, and

(ii) $\partial\xi \in \arg\min f[-p]$.

3 A Successive Shortest Path Algorithm

This section describes a successive shortest path (SSP) algorithm for the M-convex submodular flow problem, which is the basis for our algorithm. As with the successive shortest path algorithm for minimum cost flows, this algorithm is not polynomial, but only pseudo-polynomial since the number of iterations is linear in $C = \max(C_1, C_2)$, where

$$\begin{aligned} C_1 &= \max\{\max_{a \in A} |\underline{c}(a)|,\ \max_{a \in A} |\overline{c}(a)|\}, \\ C_2 &= \max_{x,y \in \mathrm{dom}\, f} \|x - y\|_\infty. \end{aligned}$$

We construct an auxiliary network $G_{\xi,x}$ with respect to flow ξ and base $x \in \mathrm{dom}\, f$. Let $G_{\xi,x} = (V, A_{\xi,x})$ be a graph with vertex set V and arc set $A_{\xi,x} = A_\xi \cup B_\xi \cup C_x$ consisting of three disjoint parts:

$$\begin{aligned}
A_\xi &= \{a \mid a \in A, \xi(a) < \overline{c}(a)\},\\
B_\xi &= \{\overline{a} \mid a \in A, \underline{c}(a) < \xi(a)\} \ \ (\overline{a}\text{: reorientation of } a),\\
C_x &= \{(u,v) \mid u, v \in V, u \neq v, \exists \alpha > 0 : x - \alpha(\chi_u - \chi_v) \in \mathrm{dom}\, f\}.
\end{aligned}$$

We define a function $c : A_{\xi,x} \to \mathbf{Z}$, representing arc capacities, by

$$c(a) = \begin{cases} \overline{c}(a) - \xi(a) & (a \in A_\xi) \\ \xi(\overline{a}) - \underline{c}(\overline{a}) & (a \in B_\xi, \overline{a} \in A) \\ \tilde{c}(x, v, u) & (a = (u,v) \in C_x). \end{cases}$$

We also define a function $l : A_{\xi,x} \to \mathbf{R}$, representing arc length, by

$$l(a) = \begin{cases} \gamma(a) & (a \in A_\xi) \\ -\gamma(\overline{a}) & (a \in B_\xi, \overline{a} \in A) \\ \Delta f(x; v, u) & (a = (u,v) \in C_x), \end{cases}$$

where

$$\Delta f(x; v, u) = f(x - \chi_u + \chi_v) - f(x).$$

Given a potential (or price function) $p \in \mathbf{R}^V$, we define the reduced length w.r.t. p as

$$l_p(a) = l(a) + p(\partial^+ a) - p(\partial^- a)$$

for each $a \in A_{\xi,x}$. In terms of the reduced length, Theorem 2.5 can be put in a more convenient and tractable form, which is used in designing efficient algorithms for the M-convex submodular flow problem MSFP$_2$. By (2.9), (2.10) and the definition of $l_p(a)$ for $a \in A_\xi \cup B_\xi$, the condition (i) in Theorem 2.5 is equivalent to the condition $l_p(a) \geq 0 \quad (a \in A_\xi \cup B_\xi)$. Theorem 2.2 shows

$$\partial\xi \in \arg\min f[-p] \iff \Delta f(\partial\xi; v, u) + p(u) - p(v) \geq 0 \quad (u, v \in V).$$

This expression leads to the condition that $l_p(a) \geq 0$ for all $a \in C_x$ with $x = \partial\xi$.

Theorem 3.1 ([16, 17, 18], Optimality condition by the reduced length). *A flow $\xi : A \to \mathbf{Z}$ satisfying (2.6) and (2.7) is optimal if and only if there exists $p : V \to \mathbf{R}$ such that*

$$l_p(a) \geq 0 \tag{3.1}$$

for all $a \in A_{\xi,x}$ with $x = \partial\xi$.

This theorem suggests the algorithm SSP. The algorithm keeps flow ξ, base x and potential p, and computes augmentation of ξ and x with reference to l_p on $A_{\xi,x}$. This ensures that ξ remains a flow. The algorithm iteratively modifies ξ, x, and p so that (3.1) is preserved for all $a \in A_{\xi,x}$, and so that $\partial\xi$ and x eventually coincide. At the end of the algorithm, we have $\partial\xi = x$ as well as the optimality condition (3.1). This means we have an optimal solution of MSFP$_2$.

We regard the reduced length $l_p(a) = l(a) + p(\partial^+ a) - p(\partial^- a)$ as the length of $a \in A_{\xi,x}$. Define

$$S^+ = \{v \mid x(v) > \partial\xi(v)\}, \quad S^- = \{v \mid x(v) < \partial\xi(v)\}$$

as the sets of vertices where $\partial\xi$ differs from x. In the auxiliary network $G_{\xi,x}$ with nonnegative reduced length, let $d : V \to \mathbf{R}$ represent the shortest path distances from S^+ to each $v \in V$ computed by using Dijkstra's algorithm and P be a shortest path from S^+ to S^- with a minimum number of arcs. It follows from the inequality $d(\partial^- a) \leq d(\partial^+ a) + l_p(a)$ that

$$l(a) + p(\partial^+ a) - p(\partial^- a) + d(\partial^+ a) - d(\partial^- a) \geq 0$$

holds for every arc $a \in A_{\xi,x}$. Hence updating $p(v)$ to $p(v) + \min\{d(v), \sum_{a \in P} l_p(a)\}$ retains the nonnegativity of the reduced length. In particular, the reduced length becomes zero for each arc in the shortest path P. In order to reduce $\sum_{v \in S^+}\{x(v) - \partial\xi(v)\}$, the algorithm augments along a shortest path (with respect to l_p) from S^+ to S^- with a minimum number of arcs and updates the base x. It is clear that augmenting ξ along P by a unit flow does not violate the capacity constraints. The base x is also updated along P so that $x(v) = \partial\xi(v)$ holds for every inner vertex v of P. For the starting point $s \in S^+$ of the shortest path P, either $\partial\xi(s)$ increases or $x(s)$ decreases by one. Hence $\sum_{v \in S^+}\{x(v) - \partial\xi(v)\}$ reduces exactly by one. The nonnegativity of the reduced length is preserved by Lemma 3.3 below. Repeat this process until the source S^+ and consequently the sink S^- become empty. When $\partial\xi$ coincides with x, the flow ξ is an optimal solution.

The details of this algorithm SSP are the following.

Algorithm : SSP

```
S0: Find $\xi \in \mathbf{Z}^A$ satisfying (2.6) and $x \in \arg\min f$. Set $p := 0$.
S1: Repeat the following Steps (1-1)–(1-3), until $S^+ = \emptyset$.
    1-1: Compute the shortest distance $d(v)$ from $S^+$ to each $v \in V \setminus S^+$ in
         $G_{\xi,x}$ with respect to the arc length $l_p$. Among the shortest paths from
         $S^+$ to $S^-$, let $P$ be one with a minimum number of arcs. (If there is no
         path from $S^+$ to $S^-$, then the problem is infeasible.)
    1-2: For each $v \in V$, put $p(v) := p(v) + \min\{d(v), \sum_{a \in P} l_p(a)\}$.
    1-3: For each arc $a \in P$,
            $a \in A_\xi \Rightarrow \xi(a) := \xi(a) + 1,$
            $a \in B_\xi \Rightarrow \xi(\overline{a}) := \xi(\overline{a}) - 1,$
            $a \in C_x \Rightarrow x(\partial^+ a) := x(\partial^+ a) - 1, x(\partial^- a) := x(\partial^- a) + 1.$
```

□

In the algorithm SSP, we do not need to evaluate the value of $c(a)$. We have only to know whether $c(a)$ is positive or zero.

To complete the proof of correctness, we show that the algorithm maintains the reduced length optimality (3.1). The following lemma is a more convenient and tractable form of lemma 2.4 for the proof.

Lemma 3.2. *Suppose that $x \in \arg\min f[-p]$ and that (u_1, v_1), (u_2, v_2), ..., $(u_r, v_r) \in C_x$ have distinct end-vertices. If $(u_i, v_i) \in C_x \cap \{a \mid l_p(a) = 0\}$ for $i = 1, \ldots, r$ and $(u_i, v_j) \notin C_x \cap \{a \mid l_p(a) = 0\}$ for any $i < j$, then $y = x + \sum_{i=1}^{r}(\chi_{v_i} - \chi_{u_i}) \in \arg\min f[-p]$.*

Lemma 3.3. *After Step 1-3, the condition of the reduced length optimality (3.1) is maintained.*

Proof. If there is a new arc after Step 1-3, it is either the reverse arc of an arc a on P or the new exchange arc.

First we consider the case that there exists a new reverse arc of an arc a on P. Since P is a shortest path, the reduced length of any arc on P is zero, and hence the reduced length of the reverse arc is zero. For any arc in $A_\xi \cup B_\xi$ that also exists before the update in Step 1-3, the reduced length optimality condition (3.1) is obviously maintained.

Let x' denote the base after Step 1-3. If there are k exchange arcs (u_i, v_i) for $i = 1, \ldots, k$ in P, then

$$x' = x + \sum_{i=1}^{k}(\chi_{v_i} - \chi_{u_i}).$$

We must check (3.1) for each exchange arc with respect to the updated base $x' = x + \sum_{i=1}^{k}(\chi_{v_i} - \chi_{u_i})$, which is

$$f(x' - \chi_s + \chi_t) - f(x') + p(s) - p(t) \geq 0 \quad (\forall s, t \in V). \tag{3.2}$$

Since P has a minimum number of arcs among the shortest paths, the numbering (u_1, v_1), (u_2, v_2), ..., (u_r, v_r) of the arcs in $P \cap C_x$ along the path P has the property that $(u_i, v_i) \in C_x \cap \{a \mid l_p(a) = 0\}$ for $i = 1, \ldots, r$ and $(u_i, v_j) \notin C_x \cap \{a \mid l_p(a) = 0\}$ for any $i < j$. It follows from lemma 3.2 that $x' \in \arg\min f[-p]$, and hence (3.2). An alternative proof is given in Appendix. □

To talk about running time we use n for the number of vertices, m for the number of arcs, and F for the upper bound on the time to evaluate f. Since $|\xi(a)| \leq C$

for all $a \in A$, $|\partial\xi(v)| \leq (n-1)C$ for all $v \in V$. Also, we have $|x(v)| \leq C$ for all $v \in V$. Thus the initial discrepancy $\| x - \partial\xi \|_1$ between x and $\partial\xi$ is at most n^2C. Since $x(S^+) - \partial\xi(S^+)$ is always a nonnegative integer and decreases with each augmentation, the algorithm terminates in $\mathrm{O}(n^2C)$ iterations. Since the number of arcs in $G_{\xi,x}$ is $|A_\xi \cup B_\xi| + |C_x| = \mathrm{O}(m) + \mathrm{O}(n^2) = \mathrm{O}(n^2)$, the bottleneck Dijkstra computation takes $\mathrm{O}(F \cdot n^2)$ time, so that each augmentation requires $\mathrm{O}(F \cdot n^2)$ time. Thus the total time complexity of algorithm SSP is $\mathrm{O}(F \cdot n^4C)$.

4 A Capacity Scaling Algorithm

4.1 Algorithm Description

We present a capacity scaling algorithm for the scalable M-convex submodular flow problem, which performs a number of scaling phases for different values of a scaling parameter α. Each scaling phase is a successive shortest path algorithm, where the amount of augmentation at once is exactly α. When there is no possibility of augmentation, the algorithm reduces the value of α by a factor of two.

A scaling phase with a specific value of α is referred to as the α-scaling phase. An auxiliary graph in an α-scaling phase is referred to as the α-auxiliary graph $G^\alpha_{\xi,x} = (V, A^\alpha_{\xi,x}) = (V, A^\alpha_\xi \cup B^\alpha_\xi \cup C^\alpha_x)$, where

$$\begin{aligned} A^\alpha_\xi &:= \{a \mid a \in A_\xi, c(a) \geq \alpha\}, \\ B^\alpha_\xi &:= \{a \mid a \in B_\xi, c(a) \geq \alpha\}, \\ C^\alpha_x &:= \{a \mid a \in C_x, c(a) \geq \alpha\}. \end{aligned}$$

We define a function $l^\alpha : A^\alpha_{\xi,x} \to \mathbf{R}$, representing arc length, by

$$l^\alpha(a) = \begin{cases} \alpha\gamma(a) & (a \in A^\alpha_\xi) \\ -\alpha\gamma(\overline{a}) & (a \in B^\alpha_\xi, \overline{a} \in A) \\ f(x + \alpha(\chi_v - \chi_u)) - f(x) & (a = (u,v) \in C^\alpha_x). \end{cases}$$

Given a potential $p \in \mathbf{R}^V$, we define

$$l^{\alpha}_{p}(a) = l^{\alpha}(a) + p(\partial^{+}a) - p(\partial^{-}a)$$

for each $a \in A^{\alpha}_{\xi,x}$.

Initially, the value of α is set to be $2^{\lfloor \log C \rfloor}$. The flow ξ and the potential p are initialized by $\xi(a) = 0$ for each arc $a \in A$ and $p(v) = 0$ for each vertex $v \in V$, respectively. We assume that this ξ satisfies (2.6) and (2.7) without loss of generality.

In the α-scaling phase, the algorithm keeps a flow ξ, a potential p, and a base x such that $\xi(a)$ is a multiple of α for every arc $a \in A$, $x(v)$ is a multiple of α for every vertex $v \in V$ and that $l^{\alpha}_{p}(a) \geq 0$ holds for every arc $a \in A^{\alpha}_{\xi,x}$. In particular the condition that $l^{\alpha}_{p}(a) \geq 0$ for $a \in C^{\alpha}_{x}$ means that x is α-local minimum of $f[-p]$. Moreover, by the scalability of f, when x is α-local minimum of $f[-p]$, we have $x/\alpha \in \arg\min(f[-p])^{\alpha}$. Here $x/\alpha \in \arg\min(f[-p])^{\alpha}$ means that x/α is a minimizer of $f(\alpha x) - \langle p, \alpha x \rangle$. In order to reduce the discrepancy between x and $\partial\xi$, measured by $\parallel x - \partial\xi \parallel_1 = \sum_v |x(v) - \partial\xi(v)|$, it repeats augmentations of the flow ξ by α along a shortest path from $S^{+}(\alpha)$ to $S^{-}(\alpha)$ with respect to l^{α}_{p}, where

$$S^{+}(\alpha) := \{v \mid x(v) - \partial\xi(v) \geq \alpha\}, \quad S^{-}(\alpha) := \{v \mid \partial\xi(v) - x(v) \geq \alpha\}.$$

It also updates the potential p and base x in order to reduce the discrepancy $\parallel x - \partial\xi \parallel_1$ and retain the nonnegativity of the reduced length. Repeat this process until the source $S^{+}(\alpha)$ and consequently the sink $S^{-}(\alpha)$ become empty. At the end of a phase we set $\alpha := \alpha/2$ and continue until $\alpha = 1$, at which point we can finish using SSP.

An algorithmic description of the capacity scaling algorithm is now given as follows. In this algorithm, we assume that $\underline{c} \leq \mathbf{0}$, $\overline{c} \geq \mathbf{0}$ and $\mathbf{0} \in \mathrm{dom}\, f$; if not, this assumption is satisfied by translations.

Algorithm : capacity scaling

S0: Set $\alpha := 2^{\lfloor \log C \rfloor}, \xi := 0$ and $p := 0$.

S1: Find x with $x/\alpha \in \arg\min(f[-p])^{\alpha}$ and $\| x - \partial\xi \|_{\infty} \leq (n-1)\alpha$. For each $a \in A_{\xi}^{\alpha}$ do if $l_{p}^{\alpha}(a) < 0$ then $\xi(a) := \xi(a) + \alpha$. For each $a \in B_{\xi}^{\alpha}$ do if $l_{p}^{\alpha}(a) < 0$ then $\xi(\overline{a}) := \xi(\overline{a}) - \alpha$.

S2: Repeat the following Steps (2-1)–(2-3), until $S^{+}(\alpha) = \emptyset$.

> **2-1:** Compute the shortest distance $d(v)$ from $S^{+}(\alpha)$ to each $v \in V \setminus S^{+}(\alpha)$ in $G_{\xi,x}^{\alpha}$ with respect to the arc length l_{p}^{α}. Among the shortest paths from $S^{+}(\alpha)$ to $S^{-}(\alpha)$, let P be one with a minimum number of arcs.
>
> **2-2:** For each $v \in V$, put $p(v) := p(v) + \min\{d(v), \sum_{a \in P} l_{p}^{\alpha}(a)\}$.
>
> **2-3:** For each arc $a \in P$,
>
> $$\begin{aligned} a \in A_{\xi}^{\alpha} &\quad\Rightarrow\quad \xi(a) := \xi(a) + \alpha, \\ a \in B_{\xi}^{\alpha} &\quad\Rightarrow\quad \xi(\overline{a}) := \xi(\overline{a}) - \alpha, \\ a \in C_{x}^{\alpha} &\quad\Rightarrow\quad x(\partial^{+}a) := x(\partial^{+}a) - \alpha, x(\partial^{-}a) := x(\partial^{-}a) + \alpha. \end{aligned}$$

S3: If $\alpha > 1$, then $\alpha := \alpha/2$ and go to S1. Else, stop. □

In the capacity scaling algorithm, we do not need to evaluate the value of $c(a)$ to determine the arc set C_{x}^{α}. We have only to check whether $c(a) \geq \alpha$ or not.

At the start of a new α-scaling phase, i.e., Step 1, we find an α-local minimum x of $f[-p]$ that lies close to $\partial\xi$, and modify ξ to remove the arcs with negative reduced length from the auxiliary graph $G_{\xi,x}^{\alpha}$. Note that the algorithm also updates the α-auxiliary graph $G_{\xi,x}^{\alpha}$ after this adjustment.

4.2 Correctness and Time Complexity

The key to the correctness of the algorithm is to maintain the condition $l_{p}^{\alpha}(a) \geq 0$ for all arcs with residual capacity at least α.

Lemma 4.1. *After each augmentation, the condition $l_{p}^{\alpha}(a) \geq 0$ holds for all arcs with residual capacity at least α.*

Proof. Let ξ and x be values before Step 2-3, and ξ' and x' values after Step 2-3. We must show that each $a \in A^{\alpha}_{\xi',x'}$ has nonnegative reduced length after the augmentation and update of p using the distance labels $d(v)$. Let p be the old potential, and q the new potential. For any previously existing arc a with nonnegative reduced length, we have $l^{\alpha}_{q}(a) = l^{\alpha}_{p}(a) + \min\{d(\partial^{+}a), l^{\alpha}_{p}(P)\} - \min\{d(\partial^{-}a), l^{\alpha}_{p}(P)\}$ where $l^{\alpha}_{p}(P) := \sum_{a \in P} l^{\alpha}_{p}(a)$. Since the distance labels satisfy $l^{\alpha}_{p}(a) + d(\partial^{+}a) \geq d(\partial^{-}a)$, we have $l^{\alpha}_{p}(a) + \min\{d(\partial^{+}a), l^{\alpha}_{p}(P)\} \geq \min\{l^{\alpha}_{p}(a) + d(\partial^{+}a), l^{\alpha}_{p}(P)\} \geq \min\{d(\partial^{-}a), l^{\alpha}_{p}(P)\}$, which implies $l^{\alpha}_{q}(a) \geq 0$. If there is an arc with residual capacity newly at least α, it is the reverse arc of an arc a on P. Since P is a shortest path, we have $l^{\alpha}_{q}(a) = 0$, and hence $l^{\alpha}_{q}(\overline{a}) = 0$. □

Lemma 4.2. *The condition* $l^{\alpha}_{p}(a) \geq 0 \;\; (a \in A^{\alpha}_{\xi,x})$ *is maintained at the start of each* α*-scaling phase.*

Proof. At the start of each α-scaling phase, i.e., Step 1, we modify ξ. The modification of ξ removes the arcs with negative reduced length from the auxiliary graph $G^{\alpha}_{\xi,x}$, and creates the reverse arcs with nonnegative reduced length, so the condition $l^{\alpha}_{p}(a) \geq 0 \;\; (a \in A^{\alpha}_{\xi,x})$ is maintained. □

We conclude our paper by analyzing the time complexity of the capacity scaling algorithm. Proximity theorem 2.1 guarantees the existence[1] of x with $x/\alpha \in \arg\min(f[-p])^{\alpha}$ and $\| x - \partial\xi \|_{\infty} \leq (n-1)\alpha$. Hence the base x in Step 1 can be found as a minimizer of $\tilde{f} : \mathbf{Z}^{V} \to \mathbf{R} \cup \{+\infty\}$ defined by

$$\tilde{f}(y) = \begin{cases} f[-p](\partial\xi + \alpha y) & (y \in B^{\alpha}) \\ +\infty & (y \notin B^{\alpha}) \end{cases}$$

where

$$B^{\alpha} = \{y \in \mathbf{Z}^{V} \mid \;\; \| y \|_{\infty} \leq n-1\}.$$

[1] An optimal solution to the original problem $f[-p]$ may not exist in the neighborhood $\{x \mid \| x - \partial\xi \|_{\infty} \leq (n-1)\alpha\}$ of $\partial\xi$, but surely in a larger one $\{x \| \| x - \partial\xi \|_{\infty} \leq (n-1)(2\alpha - 1)\}$.

This implies that we can find $x/\alpha \in \arg\min(f[-p])^\alpha$ in $\mathrm{O}(F \cdot n^3)$ time by a descent algorithm proposed in [11]; note that $\tilde{f}$ is an M-convex function as a consequence of the assumed scalability of f.

In an α-scaling phase, the discrepancy between x and $\partial\xi$ decreases by α after Step 2-3. After finding x with $x/\alpha \in \arg\min(f[-p])^\alpha$ and $\| x - \partial\xi \|_\infty \leq (n-1)\alpha$ at Step 1 of a new α-scaling phase, we have

$$\| x - \partial\xi \|_1 \leq n(n-1)\alpha < \alpha n^2.$$

Then we modify ξ to satisfy the condition $l_p^\alpha(a) \geq 0 \quad (a \in A_{\xi,x}^\alpha)$ for arcs a with residual capacity $\alpha \leq c(a) < 2\alpha$. After this modification of ξ, we have

$$\| x - \partial\xi \|_1 < \alpha n^2 + 2m \cdot \alpha.$$

Thus each scaling phase performs the shortest path augmentation at most $n^2 + 2m$ times. Each construction of the auxiliary graph can be done by $\mathrm{O}(n^2)$ evaluations of exchange capacity and f. Since we set $\alpha = 2^{\lfloor \log C \rfloor}$ initially, after $\mathrm{O}(\log C)$ scaling phases, we have $\alpha = 1$. Thus the total time complexity of the capacity scaling algorithm is $\mathrm{O}(F \cdot (n^3 + n^4) \log C) = \mathrm{O}(F \cdot n^4 \log C)$.

Appendix

An alternative proof of nonnegativity (3.2) for each exchange arc in lemma 3.3 is given here. This proof makes explicit use of the exchange axiom (M-EXC) of M-convex functions.

Put $y = x' - \chi_s + \chi_t$. By (M-EXC) there exist $a_1 \in \mathrm{supp}^+(x-y)$ and $b_1 \in \mathrm{supp}^-(x-y)$ such that

$$f(y) \geq [f(x - \chi_{a_1} + \chi_{b_1}) - f(x)] + f(y_2),$$

where $y_2 = y + \chi_{a_1} - \chi_{b_1}$. By (M-EXC) applied to (x, y_2), there exist $a_2 \in \mathrm{supp}^+(x - y_2)$ and $b_2 \in \mathrm{supp}^-(x - y_2)$ such that

$$f(y_2) \geq [f(x - \chi_{a_2} + \chi_{b_2}) - f(x)] + f(y_3),$$

where $y_3 = y_2 + \chi_{a_2} - \chi_{b_2} = y + \chi_{a_1} + \chi_{a_2} - \chi_{b_1} - \chi_{b_2}$. Repeating this $l = \| x - y \|_1 /2$ times, we obtain (a_i, b_i) $(i = 1, \ldots, l)$ such that $y = x + \sum_{i=1}^{l} (\chi_{b_i} - \chi_{a_i}) = x' - \chi_s + \chi_t$ and

$$\begin{aligned} f(x' - \chi_s + \chi_t) &\geq f(x) + \sum_{i=1}^{l} [f(x - \chi_{a_i} + \chi_{b_i}) - f(x)] \\ &\geq f(x) + \sum_{i=1}^{l} [p(b_i) - p(a_i)] \\ &= f(x) + \sum_{i=1}^{k} (p(v_i) - p(u_i)) - p(s) + p(t), \end{aligned} \tag{A.1}$$

where the second inequality is due to (3.1). See also Proposition 4.17 in [17, 18].

To evaluate $f(x')$, the second term in (3.2), we consider a bipartite graph $G(x, x') = (V^+, V^-; \hat{A})$ with vertex sets $V^+ = \{u_1, \ldots, u_k\}$ and $V^- = \{v_1, \ldots, v_k\}$ and arc set

$$\hat{A} = \{(u, v) \mid u \in V^+, v \in V^-, x - \chi_u + \chi_v \in \mathrm{dom}\, f\},$$

and associate $c(u, v) = \Delta f(x; v, u)$ with arc $(u, v) \in \hat{A}$ as its weight. We say that (x, x') satisfies *unique-min condition* if there exists in $G(x, x')$ exactly one minimum-weight perfect matching with respect to c.

The following lemma gives a necessary and sufficient condition for a bipartite graph to have a unique minimum-weight perfect matching. It also shows that the unique-min condition for a pair of integer vectors can be checked by an efficient algorithm.

Lemma A.1 ([12, 17, 18]). *Let $G = (V^+, V^-; \hat{A})$ be a bipartite graph with $|V^+| = |V^-| (= k)$, and $c : V^+ \times V^- \to \mathbf{R} \cup \{+\infty\}$ be a weight function such that: $c(u, v) <$*

$+\infty \iff (u,v) \in \hat{A}$. *There exists a unique minimum-weight perfect matching if and only if there exists a potential* $p : V^+ \cup V^- \to \mathbf{R}$ *and orderings of vertices* $V^+ = \{u_1, \ldots, u_k\}$ *and* $V^- = \{v_1, \ldots, v_k\}$ *such that*

$$c(u_i, v_j) + p(u_i) - p(v_j) \begin{cases} = 0 & (1 \le i = j \le k) \\ \ge 0 & (1 \le j < i \le k) \\ > 0 & (1 \le i < j \le k). \end{cases} \tag{A.2}$$

Since P in Step 1-1 of SSP has a minimum number of arcs among the shortest paths, the condition (A.2) holds, and it then follows from lemma A.1 that (x, x') satisfies the unique-min condition.

Denote by $\check{f}(x, x')$ the minimum weight of a perfect matching in $G(x, x')$, where $\check{f}(x, x') = +\infty$ if no perfect matching exists. The following lemma is a quantitative extension of no-shortcut lemma 2.3.

Lemma A.2 ([12, 17, 18]). *Let* f *be an M-convex function, and assume* $x \in \mathrm{dom}\, f$, $x' \in \mathbf{Z}^V$ *and* $\| x - x' \|_\infty = 1$. *If* (x, x') *satisfies the unique-min condition, then* $x' \in \mathrm{dom}\, f$ *and*

$$f(x') - f(x) = \check{f}(x, x').$$

Note that $\| x - x' \|_\infty = 1$. Lemma A.2 shows

$$f(x') = f(x) + \sum_{i=1}^{k} \Delta f(x; v_i, u_i) = f(x) + \sum_{i=1}^{k} (p(v_i) - p(u_i)). \tag{A.3}$$

The inequality (3.2) follows from (A.1) and (A.3).

Acknowledgement

The authors thank Akihisa Tamura for a critical comment. This work is supported by the Superrobust Computation Project in the 21st Century COE Program and a Grant-in-Aid of the Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

[1] W.H. Cunningham and A. Frank (1985). A Primal-dual Algorithm for Submodular Flows, *Math. Oper. Res.*, **10**, 251–262.

[2] A.W.M. Dress and W. Wenzel (1992). Valuated Matroids, *Adv. Math.*, **93**, 214–250.

[3] J. Edmonds and R. Giles (1977). A Min-max Relation for Submodular Functions on Graphs, *Ann. Discrete Math.*, **1**, 185–204.

[4] J. Edmonds and R.M. Karp (1972). Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems, *Journal of the ACM*, **19**, 248–264.

[5] L. Fleischer, S. Iwata and S. T. McCormick (2002). A Faster Capacity Scaling Algorithm for Minimum Cost Submodular Flow, *Math. Program.*, **Ser. A 92**, 119–139.

[6] A. Frank (1984). Finding Feasible Vectors of Edmonds-Giles Polyhedra, *Journal of Combinatorial Theory*, **Ser. B 36**, 221–239.

[7] A. Frank and É. Tardos (1988). Generalized Polymatroids and Submodular Flows, *Math. Program.*, **42**, 489–563.

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

[9] S. Iwata (1997). A Capacity Scaling Algorithm for Convex Cost Submodular Flows, *Math. Program.*, **76**, 299–308.

[10] S. Iwata and M. Shigeno (2002). Conjugate Scaling Algorithm for Fenchel-type Duality in Discrete Convex Optimization, *SIAM J. Optimization*, **13**, 204–211.

[11] S. Moriguchi, K. Murota and A. Shioura (2002). Scaling Algorithms for M-convex Function Minimization, *IEICE Transactions on Fundamentals*, **E85-A**, 922–929.

[12] K. Murota (1996). Valuated Matroid Intersection, I: optimality criteria, *SIAM J. Discrete Math.*, **9**, 545–561.

[13] K. Murota (1996). Valuated Matroid Intersection, II: Algorithms, *SIAM J. Discrete Math.*, **9**, 562–576.

[14] K. Murota (1996). Convexity and Steinitz's Exchange Property, *Adv. Math.*, **124**, 272–311.

[15] K. Murota (1998). Discrete Convex Analysis, *Math. Program.*, **83**, 313–371.

[16] K. Murota (1999). Submodular Flow Problem with a Nonseparable Cost Function, *Combinatorica*, **19**, 87–109.

[17] K. Murota (2001). *Discrete Convex Analysis–An Introduction.* Kyoritsu Publishing Co., Tokyo (In Japanese).

[18] K. Murota (2003). *Discrete Convex Analysis.* Society for Industrial and Applied Mathematics, Philadelphia.

[19] K. Murota and A. Tamura (2001). Application of M-convex Submodular Flow Problem to Mathematical Economics. In: P. Eades and T. Takaoka (Eds.), *Proceedings of 12th International Symposium on Algorithms and Computation, Lecture Notes in Computer Science*, **2223**, pp. 14–25. Springer-Verlag, Berlin Heidelberg New York.