Node-Connectivity Terminal Backup, Separately-Capacitated Multiflow, and Discrete Convexity

Hiroshi Hirai
Department of Mathematical Informatics,
Graduate School of Information Science and Technology,
The University of Tokyo, Japan
hirai@mist.i.u-tokyo.ac.jp

Motoki Ikeda
Department of Mathematical Informatics,
Graduate School of Information Science and Technology,
The University of Tokyo, Japan
motoki_ikeda@mist.i.u-tokyo.ac.jp

Abstract
The terminal backup problems [Anshelevich and Karagiozova, 2011] form a class of network design problems: Given an undirected graph with a requirement on terminals, the goal is to find a minimum cost subgraph satisfying the connectivity requirement. The node-connectivity terminal backup problem requires a terminal to connect other terminals with a number of node-disjoint paths. This problem is not known whether is NP-hard or tractable. Fukunaga (2016) gave a $4/3$-approximation algorithm based on LP-rounding scheme using a general LP-solver.

In this paper, we develop a combinatorial algorithm for the relaxed LP to find a half-integral optimal solution in $O(m \log(mUA) \cdot MF(kn, m + k^2n))$ time, where $m$ is the number of edges, $k$ is the number of terminals, $A$ is the maximum edge-cost, $U$ is the maximum edge-capacity, and $MF(n', m')$ is the time complexity of a max-flow algorithm in a network with $n'$ nodes and $m'$ edges. The algorithm implies that the $4/3$-approximation algorithm for the node-connectivity terminal backup problem is also efficiently implemented. For the design of algorithm, we explore a connection between the node-connectivity terminal backup problem and a new type of a multiflow, called a separately-capacitated multiflow. We show a min-max theorem which extends Lovász-Cherkassky theorem to the node-capacity setting. Our results build on discrete convex analysis for the node-connectivity terminal backup problem.

2012 ACM Subject Classification Mathematics of computing → Combinatorial optimization

Keywords and phrases terminal backup problem, node-connectivity, separately-capacitated multiflow, discrete convex analysis

Digital Object Identifier 10.4230/LIPIcs.ICALP.2020.65

Category Track A: Algorithms, Complexity and Games

Funding Hiroshi Hirai: JSPS KAKENHI Grant Number JP17K00029 and JST PRESTO Grant Number JPMJPR192A, Japan.

Acknowledgements We thank the referees for helpful comments.

1 Introduction

Network design problems are central problems in combinatorial optimization. A large number of basic combinatorial optimization problems are network design problems. Examples are spanning tree, matching, TSP, and Steiner networks. They admit a typical formulation of a network design problem: Find a minimum-cost network satisfying given connectivity requirements. The present paper addresses a relatively new class of network design problems,
65:2 Terminal Backup, Multiflow, and Discrete Convexity

called terminal backup problems. The problem is to find a cheapest subnetwork in which each terminal can send a specified amount of flows to other terminals, i.e., the data in each terminal can be backed up, possibly in a distributed manner, in other terminals.

A mathematical formulation of the terminal backup problem is given as follows. Let \((V, E), S, u, c, a, r\) be an undirected network, where \((V, E)\) is a simple undirected graph, \(S \subseteq V (|S| \geq 3)\) is a set of terminals, \(u : E \rightarrow \mathbb{Z}_+\) is a nonnegative edge-capacity function, \(c : V \setminus S \rightarrow \mathbb{Z}_+\) is a nonnegative node-capacity function, \(a : E \rightarrow \mathbb{Z}_+\) is a nonnegative edge-cost function, and \(r : S \rightarrow \mathbb{Z}_+\) is a nonnegative requirement function on terminals. The goal is to find a feasible edge-capacity function \(x\) of minimum cost \(\sum_{e \in E} a(e)x(e)\). Here an edge-capacity function \(x\) is said to be feasible if \(0 \leq x \leq u\) and each terminal \(s \in S\) has a flow from \(s\) to \(S \setminus \{s\}\), an \(\{s\}\)–\((S \setminus \{s\})\) flow, of total flow-value \(r(s)\) in the network \((V, E), S, x, c)\) capacitated by the edge-capacity \(x\) and the node-capacity \(c\).

The original formulation, due to Anshelevich and Karagiozova [1], is uncapacitated (i.e., \(u, c\) are infinity), requires \(x\) to be integer-valued, and assumes \(r(s) = 1\) for all \(s \in S\). They showed that an optimal solution can be obtained in polynomial time. Bernáth et al. [2] extended this polynomial time solvability to an arbitrary integer-valued requirement \(r\). For the setting of general edge-capacity (and infinite node-capacity), which we call the edge-connectivity terminal backup problem (ETB), it is unknown whether ETB is NP-hard or tractable.

Fukunaga [8] considered the above setting including both edge-capacity and node-capacity, which we call the node-connectivity terminal backup problem (NTB), and explored intriguing features of its fractional relaxation. The fractional ETB (FETB) and fractional NTB (FNTB) are LP-relaxations obtained from ETB and NTB, respectively, by relaxing solution \(x\) to be real-valued. Fukunaga showed the half-integrality property of FNTB, that is, there always exists an optimal solution that is half-integer-valued. Based on this property, he developed a \(4/3\)-approximation algorithm for NTB by rounding a half-integral (extreme) optimal solution. Moreover, he noticed a useful relationship between FETB and multiflow (multiflow). In fact, a solution of FETB is precisely the edge-support of a multiflow consisting of the \(r(s)\) amount of \(\{s\}\)–\((S \setminus \{s\})\) flow for each \(s \in S\). This is a consequence of Lovász–Cherkassky theorem [5, 21] in multiflow theory. In particular, FETB is equivalent to a minimum-cost multiflow problem, which is a variant of the one studied by Karzanov [19, 20] and Goldberg and Karzanov [10].

Utilizing this connection, Hirai [12] developed a combinatorial polynomial time algorithm for FETB and the corresponding multiflow problem. This algorithm uses a max-flow algorithm as a subroutine, and brings a combinatorial implementation of Fukunaga’s \(4/3\)-approximation algorithm for ETB, where he used a generic LP-solver (e.g., the ellipsoid method) to obtain a half-integral extreme optimal solution.

Our first contribution is the extension of this result to the NTB setting, implying that the \(4/3\)-approximation algorithm for NTB is also efficiently implemented.

**Theorem 1.** A half-integral optimal solution of FNTB can be obtained in \(O(m \log(mUA) \cdot MF(kn, m + k^2n))\) time.

Here \(n := |V|, m := |E|, k := |S|, U := \max_{e \in E} u(e), A := \max_{e \in E} a(e),\) and \(MF(n', m')\) is the time complexity of an algorithm for solving the max-flow problem in the network with \(n'\) nodes and \(m'\) edges.

As in the ETB case, we explore and utilize a new connection between NTB and a multiflow problem. We introduce a new notion of a free multiflow with separate node-capacity constraints or simply a separately-capacitated multiflow. Instead of the usual node-capacity...
constraints, this multiflow should satisfy the separate node-capacity constraints: For each
terminal \( s \in S \) and each node \( i \in V \), the total flow-value of flows connecting \( s \) to the other
terminals and flowing into \( i \) is at most the node capacity \( c(i) \).

Our second contribution is a min-max theorem for separately-capacitated multiflows,
which extends Lovász–Cherkassky theorem to the node-capacitated setting and implies that
a solution of FNTB is precisely the edge-support of a separately-capacitated multiflow. This
answers Fukunaga’s comment: how the computation should proceed in the node capacitated
setting remains elusive [8, p. 799].

\[\textbf{Theorem 2.} \text{The maximum flow-value of a separately-capacitated multiflow is}\]
\[\text{equal to} \quad (1/2) \sum_{s \in S} \mu_s, \quad \text{where} \quad \mu_s \quad \text{is the minimum capacity of an}\]
\[\text{cut. Moreover, a half-integral maximum multiflow exists, and it can be found in} \quad O(n \cdot MF(kn, m + k^2 n)) \text{ time.}\]

Here, a \( T - T' \) cut is a union of an edge-subset \( F \subseteq E \) and a node-subset \( X \subseteq V \setminus (T \cup T') \) such that
removing those subsets disconnects \( T \) and \( T' \), and its capacity is defined as \( u(F) + c(X) \).

Our algorithm for Theorem 1 builds on the ideas of \textit{Discrete Convex Analysis (DCA)}
beyond \( \mathbb{Z}^n \) – a theory of discrete convex functions on special graph structures generalizing
\( \mathbb{Z}^n \) (the grid graph), which has been recently differentiated from the original DCA [23] and
has been successfully applied to algorithm design for well-behaved classes of multiflow and
related network design problems [12, 13, 14, 16]. Indeed, the algorithm in [12] for FETB was
designed as: Formulate the dual of FETB as a minimization of an \( L \)-convex function
on the (Cartesian) product of trees, apply the framework of the \textit{steepest descent algorithm (SDA)},
and show that it is implemented by using a max-flow algorithm as a subroutine.

We formulate the dual of FNTB as an optimization problem on the product of the spaces
of all subtrees of a fixed tree (Section 2.1). We develop a simple cut-descent algorithm for
this optimization problem (Sections 2.2 and 2.3). Then we prove that this coincides with
SDA for an \( L \)-convex function defined on the graph structure on the space of all subtrees
(Section 3). Then the number of descents is estimated by a general theory of SDA, and the
cost-scaling method is naturally incorporated to derive the time complexity (Section 2.4).

Theorem 2 is obtained as a byproduct of these arguments. Due to the space limitation, we
omit most of technical proofs, which are given in the full version.

\[\text{Related work}\]

ETB is a \textit{survivable network design problem (SND)} with a special skew-supermodular function,
and NTB is a node connectivity version (NSND) with a special skew-supermodular biset
function. In his influential paper [18], Jain devised the iterative rounding method, and obtains
a 2-approximation algorithm for SND, provided that an extreme optimal solution of the LP-
relaxation of SND (with modified skew-supermodular functions) is available. Fleischer, Jain,
and Williamson [7] and Cheriyan, Vempala, and Vetta [4] extended this iterative rounding
2-approximation algorithm to some classes of NSND. One of important open problems in
the literature is a design of a combinatorial 2-approximation algorithm for (V)SND with the
skew-supermodular (biset) function associated with connectivity requirements. One approach
is to devise a combinatorial polynomial time algorithm to find an extreme optimal solution of
its LP-relaxation; the currently known only polynomial time algorithm is a general LP-solver
(e.g., the ellipsoid method). Our algorithm for FNTB, though it is the LP-relaxation of a very
special NSND, may give an insight on such a research direction. On this direction, Feldmann,
Könemann, Pashkovich, and Sanità [6] gave a \((2 + \epsilon)\)-approximation algorithm for SND with
a proper function by solving the LP-relaxation approximately via the multiplicative weights
method [9].
The notion of a separately-capacitated multiflow, introduced in this paper, is a new variation of S-paths packing. As seen in [24, Chapter 73], S-paths packing is one of the well-studied subjects in combinatorial optimization. Recent work [17] developed a fast algorithm for half-integral nonzero S-paths packing problem on a group-valued graph (with unit-capacity). Our derivation of Theorem 2 is different with flow-augmenting arguments such as Cherkassky’s T-operation or those in [17]. It is a future research to develop such an algorithm for a separately-capacitated multiflow. Also, exploring an integer version of Theorem 2, an analogue of Mader’s theorem [22], is an interesting future direction.

Notations
Let \( \mathbb{Z}, \mathbb{Z}_+, \mathbb{R}, \mathbb{R}_+ \) be the set of integers, nonnegative integers, reals, and nonnegative reals, respectively. Let \( \mathbb{Z}^+, \mathbb{Z}_+^* \) be the set of half-integers and nonnegative half-integers, respectively, i.e., \( \mathbb{Z}^+ := \mathbb{Z}/2 \). Let \( \overline{\mathbb{R}} := \mathbb{R} \cup \{+\infty\} \) and \( \underline{\mathbb{R}} := \mathbb{R} \cup \{-\infty\} \). Let denote \( (a)^+ := \max\{a, 0\} \) for \( a \in \mathbb{R} \). For a finite set \( V \), we often identify a function on \( V \) with a vector in \( \mathbb{R}^V \). For \( i \in V \), its characteristic function \( \chi_i : V \to \mathbb{R} \) is defined by \( \chi_i(j) = 1 \) if \( j = i \) and \( \chi_i(j) = 0 \) otherwise. For a function \( f \) on \( V \) and a subset \( U \subseteq V \), we denote \( f(U) := \sum_{i \in U} f(i) \).

In this paper, all graphs are simple and connected unless otherwise specified. For an undirected graph on nodes \( V \), the set of edges connecting \( U_1 \) and \( U_2 \) (\( U_1, U_2 \subseteq V \)) is denoted by \( \delta(U_1, U_2) \). If \( U_2 = V \setminus U_1 \), we simply denote it by \( \delta U_1 \). If \( U_1 \) is a singleton, i.e., \( U_1 = \{i\} \), then we denote \( \delta \{i\} \) by \( \delta i \). An edge connecting \( i \) and \( j \) is denoted by \( ij \).

2 Node-Connectivity Terminal Backup Problem

Let \( ((V, E), S, u, c, a, r) \) be a network. Assume that \( S = \{1, \ldots, k\} \subseteq V = \{1, \ldots, n\} \). By a perturbation technique, we may assume that \( a \) is positive; see Remark 3.

A sufficient and necessity condition for the feasibility of NTB is easily derived from the Menger’s theorem as follows. A biset is a pair of node subsets \( X, X^+ \subseteq V \) with \( X \subseteq X^+ \). We write \( \hat{X} = (X, X^+) \) for a biset. Let \( \Gamma(\hat{X}) := X^+ \setminus X \), and let \( \delta(\hat{X}) := \delta(X, V \setminus X^+) \). For \( s \in S \), define a family \( C_s \) of bisets by
\[
C_s := \{(X, X^+) \mid \{s\} \subseteq X \subseteq X^+ \subseteq V \setminus (S \setminus \{s\})\}.
\]
Let \( C := \bigcup_{s \in S} C_s \). Then an edge-capacity \( x : E \to \mathbb{Z}_+ \) is feasible if and only if
\[
x(\delta(\hat{X})) + c(\Gamma(\hat{X})) \geq r(s) \quad (\hat{X} \in C_s, s \in S).
\]
(1)
We assume that \( u \) satisfies (1) throughout the paper (otherwise NTB is infeasible).

Fukunaga [8] developed an approximation algorithm for NTB via the following relaxation problem FNTB:

\[
\text{(FNTB)} \quad \text{Minimize} \quad \sum_{e \in E} a(e)x(e)
\]
subject to
\[
x(\delta(\hat{X})) + c(\Gamma(\hat{X})) \geq r(s) \quad (s \in S, \hat{X} \in C_s),
\]
\[
0 \leq x(e) \leq u(e) \quad (e \in E).
\]
From the assumption, the polytope defined by (2) and (3) is nonempty. Also, it is known [8, Corollary 3.3] that the polytope is half-integral. Thus FNTB has a half-integral optimal solution. This can be obtained by a general LP solver [8, Lemma 4.4].
Remark 3. If \( Z := \{ e \in E \mid a(e) = 0 \} \) is nonempty, then we use the following perturbation technique based on \([10, 20]\). Recall that \( U \) is the maximum edge capacity. Define a positive edge-cost \( a' \) by \( a'(e) := 1 \) for \( e \in Z \) and \( a'(e) := (2U|Z| + 1)a(e) \) for \( e \notin Z \). Let \( x^* \) be a half-integral optimal solution of FNTB under the edge-cost \( a' \) (it exists by the half-integrality).

We prove that \( x^* \) is also optimal under the original edge-cost \( a \). It suffices to show that \( \sum_{e \in E} a(e)x^*(e) \leq \sum_{e \in E} a(e)x(e) \) for any feasible half-integral edge-capacity \( x \). It holds that \( (2U|Z| + 1)(\sum_{e \in E} a(e)x^*(e) - \sum_{e \in E} a(e)x(e)) = \sum_{e \in E} a'(e)x^*(e) - \sum_{e \in E} a'(e)x(e) - x^*(Z) + x(Z) \leq U|Z| \) and thus \( \sum_{e \in E} a(e)x^*(e) - \sum_{e \in E} a(e)x(e) \leq U|Z|/(2U|Z| + 1) < 1/2 \). By the half-integrality, we obtain \( \sum_{e \in E} a(e)x^*(e) - \sum_{e \in E} a(e)x(e) \leq 0 \).

2.1 Combinatorial Duality for FNTB

We introduce a combinatorial duality theory for FNTB. For each \( s \in S \), consider an infinite path graph \( P_s \) with one endpoint. Glue those \( k (= |S|) \) endpoints, and denote the resulting graph by \( T \). We denote the set of nodes of \( P_s \) and \( T \) also by \( P_s \) and \( T \), respectively.

We give length 1/2 for each edge in \( T \). The glued endpoint is denoted by \( 0 \), and the point in \( P_s \) \((s \in S)\) having the distance \( l \) from \( 0 \) is denoted by \((l, s)\). We denote the set of all subtrees of \( T \) by \( \mathcal{S} = \mathcal{S}(T) \). If a subtree \( T \) does not contain \( 0 \), then it is contained in some \( P_s \). Such a subtree \( T \) is said to be of \( s \)-type and is denoted by \([l, l']_s \), where \((l, s)\) and \((l', s)\) are the closest and farthest nodes from \( 0 \) in \( T \), respectively. If a subtree \( T \) contains \( 0 \), then it is said to be of \( 0 \)-type and is denoted by \([l, l, \ldots, l]_s = [l_s]_s \), where \((l_s, s)\) is the node in \( T \cap P_s \) farthest from \( 0 \) for each \( s \in S \). We identify a node on \( T \) with a subtree consisting of this node only.

For a \( 0 \)-type subtre \( T = [l_s]_s \in \mathcal{S} \), let \( \text{size}_s(T) := l_s \) for \( s \in S \), and \( \text{size}(T) := \sum_{s=1}^k \text{size}_s(T) \). For an \( s \)-type subtree \( T = [l, l']_s \in \mathcal{S} \), let \( \text{size}(T) := l' - l \). For two subtrees \( T, T' \in \mathcal{S} \), we denote the minimum distance between \( T \) and \( T' \) on \( T \) by \( \text{dist}(T, T') \), i.e., \( \text{dist}(T, T') := \min \{ d_T(v, v') \mid v \in T, v' \in T' \} \), where \( d_T \) is the shortest distance on \( T \).

We formulate a dual of FNTB as a problem of assigning a subtree for each node \( i \in V \). That is, subtrees are viewed as node-potentials. So we use \( p_i \) and \( p : V \rightarrow \mathcal{S} \) for denoting a subtree assigned for node \( i \in V \) and a potential function, respectively. Formally, let us consider the following maximization problem D TB.

\[
\text{(DTB)} \quad \text{Maximize } \sum_{s \in S} r_s \text{dist}(0, p_s) - \sum_{i \in V \setminus S} c_i \text{size}(p_i) - \sum_{i,j \in E} u_{ij}(\text{dist}(p_i, p_j) - a_{ij})^+ \\
\text{subject to } p : V \rightarrow \mathcal{S}, \quad p_s \in P_s \quad (s \in S). \tag{4}
\]

It turns out in the proof of Proposition 4 below that this seemingly strange formulation of D TB is essentially the LP-dual of FTB. If \( p : V \rightarrow \mathcal{S} \) satisfies \((4)\), then it is called a potential. See Figure 1 for an intuition for a subtree-valued potential \( p \). A potential \( p \) is said to be proper if any \( p_i \) for \( i \in V \) is contained in the minimal subtree that contains all \( p_s \) \((s \in S)\).

\[\text{Proposition 4. The optimum value of FNTB is at least that of D TB. Moreover, there exists a proper optimal potential for D TB.}\]

\[\text{Proof. Let } p : V \rightarrow \mathcal{S} \text{ be any potential (not necessarily proper). For each } s \in S, \text{ suppose that } p_s \text{ is written as } p_s = (M_s, s) \text{ for } M_s \in \mathbb{Z}_+^k. \text{ Define a new proper potential } p' : V \rightarrow \mathcal{S} \text{ by } p'_s := \left\{ \begin{array}{ll} \{\min\{l, M_s\}, \min\{l', M_s\}\} & \text{if } p_i = [l, l']_s, \\
\{\min\{l_i, M_1\}, \ldots, \min\{l_k, M_k\}\} & \text{if } p_i = [l_1, \ldots, l_k]. \end{array} \right. \]

Then the objective function value of \( p' \) does not decrease. This implies the latter part of the statement.
We next show the former part, i.e., the weak duality. The LP dual of FNTB is written as

\[
\begin{array}{l}
\text{Maximize} & \sum_{s \in S} \sum_{X \in C_s} (r_s - c(\Gamma(X))) \pi(X) - \sum_{e \in E} u_e \left( \sum_{X \in C: e \in \delta X} \pi(X) - a_e \right) \\
\text{subject to} & \pi: C \to \mathbb{R}^+.
\end{array}
\]

We show that for any proper potential \( p: V \to S \), we can construct \( \pi: C \to \mathbb{R}^+ \) such that

\[
\begin{align*}
\sum_{X \in C_s} \pi(X) &= \text{dist}(0, p_s) \quad (s \in S), \\
\sum_{\hat{X} \in C: i \in \Gamma(\hat{X})} \pi(\hat{X}) &= \text{size}(p_i) \quad (i \in V \setminus S), \\
\sum_{\hat{X} \in C: e \in \delta \hat{X}} \pi(\hat{X}) &= \text{dist}(p_i, p_j) \quad (ij \in E).
\end{align*}
\]

Then by \( \sum_{X \in C} c(\Gamma(X)) \pi(X) = \sum_{\hat{X} \in C} \sum_{i \in \Gamma(\hat{X})} c_i \pi(\hat{X}) = \sum_{i \in V \setminus S} c_i \sum_{X \in C: i \in \Gamma(X)} \pi(\hat{X}) \), the weak duality follows.

Let \( e \) be an edge in \( T \). We define a biset \( (X_e, X_e^+) \) as follows. When we remove \( e \) from \( T \), there appear two connected components. Let \( T_e \) be the component which does not contain \( 0 (\in T) \). Define \( X_e, X_e^+ \subseteq V \) by

\[
X_e := \{ i \in V \mid p_i \text{ is contained in } T_e \}, \quad X_e^+ := X_e \cup \{ i \in V \mid p_i \text{ contains } e \}.
\]

Observe that if \( e \) is an edge in \( P_s \) and \( X_e \neq \emptyset \), then \( (X_e, X_e^+) \in C_s \). Then a potential function \( \pi: C \to \mathbb{R}^+ \) defined by

\[
\pi(\hat{X}) := \frac{1}{2} |\{ e \mid \hat{X} = (X_e, X_e^+) \}| \quad (\hat{X} \in C)
\]

satisfies (5)–(7).

We remark that the technique used in the above proof is based on a tree representation of a laminar biset family; see also [11] for the relating argument that maps to each node a subtree as a potential. We also note that our algorithm below will give an algorithmic proof of the strong duality.

Figure 1 A subtree-valued potential \( p \).
We next derive from Proposition 4 the complementary slackness condition. Let \( p : V \to \mathbb{S} \) be a proper potential. By \( p \), we decompose \( V \) into \( S \cup V_0 \cup \bigcup_{s \in S} V_s \), where
\[
V_0 := \{ i \in V \setminus S \mid p_i \text{ is of 0-type} \}, \\
V_s := \{ i \in V \setminus S \mid p_i \text{ is of s-type} \} \quad (s \in S).
\]
In the next lemma, we see that it is sufficient to only consider edges \( ij \in E \) with \( \text{dist}(p_i, p_j) \geq a_{ij} \). Let denote the set of such edges by
\[
E^* := \{ ij \in E \mid \text{dist}(p_i, p_j) \geq a_{ij} \}.
\]
For each \( s \in S \), we denote a set of edges in \( E^* \) connecting \( i \) and \( V_s \) by
\[
E^{i,s} := \{ ij \in E^* \mid j \in V_s \} \quad (i \in V_0, \ s \in S).
\]
By the positivity of \( a \), we see that \( \{E^{i,1}, E^{i,2}, \ldots, E^{i,k}\} \) is a partition of \( E^* \cap \delta i \). For each \( i \in V_0 \) (\( s \in S \)), there appear two connected components when we remove \( p_i \) from \( \mathbb{T} \). Let \( T_{i,0} \) be the component which includes \( 0 (\in \mathbb{T}) \), and let \( T_{i,+} \) be the other component. Then we define the sets of edges \( E^{i,0} \) and \( E^{i,+} \) by
\[
E^{i,0} := \{ ij \in E^* \mid p_i \text{ is contained in } T_{i,0} \}, \\
E^{i,+} := \{ ij \in E^* \mid p_i \text{ is contained in } T_{i,+} \}.
\]
By the positivity of \( a \), we see that \( \{E^{i,0}, E^{i,+}\} \) is a partition of \( E^* \cap \delta i \).

Lemma 5. Let \( x : E \to \mathbb{R}_+ \) be an edge-capacity function with \( 0 \leq x \leq u \), and let \( p : V \to \mathbb{S} \) be a proper potential. If \( x \) and \( p \) satisfy the following conditions (A1–5), then \( x \) and \( p \) are optimal solutions for FNTB and DTB, respectively:

- (A1) For each \( ij \in E \), if \( \text{dist}(p_i, p_j) > a_{ij} \), then \( x_{ij} = u_{ij} \).
- (A2) For each \( ij \in E \), if \( \text{dist}(p_i, p_j) < a_{ij} \), then \( x_{ij} = 0 \).
- (A3) For each \( i \in \bigcup_{s \in S} V_s \), it holds \( x(E^{i,0}) = x(E^{i,+}) \leq c_i \). If \( \text{size}(p_i) > 0 \), then \( x(E^{i,0}) = x(E^{i,+}) = c_i \).
- (A4) For each \( i \in V_0 \) and \( s \in S \), it holds \( x(E^{i,s}) \leq c_i \) and \( x(E^{i,s}) \leq \sum_{s' \neq s} x(E^{i,s'}) \). If \( \text{size}_{s}(p_i) > 0 \), then \( x(E^{i,s}) = c_i \).
- (A5) For each \( s \in S \), it holds \( x(\delta s) \geq r_s \). If \( \text{dist}_{s}(p_s) > 0 \), then \( x(\delta s) = r_s \).

Proof. Let \( x \) and \( p \) satisfy (A1–5). For the feasibility of \( x \), it is sufficient to show that, for each \( s \in S \), there exists a flow satisfying the capacities \( c \) and \( e \) that connects \( s \) and \( S \setminus \{ s \} \) with flow-value \( r_s \). To prove this, we decompose \( x \) into a separately-capacitated multiflow. An \( S \)-path is a path connecting distinct terminals. Consider the following algorithm, which takes \( x \) as an input and outputs a function \( \lambda : P \to \mathbb{R}_+ \), where \( P \) is a set of \( S \)-paths:

0. Let \( P = \emptyset \).
1. Take \( s \in S \) and an edge \( sj \) satisfying \( x(sj) > 0 \). If such a pair does not exist, then stop the algorithm; output \( (P, \lambda) \). Otherwise, let \( j_0 \leftarrow s, j_1 \leftarrow j, \mu \leftarrow x(sj), \ t \leftarrow 1 \).
2. If \( j_t \) is a terminal, then add \( P = (j_0, j_1, \ldots, j_t) \) to \( P \) and let \( \lambda(P) := \mu > 0 \). Update \( x(e) \leftarrow x(e) - \mu \) on each edge \( e \) in \( P \), and return to Step 1. Otherwise go to Step 3.
3. If \( j_t \in \bigcup_{s \in S} V_s \), then \( j_t - j_{t-1} \in E^{j_{t-1}j_t} \) or \( j_t - j_{t-1} \in E^{j_{t-1}j_t} \) by (A2) and \( x(j_{t-1}j_t) > 0 \). In the former case, take \( j_t j_{t+1} \in E^{j_{t-1}j_t} \) with \( x(j_{t-1}j_{t+1}) > 0 \). Such an edge exists by the former part of (A3). In the latter case, take \( j_t j_{t+1} \in E^{j_{t-1}j_t} \) with \( x(j_{t-1}j_{t+1}) > 0 \). Update \( \mu \leftarrow \min\{\mu, x(j_{t-1}j_{t+1})\} \), \( t \leftarrow t + 1 \), and return to Step 2.
If \( j_t \in V_0 \), then \( j_{t-1} j_t \in E^{i,s} \) (as we will show). Take \( s' \neq s \) with maximum \( x(E^{j_t,s'}) > 0 \), and take \( j_t j_{t+1} \in E^{i,s'} \) with \( x(j_t j_{t+1}) > 0 \). Such an edge exists by \( x(j_{t-1} j_t) > 0 \) and the former part of (A4). Update

\[
\mu \leftarrow \min \left\{ \mu, x(j_t j_{t+1}), \frac{\min \left\{ \sum_{s'' \neq s'} x(E^{j_t,s''}) - x(E^{j_t,s'}), s'' \neq s, s' \right\}}{2} \right\},
\]

and \( t \leftarrow t + 1 \). Note that \( \mu > 0 \) by the maximality of \( x(E^{i,s'}) \). Return to Step 2.

Suppose that we add \( (j_0, j_1, \ldots, j_t) \) to \( \mathcal{P} \) in Step 2. Observe that \( j_{t+1} \) is at a side opposite to \( j_{t-1} \) based on \( j_t \) for each \( t = 1, \ldots, \ell - 1 \). By the positivity of \( \alpha \) and (A2), \{\( j_{t-1}, j_t, j_{t+1} \)\} are distinct and

\[
dist(p_{j_{t-1}}, p_{j_{t+1}}) = dist(p_{j_{t-1}}, p_{j_t}) + size(p_{j_t}) + dist(p_{j_t}, p_{j_{t+1}})
\]

if \( j_t \in \bigcup_{s \in S} V_s \), and

\[
dist(p_{j_{t-1}}, p_{j_{t+1}}) = dist(p_{j_{t-1}}, p_{j_t}) + size_s(p_{j_t}) + size_s(p_{j_t}) + dist(p_{j_t}, p_{j_{t+1}})
\]

if \( j_t \in V_0 \), where \( j_{t-1} \in V_s \) and \( j_{t+1} \in V_{s'} \) (\( s \neq s' \)). Since \( T \) is a tree, we can show

\[
dist(p_{j_t}, p_{j_t}) = \sum_{i=0}^{t-1} dist(p_{j_i}, p_{j_{i+1}}) + \sum_{1 \leq i \leq t-1, j_t \neq j_i} size(p_{j_i}) + size_{j_t}(p_{j_i}) + size_{j_t}(p_{j_i}) \tag{8}
\]

by an induction, where \( j_{t'} \in V_0 \) (if exists); see also [12, Lemma 3.9]. Hence \( (j_0, j_1, \ldots, j_t) \) is a “shortest path on \( T^{i,s} \) from \( j_0 \) to \( j_t \), and \( j_0, \ldots, j_t \) are distinct.

Thus after \( |V| \) executions of Step 3, the algorithm adds a path \( P \to \mathcal{P} \) in Step 2. Also the algorithm keeps (A2) and the former parts of (A3–4). To see it for (A4), suppose that

\[
\sum_{e \in E^{i,s'}} x(E^{j_t,s'}) - x(E^{j_t,s'} - E^{j_t,i}) \geq 2 \mu.
\]

Thus after the decrease of the value of \( x \) along with \( P \), it satisfies that

\[
\sum_{e \in E^{i,s'}} x(E^{j_t,s'}) - x(E^{j_t,s''}) \geq 0.
\]

After the decrease of the value of \( x \) along with a path, it becomes \( x(e) = 0 \) for at least one edge \( e \in E \), or becomes \( \sum_{e \in E^{i,s}} x(E^{i,s}) - x(E^{i,s}) = 0 \) for at least one pair of \( i \in V_0 \) and \( s \in S \). The algorithm keeps those values to be zero in the remaining execution, implying that it terminates after adding at most \( O(m + n) \) paths to \( \mathcal{P} \). To see it, suppose that after the decrease of the value of \( x \) along with a path, it becomes \( \sum_{e \in E^{i,s}} x(E^{i,s}) - x(E^{i,s}) = 0 \) for \( i \in V_0 \) and \( s \in S \). If the algorithm chooses a path \( (j_0, \ldots, j_t) \) for adding to \( \mathcal{P} \) in the remaining execution, then by the maximality of \( x(E^{i,s}) \), it should satisfy that \( j_{t-1} j_t \in E^{i,s} \) or \( j_{t+1} j_t \in E^{i,s} \). Thus \( \sum_{e \in E^{i,s}} x(E^{i,s}) - x(E^{i,s}) \) does not change by the decrease of the value of \( x \) along with \( (j_0, \ldots, j_t) \).

We have shown the algorithm always terminates in finite steps. For the output \( f = (\mathcal{P}, \lambda) \), let \( f(e) := \sum_{P \in \mathcal{P}, e \in P} \lambda(P) \) for \( e \in E \), and let \( f(i) := \sum_{P \in \mathcal{P}, i \in P} \lambda(P) \) for \( i \in V \). Also let \( \mathcal{P}_s \subseteq \mathcal{P} \) be the subset of paths connecting \( s \) to other terminals, and let \( f_s := (\mathcal{P}_s, \lambda_s) \) for \( s \in S \). Clearly, it holds that \( f(e) \leq x(e) \leq u(e) \) for \( e \in E \). For \( i \in V_s \) \((s \in S)\), if a path \( P \in \mathcal{P} \) goes through \( i \), then \( P \) must be contained in \( \mathcal{P}_s \). Thus by the former part of (A3),

\[
f_s(i) = f(\ell) \leq x(E^{i,s}) (\leq x(E^{i,s} + )) \leq c(i).
\]

Also, \( f_s(i) \leq f(i) \leq c(i) \) for any \( s' \neq s \). On the other hand, for \( i \in V_0 \), if a path in \( \mathcal{P}_s \) \((s \in S)\) goes through \( i \), then it must include an edge contained in \( E^{i,s} \). Thus by the former part of (A4), we have \( f_s(i) \leq x(E^{i,s}) \leq c(i) \). Therefore \( f \) is a separately-capacitated multiflow. Moreover, \( f_s \) satisfies the requirement \( r \) by the former part of (A5). Thus \( x \) is a feasible solution of FNTB.
We next show the optimality of \(x\) and \(p\). First observe that when the algorithm terminates, all edges \(e \in E\) satisfy \(x(e) = 0\). In fact, if there exists an edge \(e \in E\) with \(x(e) > 0\), then we can construct an \(S\)-path with edges having positive \(x\)-values by repeating to apply the former parts of (A3–4). Thus \(f(e) = x(e)\) (\(e \in E\)) for the original input \(x\). We see that

\[
\sum_{ij \in E} a_{ij}x_{ij} - \sum_{s \in S} r_s \text{dist}(0, p_s) + \sum_{i \in V \setminus S} c_i \text{size}(p_i) + \sum_{ij \in E} u_{ij}(\text{dist}(p_i, p_j) - a_{ij})^+
\]

\[
= \sum_{ij \in E} (\text{dist}(p_i, p_j) - a_{ij})^+(u_{ij} - x_{ij}) + \sum_{ij \in E} (a_{ij} - \text{dist}(p_i, p_j))^+x_{ij} + \sum_{ij \in E} x_{ij} \text{dist}(p_i, p_j)
\]

\[
= \sum_{ij \in E} (\text{dist}(p_i, p_j) - a_{ij})^+(u_{ij} - x_{ij}) + \sum_{ij \in E} (a_{ij} - \text{dist}(p_i, p_j))^+x_{ij}
\]

\[
+ \sum_{s \in S} \sum_{i \in V_s}(c_i - f(i)) \text{size}(p_i) + \sum_{i \in V_0} \sum_{s \in S}(c_i - f_s(i)) \text{size}_s(p_i) + \sum_{s \in S}(f(s) - r_s) \text{dist}(0, p_s),
\]

where we use \(a + (d - a)^+ = d + (a - d)^+\) for \(a, d \in \mathbb{R}\) and

\[
\sum_{ij \in E} f(ij) \text{dist}(p_i, p_j) + \sum_{s \in S} \sum_{i \in V_s} f(i) \text{size}(p_i) + \sum_{i \in V_0} \sum_{s \in S} f_s(i) \text{size}_s(p_i)
\]

\[
= \sum_{ij \in E} \sum_{P \in \mathcal{P}, i,j \in E(P)} \lambda(P) \text{dist}(p_i, p_j)
\]

\[
+ \sum_{s \in S} \sum_{i \in V_s} \sum_{P \in \mathcal{P}, i \in V(P)} \lambda(P) \text{size}(p_i) + \sum_{i \in V_0} \sum_{s \in S} \sum_{P \in \mathcal{P}, i \in V(P)} \lambda_s(P) \text{size}_s(p_i)
\]

\[
= \sum_{st} \sum_{P \in \mathcal{P}, P \text{ connects } st} \lambda(P) \text{dist}(p_s, p_t) = \sum_{s \in S} f(s) \text{dist}(0, p_s)
\]

by (8). We see \(f(i) = x(E^i, 0)\) (\(= x(E^i, +)\)) for \(i \in \bigcup_{s \in S} V_s\), and \(f_s(i) = x(E^i, s)\) for \(i \in V_0\) and \(s \in S\). Also \(f(s) = x(s)\) for \(s \in S\). Then (9) is zero by (A1–2) and the latter parts of (A3–5). By Proposition 4, we conclude that \(x\) and \(p\) are both optimal. \(\blacksquare\)

**Remark 6.** Suppose the input edge-capacity \(x\) satisfies \(x(\delta i) \in \mathbb{Z}_+\) for any \(i \in V\). Then \(\mu\) is always half-integral, and the integrality of \(x(\delta i)\) is also kept in the execution of the algorithm. Thus the output multiflow is half-integer-valued. This argument will be used for proving a min-max theorem (Theorem 2) for a separately-capacitated multiflow later.

The decomposition algorithm is based on [11, Lemma 4.5]; see also [14, Lemma 3.3].

The existence of an edge-capacity \(x\) satisfying (A1–5) can be checked by solving the *undirected circulation problem*. This fact leads a simple descent algorithm for DTB and FNTB. Notice that a potential \(p : V \to \mathbb{S}\) can be identified with a vector in \(\mathbb{S}^n\). For brevity we write \(p \in \mathbb{S}^n\) below. Let \(h_a = h : \mathbb{S}^n \to \mathbb{R}\) be a function defined by

\[
h(p) := -\sum_{s \in S} r_s \text{dist}(0, p_s) + \sum_{i \in V \setminus S} c_i \text{size}(p_i) + \sum_{ij \in E} u_{ij}(\text{dist}(p_i, p_j) - a_{ij})^+
\]

if \(p \in \mathbb{S}^n\) is a potential and \(h(p) := \infty\) otherwise. Then DTB is precisely a minimization of \(h\) over \(\mathbb{S}^n\). Consider the following algorithm DESCENT:
and c

65:10 Terminal Backup, Multiflow, and Discrete Convexity

Algorithm 1 DESCENT.

0. Initialize \( p \equiv 0 \) (i.e., \( p(i) = 0 \) for any \( i \in V \)).
1. Check the sufficiency of the optimality of \( p \) by searching \( x \) satisfying (A1–5).
2. If \( x \) is found, then \( x \) and \( p \) are optimal; stop.
3. Otherwise find \( q \in S^n \) with \( h(q) < h(p) \); update \( p \) by \( q \) and go to Step 1.

We give more details of DESCENT in Section 2.3. As for Step 1, we can also do Step 3 by the undirected circulation problem; \( q \) is computed by the certificate of the nonexistence of \( x \). In the following subsections, we introduce the undirected circulation problem and discuss how to find \( x \) or \( q \) in each case.

2.2 Checking the Optimality

Let \( (U, F) \) be an undirected graph, and let \( \underline{b} : F \to \mathbb{R} \) and \( \overline{b} : F \to \mathbb{R} \) be lower and upper capacity functions satisfying \( \underline{b}(e) \leq \overline{b}(e) \) for each \( e \in F \). The graph \( (U, F) \) may contain self-loops but no multiedges. The circulation problem on \( (U, F, \underline{b}, \overline{b}) \) is the problem of finding an edge-weight \( y : F \to \mathbb{R} \) satisfying \( \underline{b}(e) \leq y(e) \leq \overline{b}(e) \) for each \( e \in F \) and \( \sum_{e \in F} y(e) = 0 \) for each \( i \in U \). Such a \( y \) is called a circulation.

Let \( 3^U \) denote the set of pairs \( (Y, Z) \) of two subsets \( Y, Z \subseteq U \) with \( Y \cap Z = \emptyset \). For \( (Y, Z) \in 3^U \), let \( \chi_{Y,Z} := \sum_{i \in Y} \chi_i - \sum_{i \in Z} \chi_i \in \mathbb{R}^U \). Define the cut function \( \kappa : 3^U \to \mathbb{R} \) by

\[
\kappa(Y, Z) := \sum_{i,j \in F} ((\chi_{Y,Z}((i,j)))^+\underline{b}(ij) - (\chi_{Y,Z}((i,j)))^+\overline{b}(ij)) \quad ((Y, Z) \in 3^U).
\]

It is well-known that the feasibility of the circulation problem is characterized via the cut function. We can show it by reducing to Hoffman’s circulation theorem. A cut \( (Y, Z) \in 3^U \) with \( \kappa(Y, Z) > 0 \) is called violating, and is called maximum violating if it attains the maximum \( \kappa(Y, Z) \) among all violating cuts.

Lemma 7 (see, e.g., [16, Theorems 2.4, 2.7]). Let \( ((U, F), \underline{b}, \overline{b}) \) be an undirected network.

1. The circulation problem is feasible if and only if \( \kappa(Y, Z) \leq 0 \) for any \( (Y, Z) \in 3^U \).
2. If \( \underline{b} \) and \( \overline{b} \) are integer-valued, then there exists a feasible half-integer-valued circulation \( y : E \to \mathbb{Z}_+^n \).
3. Under the same assumption, we can obtain a feasible half-integer-valued circulation or a maximum violating cut in \( O(MF(|U|, |F|)) \) time.

Let us return to our problem. For a given proper potential \( p \in S^n \), the existence of \( x : E \to \mathbb{R}_+ \) satisfying (A1–5) reduces to the undirected circulation problem on the following network \( \mathcal{N}_p := ((U, F), \underline{c}, \overline{c}, \tau) \). See Figure 2 for the following construction.

For each \( i \in \bigcup_{s \in S} V_s \), divide \( i \) into two nodes \( U_i := \{i^0, i^+\} \), and connect nodes by an edge \( i^0i^+ \). For representing (A3), let \( \underline{c}(i^0i^+) : = -c_i \), and let \( \tau(i^0i^+) := 0 \) if \( \text{size}(p_i) > 0 \) and \( \tau(i^0i^+) := -c_i \) if \( \text{size}(p_i) = 0 \). For each \( i \in V_0 \), divide \( i \) into 2k nodes \( U_i := U^0_i \cup U^+_i \), where \( U^0_i := \{i^0, i^1, i^2, \ldots, i^k\} \) and \( U^+_i := \{i^1, i^2, \ldots, i^k, i^+\} \), and connect them by edges \( i^0i^+ \) for \( s \in S \) and \( i^0i^+ \) for distinct \( s, s' \in S \). For representing (A4), let \( \underline{c}(i^0i^+s^0s^+) := -c_i \), and let \( \tau(i^0i^+s^0s^+) := 0 \) if \( \text{size}(p_i) = 0 \) and \( \tau(i^0i^+s^0s^+) := -c_i \) if \( \text{size}(p_i) > 0 \). Also let \( \underline{c}(i^0i^+s^0s^+) := 0 \) and \( \tau(i^0i^+s^0s^+) := \infty \). For each \( s \in S \), let \( s^0 := s \) and \( U_s := \{s^0\} \), and add a self-loop \( s^0s^0 \). For representing (A5), let \( \underline{c}(s^0s^0) := -\infty \) if \( \text{dist}(0, p_s) = 0 \) and \( \underline{c}(s^0s^0) := -r_s \) if \( \text{dist}(0, p_s) > 0 \), and let \( \tau(s^0s^0) := -r_s \).
For each edge \( ij \in E \), if \( \text{dist}(p_i, p_j) < a_{ij} \), then \( x_{ij} = 0 \) by (A2). Thus we remove those edges. Let \( E_\geq \) be the set of edges \( ij \in E \) with \( \text{dist}(p_i, p_j) \geq a_{ij} \), and let \( E_\leq \) be the set of edges \( ij \in E \) with \( \text{dist}(p_i, p_j) < a_{ij} \). We replace endpoints of each edge \( ij \in E_\geq \cup E_\leq \). If \( i \in V_0 \) and \( j \in V_s \), then replace \( ij \) with \( i0 \). If \( i, j \in V_s \) and \( p_i \) is closer to 0 than \( p_j \), i.e., \( \text{dist}(0, p_i) < \text{dist}(0, p_j) \), then replace \( ij \) with \( i0 \). We identify those replaced edges with the original edges. Let \( c(ij) := 0 \) if \( ij \in E_\leq \) and \( c(ij) := u_{ij} \) if \( ij \in E_\geq \). \( U \) and \( F \) are defined as the union of all nodes and edges in the above, respectively.

\[\text{Theorem 8.}\]

Let \( N_p = ((U, F), \xi, \tau) \) be the undirected network constructed from a proper potential \( p \in S^n \). If it has a (half-integer-valued) circulation \( y : F \to \mathbb{R} \), then an edge-capacity function \( x : E \to \mathbb{R}_+ \) defined by

\[
x(e) = \begin{cases} 
  y(e) & \text{if } e \in E_\geq \cup E_\leq, \\
  0 & \text{otherwise (} e = ij \text{ with } \text{dist}(p_i, p_j) < a_{ij} \text{)}
\end{cases}
\]

satisfies (A1–5).

\[\text{Proof.}\]

We can obtain (A1–5) from definitions immediately. For example, the former part of (A4) follows from \( x(E_i^+) = -y(i^00,i^+,s) \leq -\xi(i^00,i^+,s) = c_i \) and

\[
x(E_i^+) = -y(i^00,i^+,s) = \sum_{s' \neq s} y(i^00,i^+,s') \leq \sum_{s' \neq s} -\xi(i^00,i^+,s') = \sum_{s' \neq s} x(E_i^+),
\]

and the latter part of (A4) follows from \( -y(i^00,i^+,s) \geq -\tau(i^00,i^+,s) = c_i \) for \( i \in V_0 \) and \( s \in S \) with \( \text{size}(s, p_i) > 0 \).

\[\text{2.3 Finding a Descent Direction}\]

If the algorithm in Lemma 7 outputs a circulation in \( N_p \), then an optimal edge-capacity is computed from the circulation, and \( p \) is optimal by Lemma 5 and Theorem 8. Otherwise the algorithm outputs a maximum violating cut. We show that we can find \( q \in S^n \) with \( h(q) < h(p) \) using the maximum violating cut. A basic idea is to modify each subtree \( p_i \), according to the intersection pattern of the maximum violating cut with \( U_i \), so that the objective function \( h \) decreases. This implies the necessity of Lemma 5 and the strong duality of Proposition 4.
We begin with introducing the notion of basic moves for a subtree. For an $s$-type subtree $T = [l, l']_s$, we denote its endpoints by $v_0(T) := (l, s) \in T$ and $v_+(T) := (l', s) \in T$. When we remove $T$ from $T$, there appear two connected components. Let $T'_0$ be the component containing 0 ($\in T$), and $T'_+ = \emptyset$ be the other. We can expand the subtree $T$ by adding a node next to $T$. There are two nodes next to $T$, one is contained in $T'_0$ and the other is contained in $T'_+$. The 0-expansion is the operation to add that node contained in $T_{i,0}$ to $T$, and the $+$-expansion is the operation to add that node contained in $T_{i,+}$ to $T$. If $T$ satisfies $\text{size}(T) > 0$, then we can shrink $T$ by removing $v_0(T)$ or $v_+(T)$ from $T$. The 0-shrinkage is the operation to remove $v_0(T)$ from $T$, and the $+$-shrinkage is the operation to remove $v_+(T)$ from $T$.

For a 0-type subtree $T = [s]_{s} \subseteq S$, we denote its endpoints by $v_s(T) := (l, s) \in T$ for $s \in S$. When we remove $T$ from $T$, there appear $k$ ($= |S|$) connected components. For $s \in S$, let $T'_s$ be the component which is contained in $T_s$. As above, we can expand the subtree $T$ by adding a node next to $T$. There are $k$ nodes next to $T$, and each $T'_s$ ($s \in S$) contains exactly one such a node. The $(s,+)$-expansion for $s \in S$ is the operation to add that node contained in $T'_s$ to $T$. If $T$ satisfies $\text{size}(T) > 0$ for $s \in S$, then we can shrink $T$ by removing $v_0(T)$ from $T$. The $(s,+)$-shrinkage for $s \in S$ with $\text{size}(T) > 0$ is the operation to remove $v_0(T)$ from $T$. For $s \in S$, if $\text{size}(T) = 0$ for any other $s' \in S$, then we can shrink $T$ by removing 0 ($\in T$) from $T$. The $(s,0)$-shrinkage for such $s \in S$ is the operation to remove 0 from $T$. We call these expansion and shrinkages basic moves.

Let $(Y, Z) \in 3^U$ be a cut. From $(Y, Z)$, the modification $p^{Y,Z}$ of $p$ is defined as follows.

For $s \in S$, do:
- If $s^0 \in Y$, then 0-expand and +-shrink $p_s$.
- If $s^0 \in Z$, then +-expand and 0-shrink $p_s$.

For $i \in \bigcup_{s \in S} V_s$, do:
- If $i^0 \in Y$, then 0-expand $p_i$. If $i^0 \in Z$, then 0-shrink $p_i$.
- If $i^+ \in Y$, then +-expand $p_i$. If $i^+ \in Z$, then +-shrink $p_i$.

For $i \in V_0$, do:
- If $U_i \cap (Y \cup Z) = \emptyset$, then we do the following for each $s \in S$:
  - If $i^0 \in Y$, then $(s,+)$-expand $p_i$. If $i^0 \in Z$, then $(s,+)$-shrink $p_i$.
- If $i^0 \in Z$ for some $s \in S$, then $(s,0)$-shrink $p_i$. Also do the following:
  - If $i^+ \in Y$, then $(s,+)$-expand $p_i$. If $i^+ \in Z$, then $(s,+)$-shrink $p_i$.

There may exists $i \in V$ that such a move cannot be defined, e.g., $i \in \bigcup_{s \in S} V_s$ with $\text{size}(p_i) \leq 1/2$ and $\{i^0, i^+\} \subseteq Z$, or $j \in V_0$ with $\{j^0, j^+\} \subseteq Z$. If the moves can be defined for all $i \in V$, then the cut $(Y, Z)$ is called movable. For a movable cut $(Y, Z) \in 3^U$, we denote the modified potential by $p^{Y,Z}$. We call a node $(l, s) \in T$ even if the number of edges between $(l, s)$ and 0 is even, and odd otherwise. A basic move is said to be upward if the added node is even or the removed node is odd. A basic move is said to be downward if the added node is odd or the removed node is even. A movable cut $(Y, Z) \in 3^U$ is upward-moving (resp. downward-moving) if all basic moves occurring in the modification from $p$ to $p^{Y,Z}$ are basic upward moves (resp. basic downward moves). Let denote the sets of all upward-movable cuts and downward-movable cuts by $\mathcal{M}^U$ and $\mathcal{M}^D$, respectively.

**Lemma 9.** For $(Y, Z) \in \mathcal{M}^U \cup \mathcal{M}^D$, it holds $h(p^{Y,Z}) - h(p) = -\kappa(Y, Z)/2$.

Thus we are motivated to obtain an upward- or downward-movable cut $(Y, Z)$ with a positive $\kappa(Y, Z)$ value. The following lemma says that we can do this efficiently given a maximum violating cut.
Lemma 10. Given a maximum violating cut, we can obtain an upward-movable cut $(Y, Z) \in \mathcal{M}^\uparrow$ and a downward-movable cut $(Y', Z') \in \mathcal{M}^\downarrow$ satisfying

$$\kappa(Y, Z) = \max_{(Y'', Z'') \in \mathcal{M}^\uparrow} \kappa(Y'', Z''), \quad \kappa(Y', Z') = \max_{(Y'', Z'') \in \mathcal{M}^\downarrow} \kappa(Y'', Z'')$$

(11)

in $O(kn)$ time. Moreover, at least one of $\kappa(Y, Z)$ and $\kappa(Y', Z')$ is positive.

Theorem 11. Let $\mathcal{N}_p := ((U, F), \mathcal{E}, \mathcal{C})$ be the undirected network constructed from a proper potential $p \in S^n$. Suppose that the instance is infeasible. Given a maximum violating cut, we can obtain a proper potential $\hat{p} \in S^n$ with $h(\hat{p}) < h(p)$ in $O(kn)$ time.

Proof. By Lemma 10, we can obtain an upward-movable cut $(Y, Z) \in \mathcal{M}^\uparrow$ and a downward-movable cut $(Y', Z') \in \mathcal{M}^\downarrow$ satisfying (11) in $O(kn)$ time. Let $(Y'', Z'')$ be the cut that attains maximum $\kappa$-value among $\{(Y, Z), (Y', Z')\}$, and let $q := p^{Y'', Z''}$. Then $h(q) < h(p)$ by Lemmas 9 and 10. We can make $q$ proper by the procedure given in the first part of the proof of Proposition 4.

Now we are ready to present the details of DESCENT. First construct $\mathcal{N}_p$ from the current proper potential $p \in S^n$, and run the algorithm given in Lemma 7 to solve the circulation problem; this corresponds to Step 1 given in the procedure at the end of Section 2.1. If a feasible half-integral-valued circulation is obtained, then a half-integral optimal edge-capacity $\chi$ is computed by Theorem 8; this corresponds to Step 2. Otherwise a maximum violating cut is obtained, and then a proper potential $q \in S^n$ with $h(q) < h(p)$ is computed by Theorem 11; this corresponds to Step 3. One iteration of this algorithm can be done in $O(MF(kn, m + k^2n))$ time.

The value $-h(p)$ is at most $mA$ (by Proposition 4) and $-h(p) \in \mathbb{Z}_+^n$. Thus the number of iterations is at most $O(mUA)$. Actually, this analysis of the time complexity is not tight. In fact, the number of iterations can be evaluated as $O(nA)$.

If a potential $q \in S^n$ is obtained from a potential $p$ by a modification defined by a movable cut on $\mathcal{N}_p$, then we say that $q$ is a neighbor of $p$, that is, there exists a movable cut $(Y', Z') \in \mathcal{Z}^c$ such that $q = p^{Y', Z'}$. For $p, q \in S^n$, define a distance $d_{S^n}(p, q)$ by the minimum length of a sequence $(p = p_0, p_1, \ldots, p_\ell = q)$ such that $p_t$ is a neighbor of $p_{t-1}$ for all $t = 1, \ldots, \ell$. Let $\text{opt}(h)$ denote the set of minimizers of $h$, and let $d_{S^n}(p, \text{opt}(h)) := \min_{q \in \text{opt}(h)} d_{S^n}(p, q)$.

Lemma 12. Starting with an initial potential $p_0 \in S^n$, DESCENT finds an optimal potential at most $d_{S^n}(p_0, \text{opt}(h)) + 2$ iterations.

Lemma 12 can be shown by using DCA beyond $\mathbb{Z}^n$. We will discuss it in Section 3.

Lemma 13. There exists an optimal potential $p \in \text{opt}(h)$ satisfying that for any $i \in V$, $p_i$ is contained in $(2nA, 2nA, \ldots, 2nA) \in S$.

Theorem 14. DESCENT solves FNTB in $O(nA \cdot MF(kn, m + k^2n))$ time.

Proof. We can only consider the potentials satisfying the condition in Lemma 13. Any pair of such potentials $p, q \in S$ satisfies $d_{S^n}(p, q) = O(nA)$. Then the statement follows from Lemma 12.

We note that Theorem 14 is shown under the positivity assumption of the edge-cost $a$. We prove Theorem 2 using Theorem 14.
Proof of Theorem 2. Let $f = (\mathcal{P}, \lambda)$ be a separately-capacitated multiflow. Recall that $f_s = (\mathcal{P}_s, \lambda_{\mathcal{P}_s})$, where $\mathcal{P}_s \subseteq \mathcal{P}$ is a subset of paths connecting $s$ to other terminals. Let $\text{val} f := \sum_{P \in \mathcal{P}} \lambda(P)$ and $\text{val} f_s := \sum_{P \in \mathcal{P}_s} \lambda(P)$ for $s \in S$. Then $\text{val} f_s$ is at most the capacity of any $\{s\} \setminus \{s\}$ cut. Thus $\text{val} f = \left(\frac{1}{2}\right) \sum_{s \in S} \text{val} f_s \leq \left(\frac{1}{2}\right) \sum_{s \in S} \nu_s$.

Consider an instance $((\mathcal{V}, E), S, u, c, a, r)$ of FNTB, where $a \equiv 1$ and $r_s := \nu_s$ for each $s \in S$. Since $u$ clearly satisfies (1), this instance is feasible. Then DESCENT outputs a half-integral optimal edge-capacity $x$ and an optimal potential $p$. Since $x$ and $p$ satisfy the conditions (A1–5), we can apply the decomposition algorithm in the proof of Lemma 5 for $x$, and obtain a separately-capacitated multiflow $f$. Then $\text{val} f = \left(\frac{1}{2}\right) \sum_{s \in S} f(s) \geq \left(\frac{1}{2}\right) \sum_{s \in S} \nu_s = \left(\frac{1}{2}\right) \sum_{s \in S} \nu_s$. Moreover, since $x$ comes from a half-integral circulation (Theorem 8), $x$ satisfies $x(\delta i) \in \mathbb{Z}_+$ for any $i \in V \setminus S$. In fact, for $i \in \bigcup_{s \in S} V_s$, it is observed from $x(\delta i) = -2y(i^0 i^*)$, and for $i \in V_0$, it is observed from $x(\delta i) = \sum_{s \in S} -y(i^s, i^*) = 2 \sum_{s \in S} y(i^{s,0}, i^{s,\beta})$. Then by Remark 6, the decomposition algorithm outputs a half-integral-valued multiflow.

The time complexity result follows from that FNTB can be solved in $O(n \cdot \text{MF}(kn, m + k^2 n))$ time by Theorem 14, and the decomposition algorithm runs in $O((m + kn) n)$ time.

2.4 Scaling Algorithm

The time complexity of DESCENT is pseudo-polynomial. We improve it by combining with a (cost-)scaling method.

Let $\gamma \in \mathbb{Z}_+$ be an integer such that $2^\gamma \geq A$. The scaling algorithm consists of $\gamma + 1$ phases. In $t$-th phase, solve DTB with an edge-cost $a_t : E \rightarrow \mathbb{Z}_+$ defined by $a_t(e) := \lfloor a(e)/2^t \rfloor$ (e $\in E$), i.e., minimize $h_{a_t}$ (Recall $h_a$ is defined by (10)). Here $\lfloor . \rfloor$ is the round-up operator. Note that all $a_t(e)$ are positive. Begin with $t = \mu$, and decrease $t$ one-by-one. Then, when $t = 0$, the problem coincides with the original DTB. In each $t$-phase, we use DESCENT to minimize $h_{a_t}$. At the initial phase $t = \mu$, we run DESCENT with the starting point $p_0 \in \mathbb{S}^n$, where $(p_0)_i = 0$ for all $i \in V$. For $t$-phase with $t \leq \mu - 1$, the starting point is determined from the obtained optimal potential in the previous phase. Let $2[l, l'] := [2l, 2l']$, and $2[l]_{s \in S} := [2l]_{s \in S}$. For a potential $p \in \mathbb{S}^n$, define a new potential $2p \in \mathbb{S}^n$ by $(2p)_i := 2p_i$, for $i \in V$.

Lemma 15. Let $p \in \mathbb{S}^n$ be an optimal potential for $t$-phase ($t = 1, \ldots, \mu$). Then the potential $2p \in \mathbb{S}^n$ is optimal for DTB with an edge-cost $2a_t$.

Proof. By the strong duality of Proposition 4, there exists a solution $x : E \rightarrow \mathbb{R}$ for FNTB, such that $\sum_{e \in E} a_t(e)x(e) = -h_{a_t}(p)$. Then $\sum_{e \in E} 2a_t(e)x(e) = -2h_{a_t}(2p)$ holds, which implies the optimality of $2p$ by (the weak duality of) Proposition 4.

Observe that $a_t - 1 = 2a_t - \sum_{e \in F} \chi_e$, where $F := \{e \in E \mid a_t(e) \text{ is odd}\}$. The key property is the following sensitivity result.

Lemma 16. Let $a : E \rightarrow \mathbb{Z}_+$ be a positive edge-cost. Let $e \in E$ be an edge satisfying $a(e) \geq 2$, and $a' := a - \chi_e$. Let $p \in \text{opt}(h_a)$. Then $d_{\mathbb{S}^n}(p, \text{opt}(h_{a'})) \leq 2$.

We prove Lemma 16 in Section 3.3 using the notion of discrete convexity.

Proof of Theorem 1. For the initial phase $t = \mu$, an optimal potential can be obtained in $O(n)$ iterations of DESCENT by Lemmas 12 and 13. For each remaining phase, an optimal potential can be obtained in $O(m)$ iterations of DESCENT by Lemmas 12, 15 and 16. Thus $O(n + m \log A) = O(m \log A)$ iterations of DESCENT are sufficient. Recall that we assume the positivity of the edge-cost $a$. When $a$ is not positive, the perturbation (Remark 3) is needed. Thus the maximum of edge-costs is $O(mUA)$. Then the theorem follows.
3 Discrete Convex Analysis for Node-Connectivity Terminal Backup

The theory of DCA beyond $\mathbb{Z}^n$ gives an algorithm, called the steepest descent algorithm (SDA), for minimizing L-convex functions on certain graph structures. We first introduce the L-convexity and SDA, and next show that DESCENT is precisely SDA for an L-convex function. Then Lemma 12 immediately follows. Finally, we discuss a sensitivity argument, which shows Lemma 16.

3.1 A General Theory

In this subsection, we briefly introduce a theory of discrete convexity on graph structures specialized to median graphs. See [15] for further details.

We use basic terminologies of poset and lattice. Let $\mathcal{L}$ be a poset (partially ordered set) with a partial order $\preceq$. The principal filter $\mathcal{F}_x$ and the principal ideal $\mathcal{I}_x$ of $x \in \mathcal{L}$ are defined as $\{y \in \mathcal{L} \mid y \succeq x\}$ and $\{y \in \mathcal{L} \mid y \preceq x\}$, respectively. For $x, y \in \mathcal{L}$ with $x \preceq y$, the interval $[x, y]$ is defined as the set of $z \in \mathcal{L}$ satisfying $x \preceq z \preceq y$. We consider a (meet-)semilattice having the minimum element. A median semilattice $\mathcal{L}$ is a semilattice that every principal ideal is a distributive lattice and for any $x, y, z \in \mathcal{L}$, the join $x \lor y \lor z$ exists if $x \lor y, y \lor z$, and $z \lor x$ exist. A Boolean semilattice is a median semilattice that every principal ideal is a Boolean lattice.

Let $G$ be a (possibly infinite) undirected graph. We denote the set of nodes also by $G$. Let $d = d_G$ be the shortest path metric on $G$. The (metric) interval $I(u, v)$ of $u, v \in G$ is the set of $w \in G$ satisfying $d(u, v) = d(u, w) + d(w, v)$. A median graph $G$ is a graph that for any $u, v, w \in G$, $I(u, v) \cap I(v, w) \cap I(w, u)$ is a singleton.

We consider an orientation on edges of a median graph $G$, that takes $u \searrow v$ or $u \nearrow v$ on each edge $uv$. An orientation is admissible if for any 4-cycle $(u_1, u_2, w, u_3)$, $u_1 \searrow u_2$ implies $u_4 \searrow u_3$. It is known [13, Lemma 2.4] that an admissible orientation on a median graph is acyclic. Thus we can define a poset on $G$ by the admissible orientation, i.e., if an edge $uv$ is oriented as $u \nearrow v$, then $u \preceq v$. $G$ with an admissible orientation is well-oriented if $[u, v]$ is a Boolean lattice for any $u, v$ with $u \leq v$. In a well-oriented median graph $G$, it is known [15, Proposition 2] that every principal filter of $G$ is a Boolean semilattice, and every principal ideal of $G$ is a Boolean semilattice with the reversed order.

We can define an L-convex function on a well-oriented median graph $G$. For a function $f : G \to \mathbb{R}$, define the effective domain of $f$ as $\{u \in G \mid f(u) < \infty\}$ and denote by $\text{dom } f$. If a sequence of nodes $(u = u_0, u_1, \ldots, u_\ell = v)$ satisfies that for any $i = 1, \ldots, \ell$, there exist $u', v' \in G$ with $u' \preceq v'$ such that $\{u_{i-1}, u_i\} \subseteq [u', v']$, then the sequence is said to be a $\Delta$-path connecting $u$ and $v$. A subset $X \subseteq G$ is $\Delta$-connected if for any $u, v \in X$, there exists a $\Delta$-path in $X$ connecting $u$ and $v$. A function $f : G \to \mathbb{R}$ is called L-convex if $f$ is $\Delta$-connected and the restrictions of $f$ to every principal filter and ideal are submodular. Here the submodularity on a median semilattice is a rather complicated notion; we give a formal definition in the full version.

The global optimality of an L-convex function $f$ can be characterized by a local condition; $u \in \text{dom } f$ is a minimizer of $f$ if and only if $u$ is a minimizer of $f$ restricted to $\mathcal{F}_u \cup \mathcal{I}_u$. This induces a natural minimization algorithm, called the steepest descent algorithm (SDA):

$\text{Algorithm 2 SDA.}$

0. Initialize $u \in G$ with $f(u) < \infty$.
1. Find a local minimizer $v \in \mathcal{F}_u \cup \mathcal{I}_u$ of $f$.
2. If $f(v) = f(u)$, then stop; output $u$. Otherwise update $u$ by $v$ and go to Step 1.
The number of iterations of SDA is bounded by the $\Delta$-distance from the initial point $u$ and minimizers of $f$. Here the $\Delta$-distance $d^\Delta(u,v)$ of $u,v \in G$ is the minimum length of a $\Delta$-path connecting $u$ to $v$. Let $\text{opt}(f)$ denote the set of minimizers of $f$, and let $d^\Delta(u,\text{opt}(f)) := \min_{v \in \text{opt}(f)} d^\Delta(u,v)$.

**Theorem 17** ([15, Theorem 4.3]). The number of iterations of SDA with the initial point $u \in G$ is at most $d^\Delta(u,\text{opt}(f)) + 2$.

### 3.2 Discrete Convexity in Node-Connectivity Terminal Backup

We show that the dual objective function $h$ defined in (10) is actually an L-convex function, and the algorithm DESCENT is precisely SDA. Define a graph $S$ by connecting two nodes (subtrees) $T,T' \in S$ such that $T$ and $T'$ can transform to each other by a basic move. If we can move $T$ to $T'$ by a basic downward-move (equivalently, we can move $T'$ to $T$ by a basic upward-move), we give an orientation $T \setminus T'$. The graph $S$ is a median graph, but not well-oriented. To make the graph well-oriented, we add a virtual subtree connecting to nodes $(l,s)$ and $(l+1/2,s)$ for each $l \in \mathbb{Z}^*_+$ and $s \in S$. We denote such a virtual subtree by $[l+1/2,l]_s$. Give a natural orientation to each added edge. Let $\bar{S} := S \cup \{[l+1/2,l]_s \mid l \in \mathbb{Z}^*_+, s \in S\}$. Extend $h$ to be a function on $\bar{S}$ by $h(p) := \infty$ if there exists $i \in V$ such that $p_i \in \bar{S} \setminus S$.

**Proposition 18.**

1. $\bar{S}$ is a well-oriented median graph, and so is $\bar{S}^\circ$.
2. $h$ is an L-convex function on $\bar{S}^\circ$.
3. For $p,q \in \bar{S}^\circ$, $d_{\bar{S}^\circ}(p,q) = d^\Delta(p,q)$.
4. The map $(Y,Z) \mapsto p^{YZ}$ is a bijection between $\mathcal{M}^\uparrow$ and $\mathcal{J}_p \cap \text{dom } h$, and $\mathcal{M}^\uparrow$ and $\mathcal{I}_p \cap \text{dom } h$.

**Proof of Lemma 12.** By Lemma 9 and Proposition 18 (4), the cuts $(Y,Z)$ and $(Y',Z')$ in Lemma 10 are minimizers of $h$ on $\mathcal{J}_p$ and $\mathcal{I}_p$, respectively. Therefore DESCENT is precisely SDA for $h$. Thus the number of iterations can be evaluated by Theorem 17, and the statement follows from Proposition 18 (3).

### 3.3 Sensitivity

To prove Lemma 16, we transform the instance $((V,E),S,u,c,a,r)$ of FNTB to an edge-uncapacitated one by a standard technique: Divide each edge $e \in E$ into two edges $e_1,e_2$, and add a new node $v_e$ into the middle of these two edges. Let the edge-costs of $e_1$ and $e_2$ be the same as the original edge-cost of $e$, and let the edge-capacities of $e_1$ and $e_2$ be $\infty$. Let the node-capacity of the added node be $u(e)$. The number of vertices in the new instance is $|V'| + |E'| = n + m$, and the number of edges is $2|E| = 2m$. We denote the new instance by $((V',E'),S,\bar{u},\bar{c},\bar{a},\bar{r})$.

We consider the dual problem DTB for the edge-uncapacitated instance. In this case, we say that $\bar{p} \in \bar{S}^{n+m}$ is a potential for an edge-cost $\bar{a}$ if it satisfies (4) and $\text{dist}(\bar{p}_i,\bar{p}_j) \leq \bar{a}_{ij}$ for any $ij \in E$. Then DTB is a minimization of a function $h_{\bar{a}} : \bar{S}^n \rightarrow \bar{R}$ defined by

$$\tilde{h}_{\bar{a}}(\bar{p}) := -\sum_{s \in S} r_s \text{dist}(0,\bar{p}_s) + \sum_{i \in V \setminus S} \bar{c}_i \text{size}(\bar{p}_i) \quad (12)$$

if $\bar{p}$ is a potential for $\bar{a}$ and $\tilde{h}_{\bar{a}}(\bar{p}) := \infty$ otherwise.
Let \( p \in \mathbb{S}^n \) be a potential for the original instance. We can extend \( p \) to a potential \( \tilde{p} \) for the edge-uncapacitated instance as follows: For \( v = i \in V \), let \( \tilde{p}_v := 2p_i \). For \( v = u_{ij} \) (\( i, j \in E \)), we have two cases \( \text{dist}(p_i, p_j) \leq a_{ij} \) and \( \text{dist}(p_i, p_j) > a_{ij} \). For the former case, let \( \tilde{p}_v \) be any point in \( \mathbb{T} \) (i.e., \( \text{size}(\tilde{p}_v) = 0 \)) satisfying \( \text{dist}(\tilde{p}_i, \tilde{p}_v) \leq a_{ij} \) and \( \text{dist}(\tilde{p}_v, \tilde{p}_j) \leq a_{ij} \). For the latter case, let \( \tilde{p}_v \) satisfy \( \text{dist}(\tilde{p}_i, \tilde{p}_v) = a_{ij} \), \( \text{dist}(\tilde{p}_v, \tilde{p}_j) = a_{ij} \) and \( \text{size}(\tilde{p}_v) = 2(\text{dist}(\tilde{p}_v, \tilde{p}_j) - a_{ij}) > 0 \).

**Proposition 19.** Let \( p \in \mathbb{S}^n \) be an optimal potential for the original instance. Then the extended potential \( \tilde{p} \in \mathbb{S}^{n+m} \) defined above is optimal for the edge-uncapacitated instance.

We first show Lemma 16 for an edge-uncapacitated instance. For brevity, we assume that the original instance \( ((V,E), S, u, c, a, r) \) is also an edge-uncapacitated instance. By Proposition 18 (3), the following is equivalent to Lemma 16.

**Lemma 20.** Let \( a : E \to \mathbb{Z}_+ \) be a positive edge-cost. Let \( ij \in E \) be an edge satisfying \( a(ij) \geq 2 \), and \( a' := a - \chi_{ij} \). Then for any \( p \in \text{opt}(h_a) \), it holds \( d^\Delta(p, \text{opt}(h_{a'})) \leq 2 \).

We prove Lemma 20 via the notion of normal \( \Delta \)-paths. Let \( G \) be an oriented median graph. For nodes \( u, v \in G \) with \( d^\Delta(u,v) = 1 \), let \( \{\langle u, v \rangle\} \) be the minimum interval \([u', v']\) such that \( [u, v] \subseteq [u', v'] \). A \( \Delta \)-path \((u = u_0, u_1, \ldots, u_\ell = v)\) is the normal \( \Delta \)-path from \( u \) to \( v \) if for any \( t = 1, \ldots, \ell - 1 \) and any interval \([u', v']\) with \([u_{t-1}, u_t]\) \subseteq [u', v'] it holds \([u', v'] \cap \{\langle u_t, u_{t+1}\rangle\} = \{u_t\}\). The normal \( \Delta \)-path from \( u \) to \( v \) is uniquely determined, and the length \( \ell \) equals to \( d^\Delta_G(u, v) \) [3, Theorem 6.24]. Let \( u \to v \) denote \( u_1, \) and \( u \to v \) denote \( u_{\ell-1}, \ldots, u_2 \). Also Let \( u \to^{t} v \) denote \( u_t \) for \( t = 0, \ldots, \ell \).

**Lemma 21.** Let \( p, q \in \text{dom} h_a \). Then

\[
\begin{align*}
h_a(p) + h_a(q) &\geq h_a(p \to q) + h_a(q \to p), \\
h_a(p) + h_a(q) &\geq h_a(p \to q) + h_a(q \to p).
\end{align*}
\]

**Lemma 22.** Let \( p, q \in \mathbb{S}^n \) and \( i, j \in V \). Suppose that \( \text{dist}(q_i, q_j) < \text{dist}((p \to q)_i, (p \to q)_j) \) and \( \text{dist}(q_i, q_j) < \text{dist}((p \to q)_i, (p \to q)_j) \). Then for any \( t = 1, \ldots, d^\Delta(p, q) \), it holds \( \text{dist}((p \to^{t} q)_i, (p \to^{t} q)_j) + 1/2 \leq \text{dist}((p \to^{t-1} q)_i, (p \to^{t-1} q)_j) \).

**Proof of Lemma 20.** If \( p \) is a potential for \( a' \), then \( p \in \text{opt}(h_{a'}) \). Suppose that \( p \) is not a potential for \( a' \). Take \( q \in \text{opt}(h_{a'}) \) having the minimum \( \Delta \)-distance from \( p \). Then \( q \in \text{dom} h_a \). Thus by (13) and \( p \in \text{opt}(h_a) \), we have \( h_a(q) \geq h_a(q \to p) \). If \( (q \to p) \in \text{dom} h_{a'} \), then \( h_{a'}(q \to p) = h_a(q \to p) \leq h_a(q) = h_a(q) \) and thus \( (q \to p) \in \text{opt}(h_{a'}) \); a contradiction to the minimality of \( q \). Hence \( (q \to p) \notin \text{dom} h_{a'} \), and \( \text{dist}((q \to p)_i, (q \to p)_j) \geq a'_{ij} + 1/2 > a_{ij} \geq \text{dist}(q_i, q_j) \) (by the half-integrality of \( \text{dist}(\cdot, \cdot) \)). Similarly we have \( \text{dist}((p \to q)_i, (p \to q)_j) > \text{dist}(q_i, q_j) \). Then we can apply Lemma 22 and obtain

\[
\begin{align*}
\text{dist}(p_i, p_j) &\geq \text{dist}((p \to q)_i, (p \to q)_j) + 1/2 \\
&\geq \text{dist}((p \to^{2} q)_i, (p \to^{2} q)_j) + 2/2 \\
&\geq \cdots \geq \text{dist}((p \to q)_i, (p \to q)_j) + (d^\Delta(p, q) - 1)/2.
\end{align*}
\]

By \( \text{dist}(p_i, p_j) \leq a_{ij} \) and \( \text{dist}((p \to q)_i, (p \to q)_j) \geq a'_{ij} + 1/2 = a_{ij} - 1/2 \), we have

\[
d^\Delta(p, q) \leq 1 + 2(\text{dist}(p_i, p_j) - \text{dist}((p \to q)_i,(p \to q)_j)) \leq 2.
\]

We give a sketch of a proof of Lemma 16 for an edge-capacitated instance. First construct the edge-capacitated instance \( ((V, E), S, \bar{u}, \bar{c}, \bar{a}, r) \) as above. Then an optimal potential \( \bar{p} \in \mathbb{S}^{n+m} \) is obtained from \( p \) by Proposition 19, and \( e \in E \) is divided into two edges
By Lemma 20 for $e_1$ and $e_2$, there exists an optimal potential $\bar{p}'$ for the edge-uncapacitated instance with $d^A(\bar{p}, \bar{p}') \leq 4$. By halving $\bar{p}'$, a “quarter-integral” optimal potential $p' \in \text{opt}(h)$ is obtained. Lemma 16 is then shown by rounding quarter-integral components to half-integral.

References

1. E. Anshelevich and A. Karagiozova. Terminal backup, 3D matching, and covering cubic graphs. *SIAM J. Comput.*, 40(3):678–708, 2011. doi:10.1137/090752699.
2. A. Bernáth, Y. Kobayashi, and T. Matsuoka. The generalized terminal backup problem. *SIAM J. Discrete Math.*, 29(3):1764–1782, 2015. doi:10.1137/140972858.
3. J. Chalopin, V. Chepoi, H. Hirai, and D. Osajda. Weakly modular graphs and nonpositive curvature. To appear in *Mem. Amer. Math. Soc.*
4. J. Cheriyan, S. Vempala, and A. Vetta. Network design via iterative rounding of setpair relaxations. *Combinatorica*, 26(3):255–275, 2006. doi:10.1007/s00493-006-0016-z.
5. B. V. Cherkassky. A solution of a problem of multicommodity flows in a network. *Ekonomika i Matematicheskie Metody*, 13:143–151, 1977. in Russian.
6. A. E. Feldmann, J. Köhler, and L Sanità. Fast approximation algorithms for the generalized survivable network design problem. In *Proceedings of the 27th International Symposium on Algorithms and Computation*, pages 33:1–33:12, 2016. doi:10.4230/LIPIcs.ISAAC.2016.33.
7. L. Fleischer, K. Jain, and D. P. Williamson. Iterative rounding 2-approximation algorithms for minimum-cost vertex connectivity problems. *J. Comput. Syst. Sci.*, 72(5):838–867, 2006. doi:10.1016/j.jcss.2005.05.006.
8. T. Fukunaga. Approximating the generalized terminal backup problem via half-integral multiflow relaxation. *SIAM J. Discrete Math.*, 30(2):777–800, 2016. doi:10.1137/151004288.
9. N. Garg and J. Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. *SIAM Journal on Computing*, 37(2):630–652, 2007. doi:10.1137/S0097539704446232.
10. A. V. Goldberg and A. V. Karzanov. Scaling methods for finding a maximum free multiflow of minimum cost. *Math. Oper. Res.*, 22(1):90–109, 1997. doi:10.1287/moor.22.1.90.
11. H. Hirai. Half-integrality of node-capacitated multiflows and tree-shaped facility locations on trees. *Math. Program. A*, 137(1):503–530, 2013.
12. H. Hirai. L-extendable functions and a proximity scaling algorithm for minimum cost multiflow problem. *Discrete Optim.*, 18:1–37, 2015. doi:10.1016/j.disopt.2015.07.001.
13. H. Hirai. Discrete convexity and polynomial solvability in minimum 0-extension problems. *Math. Program. A*, 155(1):1–55, 2016. doi:10.1007/s10107-014-0824-7.
14. H. Hirai. A dual descent algorithm for node-capacitated multiflow problems and its applications. *ACM Trans. Algorithms*, 15(1):15:1–15:24, 2018. doi:10.1145/3291531.
15. H. Hirai. L-convexity on graph structures. *J. Oper. Res. Soc. Jap.*, 61(1):71–109, 2018. doi:10.15807/jorsj.61.71.
16. H. Hirai and M. Ikeda. A cost-scaling algorithm for minimum-cost node-capacitated multiflow problem, 2019. arXiv:1908.01599.
17. Y. Iwata, Y. Yamanouchi, and Y. Yoshida. 0/1/all CSPs, half-integral A-path packing, and linear-time FPT algorithms. In *Proceedings of the 59th IEEE Annual Symposium on Foundations of Computer Science*, pages 462–473, 2018. doi:10.1109/FOCS.2018.00051.
18. K. Jain. A factor 2 approximation algorithm for the generalized Steiner network problem. *Combinatorica*, 21(1):39–60, 2001. doi:10.1007/s004930170004.
19. A. V. Karzanov. A minimum cost maximum multiflow problem. In *Combinatorial Methods for Flow Problems*, pages 138–156. Institute for System Studies, Moscow, 1979. in Russian.
20. A. V. Karzanov. Minimum cost multiflows in undirected networks. *Math. Program.*, 66(1):313–325, 1994. doi:10.1007/BF01581152.
L. Lovász. On some connectivity properties of Eulerian graphs. *Acta Math. Acad. Sci. Hung.*, 28(1–2):129–138, 1976. doi:10.1007/BF01902503.

W. Mader. Über die Maximalzahl kreuzungsfreier $H$-Wege. *Archiv. Math.*, 31(1):387–402, 1978. doi:10.1007/BF01226465.

K. Murota. *Discrete Convex Analysis*. SIAM, Philadelphia, 2003.

A. Schrijver. *Combinatorial Optimization—Polyhedra and Efficiency*. Springer-Verlag, Berlin, 2003.