INVITED SURVEY PAPER *Special Issue on Algorithm Engineering: Surveys*

# Algorithms for Submodular Flows*

**Satoru FUJISHIGE**†, ***Member and*** **Satoru IWATA**†, ***Nonmember***

**SUMMARY** We first describe fundamental results about submodular functions and submodular flows, which lay a basis for devising efficient algorithms for submodular flows. We then give a comprehensive survey on algorithms for submodular flows and show some possible future research directions.
***key words:*** *algorithms, network flows, submodular functions*

## 1. Introduction

The submodular flow problem, introduced by Edmonds and Giles [11], is one of the most important frameworks of efficiently solvable combinatorial optimization problems. It includes the minimum cost flow, the graph orientation, the polymatroid intersection, and the directed cut covering problems as its special cases. See also [23], [24] for further applications of submodular flows. Other frameworks named independent flows [25] and polymatroidal flows [47], [60] are equivalent to submodular flows. These three are collectively called neoflows in [29].

A number of combinatorial algorithms have been proposed as extensions of network flow algorithms. This paper surveys the state of the art in the developments of the submodular flow algorithms and also describes possible directions for further investigations.

In Sect. 2, we describe fundamental results concerning submodular functions and submodular flows. Readers should refer to [24], [29], [61] for more details of these results. In Sect. 3, we survey algorithms for submodular flows. We also show some possible future research directions in Sect. 4.

## 2. Submodular Flows

### 2.1 Submodular Functions and Base Polyhedra

Let $N$ be a finite nonempty set and $\mathcal{D}$ a family of its subsets with $\emptyset, N \in \mathcal{D}$. Suppose that $X \cup Y, X \cap Y \in \mathcal{D}$ holds for any pair of $X, Y \in \mathcal{D}$. A function $f : \mathcal{D} \to \mathbf{R}$ is said to be *submodular* if

$$f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y) \qquad (1)$$

holds for any $X, Y \in \mathcal{D}$. When $f(\emptyset) = 0$, the pair $(\mathcal{D}, f)$ is called a *submodular system*.

The set $\mathbf{R}^N$ of all the functions $x : N \to \mathbf{R}$ forms an $|N|$-dimensional linear space. For each $v \in N$, we define $\vec{v} \in \mathbf{R}^N$ as $\vec{v}(v) = 1$ and $\vec{v}(u) = 0$ for $u \neq v$.

Let $\mathbf{R}_N$ denote the dual linear space of $\mathbf{R}^N$ and $\langle p, x\rangle$ the inner product of $p \in \mathbf{R}_N$ and $x \in \mathbf{R}^N$. We may use $\langle p, x\rangle_N$ if we want to emphasize the underlying set $N$. For $p \in \mathbf{R}_N$ we write $p(\vec{v}) = \langle p, \vec{v}\rangle$ as $p(v)$ for simplicity.

The characteristic vector $\chi_X \in \mathbf{R}_N$ of $X \subseteq N$ is defined by $\chi_X(v) = 1$ for $v \in X$ and $\chi_X(v) = 0$ for $v \notin X$. A vector $x \in \mathbf{R}^N$ is often identified with a modular function defined by $x(X) = \langle \chi_X, x\rangle$.

Associated with a submodular system $(\mathcal{D}, f)$, the *submodular polyhedron* $\mathrm{P}(f)$ and the *base polyhedron* $\mathrm{B}(f)$ in $\mathbf{R}^N$ are defined as follows:

$$\mathrm{P}(f) = \{x \mid x \in \mathbf{R}^N,\ \forall X \in \mathcal{D} : x(X) \leq f(X)\},$$
$$\mathrm{B}(f) = \{x \mid x \in \mathrm{P}(f),\ x(N) = f(N)\}.$$

Consider the linear program

$$\langle \mathrm{LP}\rangle \quad \text{Maximize } \langle p, x\rangle \quad \text{subject to } x \in \mathrm{B}(f)$$

on the base polyhedron. An optimal solution is called a *p-maximum base*. Let $p_1 > \cdots > p_k$ be the distinct values of $p(v)$, and put $N_i = \{v \mid p(v) \geq p_i\}$. Then the following theorem characterizes the $p$-maximum bases. Note that, if $N_i \notin \mathcal{D}$ for some $i$, then the linear program is unbounded and there is no optimal solution.

**Theorem 2.1:** (Edmonds [10], Fujishige–Tomizawa [31]): A base $x \in \mathrm{B}(f)$ is $p$-maximum if and only if $x(N_i) = f(N_i)$ holds for every $i$. □

Theorem 2.1 implies that the set of $p$-maximum bases, $\mathrm{B}_p(f)$, is also the base polyhedron of some submodular system. Note that $\mathrm{B}_p(f)$ is a face of $\mathrm{B}(f)$.

**Corollary 2.2:** Let $f_p : \mathcal{D} \to \mathbf{R}$ be defined by

$$f_p(X) = \sum_{i=1}^{k} \{f((X \cap N_i) \cup N_{i-1}) - f(N_{i-1})\},$$

where $N_0 = \emptyset$. Then $f_p$ is a submodular function, and $\mathrm{B}(f_p) = \mathrm{B}_p(f)$ holds. □

Manuscript received July 2, 1999.
Manuscript revised September 2, 1999.
†The authors are with Division of Systems Science, Graduate School of Engineering Science, Osaka University, Toyonaka-shi, 560-8531 Japan.
*This work is supported by a Grant-in-Aid for Scientific Research from the Ministry of Education, Science, Sports and Culture of Japan.

### 2.2 Exchange Capacity

For a base $x \in \mathrm{B}(f)$ and a pair of distinct nodes $v, u \in N$, we define the *exchange capacity* $\widetilde{c}(x, v, u)$ by

$$\widetilde{c}(x, v, u) = \max\{\alpha \mid \alpha \in \mathbf{R},\ x + \alpha(\vec{v} - \vec{u}) \in \mathrm{B}(f)\}.$$

The exchange capacity can be written as

$$\begin{aligned}&\widetilde{c}(x, v, u)\\ &\quad = \min\{f(X) - x(X) \mid v \in X \in \mathcal{D},\ u \notin X\}.\end{aligned}$$

The *exchangeability graph* is a directed graph with the node set $N$ and the arc set $E_x = \{(u, v) \mid \widetilde{c}(x, v, u) > 0\}$. The exchangeability graph is transitive. Namely, $(u, v) \in E_x$ and $(v, w) \in E_x$ imply $(u, w) \in E_x$. The following lemma characterizes the optimality of a base in terms of the exchangeability graph.

**Lemma 2.3:** A base $x \in \mathrm{B}(f)$ is $p$-maximum if and only if $p(u) \geq p(v)$ holds for any $(u, v) \in E_x$. □

### 2.3 Submodular Flow Problem

Let $G = (N, A)$ be a directed graph with upper and lower capacity bounds $\overline{c}, \underline{c} \in \mathbf{R}^A$ and the cost function $d \in \mathbf{R}_A$. For each $X \subseteq N$, let $\Delta^+X$ ($\Delta^-X$) denotes the set of of arcs leaving (entering) $X$. Consider a submodular system $(\mathcal{D}, f)$ on $N$ with $f(N) = 0$. Then the *submodular flow problem* is formulated as follows [11].

$$\begin{array}{lll}\langle\mathrm{SF}\rangle & \text{Minimize} & \langle d, \varphi\rangle_A \\ & \text{subject to} & \underline{c}(a) \leq \varphi(a) \leq \overline{c}(a) \quad (a \in A) \\ & & \partial\varphi \in \mathrm{B}(f),\end{array}$$

where we define $\partial\varphi(X) = \varphi(\Delta^+X) - \varphi(\Delta^-X)$ for each $X \subseteq N$. A feasible solution $\varphi$ is called a *submodular flow*. If $d(a^*) = -1$ for some specific arc $a^* \in A$ and $d(a) = 0$ for the other arcs $a \in A$, we call $\langle\mathrm{SF}\rangle$ the *maximum submodular flow problem*.

**Theorem 2.4:** There exists a submodular flow if and only if

$$\underline{c}(\Delta^+X) - \overline{c}(\Delta^-X) \leq f(X) \qquad (2)$$

holds for every $X \in \mathcal{D}$. If in addition $\overline{c}, \underline{c}, f$ are integer-valued, then there exists an integral submodular flow. □

Given a submodular flow $\varphi$, we have $\underline{c}(\Delta^+X) - \overline{c}(\Delta^-X) \leq \partial\varphi(X) \leq f(X)$, which implies (2). The converse follows from the linear programming formulation in Sect. 2.4 (except for the integrality property).

**Theorem 2.5:** A submodular flow $\varphi$ is optimal if and only if there exists an appropriate $p \in \mathbf{R}_N$ such that

**(a)** For any $a = (u, v) \in A$,

$$\begin{aligned}d(a) + p(u) > p(v) \ &\Rightarrow\ \varphi(a) = \underline{c}(a),\\ d(a) + p(u) < p(v) \ &\Rightarrow\ \varphi(a) = \overline{c}(a).\end{aligned}$$

**(b)** For any distinct $u, v \in N$,

$$p(u) < p(v) \Rightarrow \widetilde{c}(\partial\varphi, v, u) = 0.$$

Moreover, if $d$ is an integral vector, then we may restrict the above $p$ to be integral. □

### 2.4 Linear Programming Formulation

This section deals with a linear programming formulation of the submodular flow problem. We regard vectors in $\mathbf{R}^N$, $\mathbf{R}^A$, $\mathbf{R}^{\mathcal{D}}$ as column vectors and those in $\mathbf{R}_N$, $\mathbf{R}_A$, $\mathbf{R}_{\mathcal{D}}$ as row vectors.

Let $H : \mathcal{D} \times N \to \{0, 1\}$ be the matrix whose $(X, v)$-component is 1 if $v \in X$ and 0 otherwise. Denote by $D$ the incidence matrix of the directed graph $G = (N, A)$. Then the submodular flow problem $\langle\mathrm{SF}\rangle$ is written as

$$\begin{array}{lll}\langle\mathrm{LSF}\rangle & \text{Minimize} & d\varphi \\ & \text{subject to} & \underline{c} \leq \varphi \leq \overline{c} \\ & & x = D\varphi \\ & & Hx \leq f.\end{array}$$

The linear programming dual of this $\langle\mathrm{LSF}\rangle$ is

$$\begin{array}{lll}\langle\mathrm{DSF}\rangle & \text{Maximize} & \lambda\underline{c} - \mu\overline{c} - \xi f \\ & \text{subject to} & \xi H = \pi \\ & & \lambda - \mu - \pi D = d \\ & & \lambda, \mu, \xi \geq 0.\end{array}$$

The following lemma shows an uncrossing property of a dual optimal solution.

**Lemma 2.6:** There exists an optimal solution of $\langle\mathrm{DSF}\rangle$ such that $\mathcal{D}^* = \{X \mid X \in \mathcal{D},\ \xi(X) > 0\}$ forms a chain in $\mathcal{D}$. □

If $(\lambda, \mu, \pi, \xi)$ is a dual optimal solution with uncrossing property, $p = \pi$ satisfies the conditions (a) and (b), which completes the proof of Theorem 2.5 except for its integrality assertion. Theorem 2.4 also follows from Lemma 2.6 and the Farkas lemma, except for the integrality assertion.

### 2.5 Total Dual Integrality

A matrix is said to be *totally unimodular* if all the minors are $\pm 1$ or 0. By definition, every entry of totally unimodular matrix must be $\pm 1$ or 0. Incidence matrices of directed graphs and interval matrices are typical examples of totally unimodular matrices.

By the duality theorem in linear programming, a system of linear inequalies $Mx \leq b$ satisfies

$$\max\{dx \mid Mx \leq b\} = \min\{yb \mid yM = d,\ y \geq 0\},$$

provided that the left hand side is a finite value. In particular, if the coefficient matrix $M$ is totally unimodular and $d$ is an integer vector, then there exists an integer vector $y$ that achieves optimality of the right hand side.

A system of linear inequalities in general is said to be *totally dual integral* if every bounded feasible linear programming problem obtained by adding a linear objective function with integral coefficients allows an integral optimal solution to the dual problem. In particular, the system of linear inequalities $Mx \leq b$ with totally unimodular $M$ is totally dual integral.

**Theorem 2.7** (Hoffman [48], Edmonds–Giles [11]): If $b$ is an integral vector and $Mx \leq b$ is totally dual integral, then the polyhedron described by this system of linear inequalities is an integral polyhedron. □

The following theorem shows the total dual integrality of the system of linear inequalities that describes the submodular flow problem.

**Theorem 2.8** (Edmonds–Giles [11]): If $d$ is an integer vector and the dual problem ⟨DSF⟩ has an optimal solution, then ⟨DSF⟩ has an integral optimal solution. □

The integrality assertion of Theorem 2.5 follows immediately from the total dual integrality. In addition, it follows from Theorem 2.7 that a feasible submodular flow problem with integral $\overline{c}$, $\underline{c}$ and $f$ has an integral feasible solution, which implies the integrality assertion of Theorem 2.4.

### 2.6 Crossing-Submodular Functions

Edmonds and Giles [11] originally formulated the submodular flow problem in terms of crossing-submodular functions. This is because their motivation came from the Lucchesi–Younger theorem on directed-cut coverings [62].

A pair of subsets $X, Y \subseteq N$ is said to be *crossing* if $X \cap Y \neq \emptyset$, $X \backslash Y \neq \emptyset$, $Y \backslash X \neq \emptyset$, and $X \cup Y \neq N$. A *crossing family* is a family $\mathcal{F} \subseteq 2^N$ such that $X \cup Y, X \cap Y \in \mathcal{F}$ holds for any crossing pair of $X, Y \in \mathcal{F}$. A function $f : \mathcal{F} \to \mathbf{R}$ on a crossing family $\mathcal{F}$ is called *crossing-submodular* if the submodularity inequality (1) holds for each crossing pair $X, Y \in \mathcal{F}$. In particular, if $\emptyset, N \in \mathcal{F}$ and $f(\emptyset) = 0$, we consider a polyhedron

$$\mathrm{B}(f) = \left\{ x \,\middle|\, \begin{array}{l} x \in \mathbf{R}^N, x(N) = f(N) \\ \forall X \in \mathcal{F} : x(X) \leq f(X) \end{array} \right\},$$

which may possibly be empty.

Let $f^\sharp$ be a function on $\overline{\mathcal{F}} = \{N \backslash X \mid X \in \mathcal{F}\}$ defined by

$$f^\sharp(Y) = f(N) - f(N \backslash Y) \qquad (Y \in \overline{\mathcal{F}}).$$

Then we have the following theorem.

**Theorem 2.9** (Fujishige [26]): The polyhedron $\mathrm{B}(f)$ associated with a crossing-submodular function $f$ is nonempty if and only if every partition $\{Z_1, \cdots, Z_k\}$ of $N$ satisfies

$$\sum_{i=1}^{k} f(Z_i) \geq f(N)$$

$$\sum_{i=1}^{k} f^\sharp(Z_i) \leq f^\sharp(N).$$

Moreover, if $\mathrm{B}(f)$ is nonempty, there uniquely exists a submodular system $(\mathcal{D}, \widetilde{f})$ that satisfies $\mathrm{B}(f) = \mathrm{B}(\widetilde{f})$. □

If $\mathrm{B}(f)$ is nonempty, the function $\widetilde{f}$ in Theorem 2.9 is given by

$$\widetilde{f}(X) = \min \sum_{i \in I} \sum_{j \in J_i} f(X_{ij}), \tag{3}$$

where $\{X_i\}_{i \in I}$ with $X_i = \bigcap_{j \in J_i} X_{ij}$ is a partition of $N$ and $\{N \backslash X_{ij}\}_{j \in J_i}$ is a partition of $N \backslash X_i$ for each $i \in I$. Let $\mathcal{D}$ be the family of those $X$'s that allow such $\{X_{ij}\}_{ij}$. Then $\mathcal{D}$ forms a distributive lattice.

As a consequence of Theorem 2.9, it turns out that algorithms designed for submodular functions are also applicable to those problems described by crossing-submodular functions, provided that they rely only on exchange capacities for the associated base polyhedra. When it comes to the greedy algorithm, which refers the function values, we need some sophisticated technique such as the bitruncation algorithm of Frank and Tardos [24] (see also [68]). As for the submodular flow problems with crossing-submodular functions, we need to modify the feasibility condition (Theorem 2.4) [20], while the optimality condition (Theorem 2.5) remains valid. In the rest of this paper, we restrict ourselves to submodular functions for the sake of simplicity.

## 3. Submodular Flow Algorithms

When we talk about submodular flow algorithms, we usually assume that an oracle for computing exchange capacities is available. This assumption is justified by two different reasons. First, if we have an oracle for the submodular function value, we can compute an exchange capacity in strongly polynomial time [44], [52], [76]. Second, in most applications we can design efficient subroutines for computing exchange capacities. We henceforth denote by $h$ the time required for computing an exchange capacity. We also denote by $C$ and $U$ the maximum absolute value of arc costs and capacity bounds, respectively.

In order to design combinatorial algorithms for submodular flow problems, it is quite natural to attempt extensions of the existing network flow algorithms [2]. In fact, classical minimum cost flow algorithms such as the network simplex, primal-dual, out-of-kilter, and cycle canceling methods are successfully extended to solve the submodular flow problem by Barahona–Cunningham [5], Cunningham–Frank [8], Fujishige [28], and Zimmermann [86], respectively. Fujishige [25] also extended the successive-shortest-path and cycle canceling methods to the independent flow problem, which is equivalent to the submodular flow problem.

The first combinatorial algorithm for the submodular flow problem with polynomial running time is a cost-scaling primal-dual algorithm due to Cunningham–Frank [8]. This algorithm in its original form runs in $\mathrm{O}(mM \log C)$ time, where $m$ denotes the number of arcs and $M$ the time for solving a maximum submodular flow problem. A variant suggested by Fujishige, Röck, and Zimmermann [30] in the proof of their proximity lemma in fact can be used to reduce the bound to $\mathrm{O}(nM \log C)$.

The first polynomial algorithm for the maximum submodular flow problem, due to Frank [20], runs in $\mathrm{O}(n^5 h)$ time with the aid of the lexicographically shortest path technique originated by Schösleben [73] for polymatroid intersection and used by Lawler and Martel [60] for polymatroidal flows. Then Tardos, Tovey, and Trick [81] improved this algorithm to run in $\mathrm{O}(n^4 h)$ time by incorporating the layered augmenting path method of Dinitz [9]. As a generalization of the preflow-push algorithm of Goldberg and Tarjan [40] for the maximum flow problem, Fujishige and Zhang [32] devised a push/relabel algorithm for the submodular intersection problem. This algorithm can be adjusted to solve the maximum submodular flow problem in $\mathrm{O}(n^3 h)$ time. Thus the Cunningham–Frank algorithm can be made to run in $\mathrm{O}(n^4 h \log C)$ time [55], which is currently the best time bound for the submodular flow problem.

Gallo, Grigoriadis, and Tarjan (GGT) [37] showed that the preflow-push algorithm can be extended without increasing the time complexity to solve monotone parametric maximum flow problems. Extensions and applications of the GGT method have been discussed by Gusfield–Martel [45], Gusfield–Tardos [46], McCormick [63], and Fleischer [15]. Iwata, Murota, and Shigeno [56] extended the GGT result to the intersection problem on a pair of strong map sequences of submodular systems [82]. This reveals an algorithmic advantage of the concept of strong maps, which has been investigated mainly from structural point of view [58].

The first strongly polynomial algorithm for solving the submodular flow problem is due to Frank and Tardos [22]. This is an application of simultaneous Diophantine approximation and substantially generalizes the first strongly polynomial minimum cost flow algorithm of Tardos [79]. A more direct generalization of the Tardos algorithm to the submodular flow problem is described by Fujishige, Röck, and Zimmermann [30] with the aid of the tree-projection method of Fujishige [27]. Both of these algorithms call a subroutine to solve submodular flow problems with small integer costs. In particular, the latter algorithm combined with the improved Cunningham–Frank algorithm runs in $\mathrm{O}(n^6 h \log n)$ time, which is currently the best strongly polynomial bound.

Another approach that leads to strongly polynomial bounds comes from cycle/cut canceling. Extending a cycle canceling minimum cost flow algorithms in [78], Iwata, McCormick, and Shigeno [53] presents a cycle canceling algorithm which adopts a lexicographic cycle selection rule within the successive approximation framework. The algorithm runs in $\mathrm{O}(n^4 h \log nC)$ time and its strongly polynomial variant in $\mathrm{O}(n^6 h \log n)$ time. They also show that a cycle canceling minimum cost flow algorithm of Goldberg [38] can also be extended to the submodular flow problem by the same lexicographic technique. This algorithm can be regarded as a successive approximation version of the out-of-kilter method.

In the dual side, Iwata, McCormick, and Shigeno [54] have extended the cut canceling algorithm in [78] to the submodular flow problem. The resulting algorithm runs in $\mathrm{O}(n^6 h \log nU)$ time. Incorporating the maximum-mean submodular cut computation, they have devised a strongly polynomial version which runs in $\mathrm{O}(n^8 h \log n)$ time. The same technique shows that an extension of the maximum-mean cut canceling algorithm of Ervolina and McCormick [13] runs in strongly polynomial time. This contrasts with the primal side, where Cui and Fujishige [7] extended the minimum-mean cycle canceling algorithm of Goldberg and Tarjan [41], but the running time is only pseudopolynomial.

As the weakly polynomial bounds suggest, we may regard the cycle canceling in [53] as cost scaling and the cut canceling algorithm in [54] as capacity scaling. Extending the Edmonds–Karp [12] algorithm for minimum-cost flow, Iwata [51] devised the first capacity scaling algorithm for the submodular flow problem. Since rounding a submodular function in a straightforward way may destroy the submodularity, the capacity scaling approach seemed more difficult than the cost scaling. The algorithm in [51] obviates this difficulty by adding a small but strictly submodular function before rounding. The resulting scaling scheme, however, calls maximum submodular flow computations $\mathrm{O}(n^4 \log U)$ times, and hence the overall running time is $\mathrm{O}(n^7 h \log U)$, which is less attractive than the cost scaling approach. Introducing a variant of the Dijkstra shortest path algorithm modified to deal with exchange capacity arcs, Fleischer, Iwata, and McCormick [16] have improved this algorithm to run in $\mathrm{O}(n^4 h \log U)$ time. With the aid of the same technique as in [54], they also make the algorithm run in $\mathrm{O}(n^6 h \log n)$ time, which matches the best known strongly polynomial bound for the submodular flow problem. This algorithm can be regarded as an extension of a fast cut canceling algorithm in [78].

While the above mentioned cycle/cut canceling al-

**Table 1** Algorithms for minimum cost submodular flows.

| Reference | Technique | Complexity | Year |
|---|---|---|---|
| Fujishige [25] | Successive Shortest Path | | 1978 |
| Fujishige [25] | Cycle Canceling | | 1978 |
| Zimmermann [86] | Cycle Canceling | | 1982 |
| Barahona–Cunningham [5] | Network Simplex | | 1984 |
| Cunningham–Frank [8] | Primal-Dual & Cost Scaling | $\rightarrow \mathrm{O}(n^4 h \log C)$ | 1985 |
| Fujishige [28] | Out-of-Kilter | | 1987 |
| Cui–Fujishige [7] | Min-Mean Cycle Canceling | | 1988 |
| Frank–Tardos [22] | Diophantine Approximation | Strongly Polynomial | 1987 |
| Fujishige–Röck–Zimmermann [30] | Tree-Projection | $\rightarrow \mathrm{O}(n^6 h \log n)$ | 1989 |
| Zimmermann [87] | Min-Ratio Cycle Canceling | | 1992 |
| McCormick–Ervolina [64] | Most Helpful Cut Canceling | $\mathrm{O}(n^7 h^* \log nCU)$ | 1993 |
| Wallacher–Zimmermann [84] | Polynomial Cycle Canceling | $\mathrm{O}(n^8 h \log nCU)$ | 1994 |
| Iwata [51] | Capacity Scaling | $\mathrm{O}(n^7 h \log U)$ | 1997 |
| Iwata–McCormick–Shigeno [53] | Lexicographic Cycle Canceling | $\mathrm{O}(n^4 h \min\{\log nC,\ n^2 \log n\})$ | 1998 |
| Iwata–McCormick–Shigeno [54] | Adjust-Flow Cut Canceling | $\mathrm{O}(n^6 h \min\{\log nU,\ n^2 \log n\})$ | 1999 |
| Fleischer–Iwata–McCormick [16] | Submodular Dijkstra | $\mathrm{O}(n^4 h \min\{\log U,\ n^2 \log n\})$ | 1999 |

**Table 2** Algorithms for maximum submodular flows.

| Reference | Technique | Complexity | Year |
|---|---|---|---|
| Fujishige [25] | Augmenting Path | | 1978 |
| Shönsleben [73] | | | 1980 |
| Lawler–Martel [60] | Lex Shortest Path | $O(n^5 h)$ | 1982 |
| Frank [20] | | | 1984 |
| Tardos–Tovey–Trick [81] | Layered Augmentation | $O(n^4 h)$ | 1986 |
| Fujishige–Zhang [32] | Push/Relabel | $O(n^3 h)$ | 1992 |

**Table 3** Minimum cost flow algorithms yet to be extended to submodular flow problems.

| Algorithm | Reference |
|---|---|
| Most Helpful Cycle Canceling | Barahona–Tardos [6] |
| Min-Mean Cycle Canceling | Goldberg–Tarjan [41] |
| Successive Approximation | Goldberg–Tarjan [42] |
| Double Scaling | Ahuja–Goldberg–Orlin–Tarjan [1] |
| Strongly Polynomial Capacity Scaling | Fujishige [27], Galil–Tardos [36], Orlin [69] |
| Polynomial Dual Network Simplex | Orlin–Plotkin–Tardos [72], Armstrong–Jin [4] |
| Polynomial Primal Network Simplex | Orlin [70] |

gorithms make primal or dual approximate optimality geometrically converge, there are minimum cost flow algorithms of another type that yield geometric convergence of the primal or dual objective value. Those algorithms include the Barahona–Tardos [6] algorithm that cancels the most helpful node disjoint cycles, its dual version [14] that cancels most helpful total cuts, and the minimum-ratio cycle canceling algorithm by Wallacher [83]. McCormick and Ervolina [64] have extended their most helpful total cut canceling algorithm in [14] to the submodular flow problem. Wallacher and Zimmermann [84] have improved a pseudopolynomial minimum-ratio cycle canceling algorithm of Zimmermann [87] to obtain a weakly polynomial time bound $\mathrm{O}(n^8 h \log nCU)$.

These algorithms are fairly slow and not strongly polynomial. Hence these approaches do not seem to find an appropriate place to be applied to the submodular flow problem. However, extending the minimum-ratio cycle canceling algorithm, Wayne [85] has devised the first combinatorial polynomial algorithm for the minimum cost generalized flow problem, which is another interesting generalization of the minimum cost flow problem with many applications.

For the 0-1 submodular flow problem, where lower and upper bounds on each arc are zero and one, Frank [19] presented the first strongly polynomial algorithm. This special case still contains important applications such as the weighted matroid intersection [18], [59] and the directed cut covering [17], [62]. Introducing a novel representation technique for crossing families, Gabow [35] improved the Frank algorithm. In particular, the time complexity for finding a minimum cost directed cut covering has been improved from $\mathrm{O}(mn^3)$ to $\mathrm{O}(mn^2)$. Gabow [34] also presented a successive approximation algorithm for the 0-1 submodular flow problem. For the independent assignment problem [50], which is equivalent to the weighted matroid intersection, Fujishige and Zhang [33] extended an assignment algorithm of Orlin and Ahuja [71] based on approximate optimality. Shigeno and Iwata [77] presented a fairly simple algorithm for the weighted matroid intersection incorporating the approximate optimality with the weight splitting approach of Frank [18].

The history and the state of the art of the algorithms for minimum cost submodular flows and maximum submodular flows are, respectively summarized in Tables 1 and 2.

## 4. Future Research Directions

This section discusses possible directions of research on submodular flow algorithms. A natural question is to ask if we can extend an arbitrary minimum cost flow algorithm to the submodular flow problem. Actually, we have several open problems in this respect. Table 3 provides a list of interesting minimum cost flow algorithms that have not been extented to the submodular flow problem. For maximum submodular flow problems, it will be interesting to extend the excess scaling method of Ahuja–Orlin [3] and the binary blocking flow method of Goldberg–Rao [39].

Another important direction is to further extend submodular flow algorithms to more general models. In this respect, lattice polyhedra of Hoffman–Schwartz [49] and a more general TDI system introduced by Schrijver [75] may be interesting places to look at algorithmically (also see [21], [43], [74]).

In connection with the discrete convex analysis [65], [66], Murota [67] recently introduced the submodular flow problem with M-convex cost functions, for which Iwata–Shigeno [57] have devised a polynomial algorithm based on a new scaling framework.

### References


[1] R.K. Ahuja, A.V. Goldberg, J.B. Orlin, and R.E. Tarjan, "Finding minimum-cost flows by double scaling," Math. Programming, vol.53, pp.243–266, 1992.

[2] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin, Network Flows — Theory, Algorithms, and Applications, Prentice Hall, 1993.

[3] R.K. Ahuja and J.B. Orlin, "A fast and simple algorithm for the maximum flow problem," Oper. Res., vol.37, pp.748–759, 1989.

[4] R.D. Armstrong and Z. Jin, "A new strongly polynomial dual network simplex algorithm," Math. Programming, vol.78, pp.131–148, 1997.

[5] F. Barahona and W.H. Cunningham, "A submodular network simplex method," Math. Programming Stud., vol.22, pp.9–31, 1984.

[6] F. Barahona and É. Tardos, "Note on Weintraub's minimum cost flow algorithm," SIAM J. Comput., vol.18, pp.579–583, 1989.

[7] W. Cui and S. Fujishige, "A primal algorithm for the submodular flow problem with minimum-mean cycle selection," J. Oper. Res. Soc. Japan, vol.31, pp.431–440, 1988.

[8] W.H. Cunningham and A. Frank, "A primal-dual algorithm for submodular flows," Math. Oper. Res., vol.10, pp.251–262, 1985.

[9] E.A. Dinic, "Algorithm for solution of a problem of maximum flow in a network with power estimation," Soviet Math. Dokl., vol.11, pp.1277–1280, 1970.

[10] J. Edmonds, "Submodular functions, matroids, and certain polyhedra," in Combinatorial Structures and Their Applications, eds. R. Guy, H. Hanani, N. Sauer and J. Schönheim, pp.69–87, Gordon and Breach, 1970.

[11] J. Edmonds and R. Giles, "A min-max relation for submodular functions on graphs," Ann. Discrete Math., vol.1, pp.185–204, 1977.

[12] J. Edmonds and R.M. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems," J. ACM, vol.19, pp.248–264, 1972.

[13] T.R. Ervolina and S.T. McCormick, "Two strongly polynomial cut canceling algorithms for minimum cost network flow," Discrete Appl. Math., vol.46, pp.133–165, 1993.

[14] T.R. Ervolina and S.T. McCormick, "Canceling most helpful total cuts for minimum cost network flow," Networks, vol.23, pp.41–52, 1993.

[15] L. Fleischer, "Universally maximum flow with piecewise-constant capacities," Proc. 7th MPS Conference on Integer Programming and Combinatorial Optimization, pp.151–165, Springer-Verlag, 1998.

[16] L. Fleischer, S. Iwata, and S.T. McCormick, "A faster capacity scaling algorithm for minimum cost submodular flow," Discrete Mathematics and Systems Science Research Report, 99-08, Osaka University, 1999.

[17] A. Frank, "How to make a digraph strongly connected," Combinatorica, vol.1, pp.145–153, 1981.

[18] A. Frank, "A weighted matroid intersection algorithm," J. Algorithms, vol.2, pp.328–336, 1981.

[19] A. Frank, "An algorithm for submodular functions on graphs," Ann. Discrete Math., vol.16, pp.97–120, 1982.

[20] A. Frank, "Finding feasible vectors of Edmonds–Giles polyhedra," J. Combin. Theory, Ser. B, vol.36, pp.221–239, 1984.

[21] A. Frank, "Increasing the rooted-connectivity of a digraph by one," Math. Programming, vol.84, pp.565–576, 1999.

[22] A. Frank and É. Tardos, "An application of simultaneous Diophantine approximation in combinatorial optimization," Combinatorica, vol.7, pp.49–65, 1987.

[23] A. Frank and É. Tardos, "An application of submodular flows," Linear Algebra Appl., vol.114, pp.329–348, 1988.

[24] A. Frank and É. Tardos, "Generalized polymatroids and submodular flows," Math. Programming, vol.42, pp.489–563, 1988.

[25] S. Fujishige, "Algorithms for solving the independent-flow problems," J. Oper. Res. Soc. Japan, vol.21, pp.189–204, 1978.

[26] S. Fujishige, "Structures of polyhedra determined by submodular functions on crossing families," Math. Programming, vol.29, pp.125–141, 1984.

[27] S. Fujishige, "A capacity-rounding algorithm for the minimum cost circulation problem – A dual framework of the Tardos algorithm," Math. Programming, vol.35, pp.298–308, 1986.

[28] S. Fujishige, "An out-of-kilter method for submodular flows," Discrete Appl. Math., vol.17, pp.3–16, 1987.

[29] S. Fujishige, Submodular Functions and Optimization, North-Holland, 1991.

[30] S. Fujishige, H. Röck, and U. Zimmermann, "A strongly polynomial algorithm for minimum cost submodular flow problems," Math. Oper. Res, vol.14, pp.60–69, 1989.

[31] S. Fujishige and N. Tomizawa, "A note on submodular functions on distributive lattices," J. Oper. Res. Soc. Japan, vol.26, pp.309–318, 1983.

[32] S. Fujishige and X. Zhang, "New algorithms for the inter-

section problem of submodular systems," Japan. J. Indust. Appl. Math., vol.9, pp.369–382, 1992.

[33] S. Fujishige and X. Zhang, "An efficient cost scaling algorithm for the independent assignment problem," J. Oper. Res. Soc. Japan, vol.38, pp.124–136, 1995.

[34] H. Gabow, "A framework for cost-scaling algorithms for submodular flow problems," Proc. 34th IEEE Annual Symposium on Foundations of Computer Science, pp.449–458, 1993.

[35] H. Gabow, "Centroids, representations, and submodular flows," J. Algorithms, vol.18, pp.586–628, 1995.

[36] Z. Galil and É. Tardos, "An $\mathrm{O}(n^2(m + n \log n) \log n)$ min-cost flow algorithm," J. ACM, vol.35, pp.374–386, 1988.

[37] G. Gallo, M.D. Grigoriadis, and R.E. Tarjan, "A fast parametric network flow algorithm and applications," SIAM J. Comput., vol.18, pp.30–55, 1989.

[38] A.V. Goldberg, "Scaling algorithms for the shortest paths problem," SIAM J. Comput., vol.24, pp.494–504, 1995.

[39] A.V. Goldberg and S. Rao, "Beyond the flow decomposition barrier," Proc. 38th Annual Symposium on Foundations of Computer Science, pp.2–11, 1997.

[40] A.V. Goldberg and R.E. Tarjan, "A new approach to the maximum flow problem," J. ACM, vol.35, pp.921–940, 1988.

[41] A.V. Goldberg and R.E. Tarjan, "Finding minimum-cost circulations by canceling negative cycles," J. ACM, vol.36, pp.873–886, 1989.

[42] A.V. Goldberg and R.E. Tarjan, "Finding minimum-cost circulations by successive approximation," Math. Oper. Res., vol.15, pp.430–466, 1990.

[43] H. Gröflin and A.J. Hoffman, "Lattice polyhedra II – Generalizations, constructions and examples," Ann. Discrete Math., vol.15, pp.189–203, 1982.

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

[45] D. Gusfield and C.U. Martel, "A fast algorithm for the generalized parametric minimum cut problem and applications," Algorithmica, vol.7, pp.499–519, 1992.

[46] D. Gusfield and É. Tardos, "A faster parametric minimum-cut algorithm," Algorithmica, vol.11, pp.278–290, 1994.

[47] R. Hassin, "Minimum cost flow with set-constraints," Networks, vol.12, pp.1–21, 1982.

[48] A.J. Hoffman, "A generalization of max-flow min-cut," Math. Programming, vol.6, pp.352–359, 1974.

[49] A.J. Hoffman and D. Schwartz, "On lattice polyhedra," in Combinatorics, eds. A. Hajnal and V.T. Sós, North-Holland, pp.593–598, 1978.

[50] M. Iri and N. Tomizawa, "An algorithm for finding an optimal "independent assignment"," J. Oper. Res. Soc. Japan, vol.19, pp.32–57, 1976.

[51] S. Iwata, "A capacity scaling algorithm for convex cost submodular flows," Math. Programming, vol.76, pp.299–308, 1997.

[52] S. Iwata, L. Fleischer, and S. Fujishige, "A combinatorial, strongly polynomial-time algorithm for minimizing submodular functions," Proc. 32nd ACM Symposium on Theory of Computing, to appear.

[53] S. Iwata, S.T. McCormick, and M. Shigeno, "A faster algorithm for minimum cost submodular flows," Proc. 9th ACM/SIAM Symposium on Discrete Algorithms, pp.167–174, 1998.

[54] S. Iwata, S.T. McCormick, and M. Shigeno, "A strongly polynomial cut canceling algorithm for the submodular flow problem," Proc. 7th MPS Conference on Integer Programming and Combinatorial Optimization, pp.259–272, Springer-Verlag, 1999.

[55] S. Iwata, S.T. McCormick, and M. Shigeno, "A fast cost scaling algorithm for submodular flow," Discrete Mathematics and Systems Science Research Report, vol.99-09, Osaka University, 1999.

[56] S. Iwata, K. Murota, and M. Shigeno, "A fast parametric submodular intersection algorithm for strong map sequences," Math. Oper. Res., vol.22, pp.803–813, 1997.

[57] S. Iwata and M. Shigeno, "Conjugate scaling technique for Fenchel-type duality in discrete convex optimization," Discrete Mathematics and Systems Science Research Report, vol.99-02, Osaka University, 1999.

[58] J. Kung, "Strong maps," in Theory of Matroids, ed. N. White, Cambridge Univ. Press, pp.224–253, 1986.

[59] E.L. Lawler, "Matroid intersection algorithms," Math. Programming, vol.9, pp.31–56, 1975.

[60] E.L. Lawler and C.U. Martel, "Computing maximal polymatroidal network flows," Math. Oper. Res., vol.7, pp.334–347, 1982.

[61] L. Lovász, "Submodular functions and convexity," in Mathematical Programming – The State of the Art, eds. A. Bachem, M. Grötschel and B. Korte, pp.235–257, Springer-Verlag, 1983.

[62] C. Lucchesi and D.H. Younger, "A minimax relation for directed graphs," J. London Math. Soc., vol.17, pp.369–374, 1978.

[63] S.T. McCormick, "Fast algorithms for parametric scheduling come from extensions to parametric maximum flow," Proc. 28th ACM Symposium on Theory of Computing, pp.319–328, 1996.

[64] S.T. McCormick and T.R. Ervolina, "Canceling most helpful total submodular cuts for submodular flow," Proc. 3rd MPS Conference on Integer Programming and Combinatorial Optimization, pp.343–353, 1993.

[65] K. Murota, "Convexity and Steinitz's exchange property," Adv. Math., vol.124, pp.272–311, 1996.

[66] K. Murota, "Discrete convex analysis," Math. Programming, vol.83, pp.313–371, 1998.

[67] K. Murota, "Submodular flow problem with a nonseparable cost function," Combinatorica, vol.19, pp.87–109, 1999.

[68] T. Naitoh and S. Fujishige, "A note on the Frank–Tardos bitruncation algorithm for crossing-submodular functions," Math. Programming, vol.53, pp.361–363, 1992.

[69] J.B. Orlin, "A faster strongly polynomial minimum cost flow algorithm," Oper. Res., vol.41, pp.338–350, 1993.

[70] J.B. Orlin, "A polynomial time primal network simplex algorithm for minimum cost flows," Math. Programming, vol.78, pp.109–129, 1997.

[71] J.B. Orlin and R.K. Ahuja, "New scaling algorithms for the assignment and minimum mean cycle problems," Math. Programming, vol.54, pp.41–56, 1992.

[72] J.B. Orlin, S.A. Plotkin, and É. Tardos, "Polynomial dual network simplex algorithms," Math. Programming, vol.60, pp.255–276, 1993.

[73] P. Schönsleben, Ganzzahlige Polymatroid Intersektions Algorithmen, Dissertation, ETH Zürich, 1980.

[74] A. Schrijver, "Min-max relations for directed graphs," Ann. Discrete Math., vol.16, pp.127–146, 1982.

[75] A. Schrijver, "Proving total dual integrality with cross-free families – A general framework," Math. Programming, vol.29, pp.15–27, 1984.

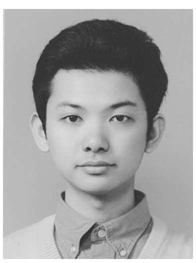
[76] A. Schrijver, "A combinatorial algorithm minimizing submodular functions in strongly polynomial time," J. Combin. Theory, Ser. B, submitted.

[77] M. Shigeno and S. Iwata, "A dual approximation approach to weighted matroid intersection," Oper. Res. Lett., vol.18, pp.153–156, 1995.

[78] 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., to appear.
[79] É. Tardos, "A strongly polynomial minimum cost circulation algorithm," Combinatorica, vol.5, pp.247–256, 1985.
[80] É. Tardos, "A strongly polynomial algorithm to solve combinatorial linear programs," Oper. Res., vol.34, pp.250–256, 1986.
[81] É. Tardos, C.A. Tovey, and M.A. Trick, "Layered augmenting path algorithms," Math. Oper. Res., vol.11, pp.362–370, 1986.
[82] D.M. Topkis, "Minimizing a submodular function on a lattice," Oper. Res., vol.26, pp.305–321, 1978.
[83] C. Wallacher, "A generalization of the minimum-mean cycle selection rule in cycle canceling algorithms," Report, Inst. Ang. Math., Braunschweig, 1991.
[84] C. Wallacher and U. Zimmermann, "A polynomial cycle canceling algorithm for submodular flows," Math. Programming, vol.86, pp.1–15, 1999.
[85] K. Wayne, "A polynomial combinatorial algorithm for generalized minimum cost flow," Proc. 31st Annual ACM Symposium on Theory of Computing, pp.11–18, 1999.
[86] U. Zimmermann, "Minimization on submodular flows," Discrete Appl. Math., vol.4, pp.303–323, 1982.
[87] U. Zimmermann, "Negative circuits for flows and submodular flows," Discrete Appl. Math., vol.36, pp.179–189, 1992.

**Satoru Fujishige** received B.E., M.E., and Ph.D. degrees in Applied Mathematics and Physics from Kyoto University in 1970, 1972, and 1975, respectively. He was a Research Associate (1975–1976) and an Assistant Professor (1976–1979) at University of Tokyo, and an Associate Professor (1979–1988) and a Professor (1988–1997) at University of Tsukuba. Since 1997 he has been a Professor at Osaka University. His major research interests are in combinatorial optimization and algorithms. He is the author of several books including "Submodular Functions and Optimization" (North-Holland, 1991).



**Satoru Iwata** received M.E. from University of Tokyo in 1993 and Ph.D. from Kyoto University in 1996. He is an Associate Professor at Osaka University. His major research interests are in discrete optimization and systems analysis.