Approximating the generalized terminal backup problem
via half-integral multiflow relaxation

Takuro Fukunaga*

Abstract

We consider a network design problem called the generalized terminal backup problem. Whereas earlier work investigated the edge-connectivity constraints only, we consider both edge- and node-connectivity constraints for this problem. A major contribution of this paper is the development of a strongly polynomial-time 4/3-approximation algorithm for the problem. Specifically, we show that a linear programming relaxation of the problem is half-integral, and that the half-integral optimal solution can be rounded to a 4/3-approximate solution. We also prove that the linear programming relaxation of the problem with the edge-connectivity constraints is equivalent to minimizing the cost of half-integral multiflows that satisfy flow demands given from terminals. This observation presents a strongly polynomial-time algorithm for computing a minimum cost half-integral multiflow under flow demand constraints.

1 Introduction

1.1 Generalized terminal backup problem

The network design problem is the problem of constructing a low cost network that satisfies given constraints. It includes many fundamental optimization problems, and has been extensively studied. In this paper, we consider a network design problem called the generalized terminal backup problem, recently introduced by Bernáth and Kobayashi [4].

The generalized terminal backup problem is defined as follows. Let \( \mathbb{Q}_+ \) and \( \mathbb{Z}_+ \) denote the sets of non-negative rational numbers and non-negative integers, respectively. Let \( G = (V,E) \) be an undirected graph with node set \( V \) and edge set \( E \), \( c : E \to \mathbb{Q}_+ \) be an edge cost function, and let \( u : E \to \mathbb{Z}_+ \) be an edge capacity function. A subset \( T \) of \( V \) denotes the terminal node set in which each terminal \( t \) is associated with a connectivity requirement \( r(t) \in \mathbb{Z}_+ \). A solution is a multiple edge set on \( V \) containing at most \( u(e) \) edges parallel to \( e \in E \). The objective is to find a solution \( F \) that minimizes \( \sum_{e \in F} c(e) \) under certain constraints. In Bernáth and Kobayashi [4], the subgraph \((V,F)\) was required to contain \( r(t) \) edge-disjoint paths that connect each \( t \in T \) to other terminals. In addition to these edge-connectivity constraints, we consider node-connectivity constraints, under which the paths must be inner disjoint (i.e., disjoint in edges and nodes in \( V \setminus T \)) rather than edge-disjoint. To avoid confusion, we refer to the problem as edge-connectivity terminal backup when the edge-connectivity constraints are required, and as node-connectivity terminal backup when the node-connectivity constraints are imposed.

*National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, Japan. JST, ERATO, Kawarabayashi Large Graph Project, Japan. Email: takuro@nii.ac.jp
The generalized terminal backup problem models a natural data management situation. Suppose that each terminal represents a data storage server in a network, and \( r(t) \) is the amount of data stored in the server at a terminal \( t \). Backup data must be stored in servers different from that storing the original data. To this end, a sub-network that transfers data stored at one terminal to other terminals is required. We assume that edges can transfer a single unit of data per time unit. Hence, transferring data from terminal \( t \) to other terminals within one time unit requires \( r(t) \) edge-disjoint paths from \( t \) to \( T \setminus \{t\} \), which is represented by the edge-connectivity constraints. When nodes are also capacitated, \( r(t) \) inner-disjoint paths are required; these requirements are met by the node-connectivity constraints.

The generalized terminal backup problem is interesting also from theoretical point of view. When \( r \equiv 1 \), the problem is called the terminal backup problem. Note that there is no difference between the edge- and the node-connectivity constraints when \( r \equiv 1 \). Anshelevich and Karagiozova [1] demonstrated that the terminal backup problem is reducible to the simplex matching problem, which is solvable in polynomial time. On the other hand, when \( T = V \), the generalized terminal backup problem is equivalent to the capacitated \( b \)-edge cover problem with degree lower bound \( b(v) = r(v) \) for \( v \in V \). Since the capacitated \( b \)-edge cover problem admits a polynomial-time algorithm, the generalized terminal backup problem is solvable in polynomial time also when \( T = V \). Therefore, we may naturally ask whether the generalized terminal backup problem is solvable in polynomial time. Bernáth and Kobayashi [4] proposed a polynomial-time algorithm for the uncapacitated case (i.e., \( u(e) = +\infty \) for each \( e \in E \)) in the edge-connectivity terminal backup. Their result partially answers the above question, but their assumptions may overly stringent in some situations; that is, their algorithm admits unfavorable solutions that select too many copies of a cheap edge. Moreover, their algorithm cannot deal with the node-connectivity constraints. Unfortunately, when the edge-capacities are bounded or node-connectivity constraints imposed, we do not know whether the generalized terminal backup problem is NP-hard or admits a polynomial-time algorithm. Instead, we propose approximation algorithms.

**Theorem 1.** There exist a strongly polynomial-time \( 4/3 \)-approximation algorithm for the general-
ized terminal backup problem.

The present study contributes two major advances to the generalized terminal backup problem.

- Bernáth and Kobayashi [4] discussed the generalized terminal backup problem in the uncapacitated setting with edge-connectivity constraints, noting that the problem in the capacitated setting is open. Here, we discuss the capacitated setting, and introduce the node-connectivity constraints.

- The generalized terminal backup problem can be formulated as the problem of covering skew supermodular biset functions, which is known to admit a 2-approximation algorithm. On the other hand, as stated in Theorem 1, we develop \( 4/3 \)-approximation algorithms, that outperform this 2-approximation algorithm.

Let us explain the second advance more specifically. Given an edge set \( F \) and a nonempty subset \( X \) of \( V \), let \( \delta_F(X) \) denote the set of edges in \( F \) with one end node in \( X \) and the other in \( V \setminus X \). Let \( f^\lambda: 2^V \to \mathbb{Z}_+ \) be a function such that \( f^\lambda(X) = r(t) \) if \( X \cap T = \{t\} \), and \( f^\lambda(X) = 0 \) otherwise. By the edge-connectivity version of Menger’s theorem, \( F \) satisfies the edge-connectivity
constraints if and only if $|\delta_F(X)| \geq f^\lambda(X)$ for each $X \subset V$. Bernáth and Kobayashi [4] showed that the function $f^\lambda$ is skew supermodular (skew supermodularity is defined in Section 2). For any skew supermodular set function $h$, Jain [10] proposed a seminal 2-approximation algorithm for computing a minimum-cost edge set $F$ satisfying $|\delta_F(X)| \geq h(X)$, $X \subset V$. Although the node-connectivity constraints cannot be captured by set functions as the edge-connectivity constraints, they can be regarded as a request for covering a skew supermodular biset function, to which the 2-approximation algorithm is extended [8] (see Section 2). Therefore, the generalized terminal backup problem admits 2-approximation algorithms, regardless of the imposed connectivity constraints. One of our contributions is to improve these 2-approximations to $4/3$-approximations.

Both of the above 2-approximation algorithms involve iterative rounding of the linear programming (LP) relaxations. Primarily, their performance analyses prove that the value of a variable in each extreme point solution of the LP relaxations is at least $1/2$. Once this property of extreme point solutions is proven, the variables can be repeatedly rounded until a 2-approximate solution is obtained. Our $4/3$-approximation algorithms are based on the same LP relaxations as the iterative rounding algorithms. We show that, in the generalized terminal backup problem, all variables in extreme point solutions of the relaxation take half-integral values. We also prove that the half-integral solution can be rounded into an integer solution with loss of factor at most $4/3$.

It may be helpful for understanding our result to see the well-studied special case of $T = V$ and $u(e) = 1$ for each $e \in E$ (i.e., feasible solutions are simple $r$-edge covers). In this case, our LP relaxation minimizes $\sum_{e \in E} c(e)x(e)$ subject to $\sum_{e \in \delta(v)} x(e) \geq r(v)$ for each $v \in V$ and $0 \leq x(e) \leq 1$ for each $e \in E$, where $\delta(v)$ is the set of edges incident to the node $v$. It has been already known that an extreme point solution of this LP is half-integral, and the edges in $\{e \in E : x(e) = 1/2\}$ form odd cycles. The half-integral variables of the edges on an odd cycle can be rounded as follows. Suppose that edges $e_1, \ldots, e_k$ appears in the cycle in this order, where $k$ is the cycle length (i.e., odd integer larger than one). For each $i, j \in \{1, \ldots, k\}$, we define $x'_i(e_j) = 1$ if $j \geq i$ and $j \equiv i \mod 2$, or if $j < i$ and $j \equiv i + 1 \mod 2$, and $x'_i(e_j) = 0$ otherwise. See Figure 1 for an illustration of this definition. Note that exactly $(k+1)/2$ variables in $x'_1(e_j), \ldots, x'_k(e_j)$ are equal to one, and the other $(k-1)/2$ variables are equal to zero for each $j$. This means that

$$\sum_{i=1}^{k} \sum_{j=1}^{k} c(e_j)x'_i(e_j) = \sum_{j=1}^{k} c(e_j) \cdot \frac{k+1}{2} = (k+1)\sum_{j=1}^{k} c(e_j)x(e_j).$$

Let $i^*$ minimize $\sum_{j=1}^{k} c(e_j)x'_{i^*}(e_j)$ in $i^* \in \{1, \ldots, k\}$. Then, since

$$\sum_{j=1}^{k} c(e_j)x'_i(e_j) \leq \sum_{i=1}^{k} \sum_{j=1}^{k} c(e_j)x'_i(e_j)/k,$$

replacing $x(e_1), \ldots, x(e_k)$ by $x'_{i^*}(e_1), \ldots, x'_{i^*}(e_k)$ increases their costs by a factor at most $(k+1)/k \leq 4/3$. We also observe that the feasibility of the solution is preserved even after the replacement. By applying this rounding for each odd cycle, the half-integral solution can be transformed into a $4/3$-approximate integer solution.

Our result is obtained by extending the characterization of the edge structure whose corresponding variables are not integers, but the extension is not immediate. As in the above special case, those edges form cycles in the generalized terminal backup problem if the solution is a minimal
feasible solution for the LP relaxation. However, the length of a cycle is not necessarily odd, and it
is not clear how the half-integral solution should be rounded; In the above special case, we round
up and down variables of edges on a cycle alternatively, but this obviously does not preserve the
feasibility in the generalized terminal backup problem. The key ingredient in our result is to char-
acterize the relationship between the cycles and the node sets or bisets corresponding to linearly
independent tight constraints in the LP relaxation. We show that a cycle crosses maximal tight
node set or bisets an odd number of times, which extends the property that the length of each cycle
is odd in the special case. Our rounding algorithm decides how to round a non-integer variable
from the direction of the crossing between the corresponding edge and a tight node set or biset.

1.2 Minimum cost multiflow problem

Multiflows are closely related to the generalized terminal backup problem. Among the many mul-
tiflow variants, we focus on the type sometimes called free multiflows. For \( t, t' \in T \), \( A_{t,t'} \) denotes
the set of paths that terminate at \( t \) and \( t' \). Let \( A_t \) denote \( \bigcup_{t' \in T \setminus \{t\}} A_{t,t'} \), and \( A \) denote \( \bigcup_{t \in T} A_t \).\n
**E** (\( A \)) and **V** (\( A \)) denote the sets of edges and nodes in \( A \in A \), respectively. We define a multiflow
as a function \( \psi: A \rightarrow \mathbb{Q}_+ \). In the edge-capacitated setting, an edge capacity \( u(e) \in \mathbb{Z}_+ \) is given, and
we must satisfy \( \sum \{ \psi(A) : A \in A, e \in E(A) \} \leq u(e) \) for each \( e \in E \). In the node-capacitated
setting, a node capacity \( u(v) \in \mathbb{Z}_+ \) is given and \( \sum \{ \psi(A) : A \in A, v \in V(A) \} \leq u(v) \) is required for
each \( v \in V \). The multiflow \( \psi \) is called an integral multiflow if \( \psi(A) \in \mathbb{Z}_+ \) for each \( A \in A \), and is
called a half-integral multiflow if \( 2\psi(A) \in \mathbb{Z}_+ \) for each \( A \in A \). Let \( c(A) \) denote \( \sum_{e \in E(A)} c(e) \) for
\( A \in A \). The cost of \( \psi \) is given by \( \sum_{A \in A} \psi(A)c(A) \).

In the edge-connectivity terminal backup, the connectivity requirement from a terminal \( t \)
equates to requiring that a flow of amount \( r(t) \) can be delivered from \( t \) to \( T \setminus \{t\} \) in the graph \((V,F)\)
with unit edge-capacities if \( F \) is a feasible solution. This condition appears similar to the constraint
that the graph \((V,F)\) with unit edge-capacities admits a multiflow \( \psi \) such that \( \sum_{A \in A_t} \psi(A) \geq r(t) \).
We note that \((V,F)\) with unit edge-capacities admits a multiflow \( \psi \) if and only if the number of
copies of \( e \in E \) in \( F \) is at least \( \sum_{A \in A} \psi(A) \). These observations suggest a correspondence be-
tween the edge-connectivity terminal backup and the problem of finding a minimum cost multiflow.
ψ under the constraint that \( \sum_{A \in \mathcal{A}} \psi(A) \geq r(t) \) for \( t \in T \) in the edge-capacitated setting. We refer to such a multiflow computation as the minimum cost multiflow problem (in the edge-capacitated setting). The same correspondence exists between the node-connectivity terminal backup and the node-capacitated setting in the minimum cost multiflow problem.

However, the generalized terminal backup and the minimum cost multiflow problems are not equivalent. Especially, the minimum cost multiflow problem can be formulated in LP, whereas the generalized terminal backup problem is an integer programming problem. Even if multiflows are restricted to integral multiflows, the two problems are not equivalent. To observe this, let \( G = (V, E) \) be a star with an odd number of leaves. We assume that \( T \) is the set of leaves, and each edge incurs one unit of cost. This star is a feasible solution to the terminal backup problem (i.e., \( r(t) = 1 \) for \( t \in T \)). In contrast, setting \( r \equiv 1 \) and \( u \equiv 1 \) admits no integral multiflow in the edge-capacitated setting, and no feasible (fractional) multiflows in the node-capacitated setting.

Nevertheless, similarities exist between terminal backups and multiflows. As mentioned above, we will show that an LP relaxation of the generalized terminal backup problem always admits a half-integral optimal solution. Similarly, half-integrality results are frequently reported for multiflows. Lovász [13] and Cherkassky [7] investigated \( r \equiv 0 \) in the edge-capacitated setting, and showed that a half-integral multiflow maximizes \( \sum_{A \in \mathcal{A}} \psi(A) \) over all multiflows \( \psi \). Using an identical objective function to ours, Karzanov [12, 11] sought to minimize the cost of multiflows. His feasible multiflow solutions are those attaining \( \max \sum_{A \in \mathcal{A}} \psi(A) \) in the edge-capacitated setting with \( r \equiv 0 \), and he showed that the minimum cost is achieved by a half-integral multiflow. Babenko and Karzanov [2] and Hirai [9] extended Karzanov’s result to node-cost minimization in the node-capacitated setting. In this scenario also, the optimal multiflow is half-integral.

In the present paper, we present a useful relationship between the generalized terminal backup problem and the minimum cost multiflow problem in the edge-capacitated setting. We prove that the optimal solution of the LP used to approximate the edge-connectivity terminal backup is a half-integral multiflow, which also optimizes the minimum cost multiflow problem. Thereby, we can compute the minimum cost half-integral multiflow by solving the LP relaxation. This result is summarized in the following theorem.

**Theorem 2.** The minimum cost multiflow problem admits a half-integral optimal solution in the edge-capacitated setting, which can be computed in strongly polynomial time.

In contrast, we find no useful relationship between the node-connectivity terminal backup and the node-capacitated setting of the minimum cost multiflow problem. We can only show that the LP relaxation of the node-connectivity terminal backup also has an optimal solution which is a half-integral multiflow in the edge-capacitated setting.

Despite its natural formulation, the minimum cost multiflow problem has not been previously investigated to our knowledge. We emphasize that Theorem 2 cannot be derived from previously known results on multiflows. The minimum cost multiflow problem may be solvable by reducing it to minimum cost maximum multiflow problems that (as mentioned above) admit polynomial-time algorithms. A naive reduction can be implemented as follows. Let \( \psi^* \) be a minimum cost multiflow that satisfies the flow demands from terminals, and let \( \nu(t) = \sum_{A \in \mathcal{A}} \psi^*(A) \) for each \( t \in T \). For each \( t \in T \), we add a new node \( t' \) and connect \( t \) and \( t' \) by a new edge of capacity \( \nu(t) \). The new terminal set \( T' \) is defined as \( \{t': t \in T\} \). Now the multiflow \( \psi^* \) can be extended to the multiflow of
maximum flow value for the terminal set $T'$. Applying the algorithm in [12] to this new instance, we can solve the original problem. Moreover, if $\nu(t)$ is an integer for each $t \in T$, this reduction together with the half-integrality result in [11, 12] implies that an optimal multiflow in the minimum cost multiflow problem is half-integral. However, this naive reduction has two limitations. First, $\nu(t)$ is indeterminable without computing $\psi^*$. We only known that $\nu(t)$ cannot be smaller than $r(t)$. Second, we cannot ascertain that $\nu(t)$ is always an integer for each $t \in T$. Hence, this naive reduction seems to yield neither a polynomial-time algorithm nor the half-integrality of optimal multiflows claimed in Theorem 2.

Applying a structural result in [4] on the generalized terminal backup problem, it is easily shown that any integral solution to the edge-connectivity terminal backup provides a half-integral multiflow at the same cost. However, since the way to find an optimal solution for the edge-connectivity terminal backup is unknown, Theorem 2 is not derivable from this relationship. In proving the half-integrality of the LP relaxation required for Theorem 1 we immediately imply the quarter-integrality of a minimum cost multiflow (i.e., $4\psi(A) \in \mathbb{Z}_+$ for each $A \in \mathcal{A}$). The proof of Theorem 2 requires deeper investigation into the structure of half-integrality LP solutions.

1.3 Structure of this paper

Section 2 introduces notations and essential preliminaries on bisets. Section 3 proves that an LP relaxation of the generalized terminal backup problem admits half-integral optimal solutions, and characterizes the edges assigned with half-integral values. Section 4 introduces our $4/3$-approximation algorithm for the generalized terminal backup problem, which proves Theorem 1. Section 5 discusses the relationship between the generalized terminal backup and minimum cost multiflow problems, which presents the proof of Theorem 2. Section 6 concludes the paper.

2 Preliminaries

2.1 Bisets

A biset $\hat{X}$ is defined as an ordered pair $(X, X^+)$ of node sets $X$ and $X^+$ with $X \subseteq X^+ \subseteq V$. The former and latter elements are respectively called the inner part and outer part of the biset. Throughout the paper, we denote the inner part of a biset $\hat{X}$ by $X$, and the outer part by $X^+$. $X^+ \setminus X$ is called the neighbor of $\hat{X}$, and is denoted by $\Gamma(\hat{X})$. $\mathcal{V}$ is the family of all bisets of $V$. For an edge set $F$ and a biset $\hat{X}$, $\delta_F(\hat{X})$ denotes the set of edges in $F$ with one end node in $X$ and the other in $V \setminus X^+$. We identify a node $v \in V$ with the biset $(\{v\}, \{v\})$. Thereby $\delta_F(v)$ denotes the set of edges incident to $v$ in $F$. For simplicity, we write $\delta_F(\hat{X})$ as $\delta(\hat{X})$ when the edge set is unambiguously $E$. If an edge $e$ is in $\delta(\hat{X})$, we say that $e$ is incident to $\hat{X}$.

For two bisets $\hat{X}$ and $\hat{Y}$, we define $\hat{X} \cap \hat{Y}$ as $(X \cap Y, X^+ \cap Y^+)$, $\hat{X} \cup \hat{Y}$ as $(X \cup Y, X^+ \cup Y^+)$, and $\hat{X} \setminus \hat{Y}$ as $(X \setminus Y^+, X^+ \setminus Y)$. If $X \subseteq Y$ and $X^+ \subseteq Y^+$, then we write $\hat{X} \subseteq \hat{Y}$. This inclusion relationship defines a partial order on the bisets, from which we define the maximality and minimality among the bisets.

We say that $\hat{X}$ and $\hat{Y}$ are strongly disjoint when $X \cap Y^+ = \emptyset = X^+ \cap Y$. $\hat{X}$ and $\hat{Y}$ are called noncrossing when strongly disjoint, $\hat{X} \subseteq \hat{Y}$, or when $\hat{Y} \subseteq \hat{X}$. Otherwise, $\hat{X}$ and $\hat{Y}$ are called crossing. A family of bisets is called laminar if each pair of bisets in the family is noncrossing.
The laminarity naturally defines a child-parent relationship among bisets (or a forest structure on bisets). Let \( \mathcal{L} \) be a laminar family of bisets. If \( \hat{X}, \hat{Y}, \hat{Z} \in \mathcal{L} \) satisfy \( \hat{X} \subseteq \hat{Y} \) and \( \hat{X} \subseteq \hat{Z} \), laminarity implies that \( \hat{Y} \subseteq \hat{Z} \) or \( \hat{Z} \subseteq \hat{Y} \). Hence, each \( \hat{X} \in \mathcal{L} \) admits a unique minimal biset \( \hat{Y} \in \mathcal{L} \) with \( \hat{X} \subseteq \hat{Y} \) unless \( \hat{X} \) is maximal in \( \mathcal{L} \). Such a biset \( \hat{Y} \) is defined as the parent of \( \hat{X} \), and \( \hat{X} \) is a child of \( \hat{Y} \). This child-parent relationship naturally leads to terminologies such as “ancestor” and “descendant.” For a biset \( \hat{Y} \) in a laminar family \( \mathcal{L} \) and an edge set \( F \), we let \( F^+_{\hat{Y}}(\hat{Y}) \) and \( F^-_{\hat{Y}}(\hat{Y}) \) respectively denote \( \delta_F(\hat{Y}) \setminus (\bigcup_{X \in \mathcal{X}} \delta_F(X)) \) and \( (\bigcup_{X \in \mathcal{X}} \delta_F(X)) \setminus \delta_F(\hat{Y}) \), where \( \mathcal{X} \) denotes the set of children of \( \hat{Y} \) in \( \mathcal{L} \). If \( \hat{Y} \) has no child, \( F^+_{\hat{Y}}(\hat{Y}) = \delta_F(\hat{Y}) \) and \( F^-_{\hat{Y}}(\hat{Y}) = \emptyset \).

### 2.2 Bisets and connectivity of graphs

For \( t \in T \), let \( \mathcal{C}(t) = \{ \hat{X} \in \mathcal{V} : X \cap T = X^+ \cap T = \{ t \} \} \). We denote \( \bigcup_{t \in T} \mathcal{C}(t) \) by \( \mathcal{C} \). For a vector \( x \in \mathcal{Q}_{\mathcal{H}}^E \) and \( E' \subseteq E \), let \( x(E') \) represent \( \sum_{e \in E'} x(e) \). We define a biset function \( f^\kappa \) by

\[
f^\kappa(\hat{X}) = \begin{cases} r(t) - |\Gamma(\hat{X})|, & \text{if } \hat{X} \in \mathcal{C}(t) \text{ for some } t \in T, \\ 0, & \text{otherwise} \end{cases}
\]

for each \( \hat{X} \in \mathcal{V} \). According to the node-connectivity version of Menger’s theorem, the graph \( (V, F) \) contains \( r(t) \) inner-disjoint paths between \( t \) and \( T \setminus \{ t \} \) if and only if \( |\delta_F(\hat{X})| + |\Gamma(\hat{X})| \geq r(t) \) for each \( \hat{X} \in \mathcal{C}(t) \). This condition is equivalent to \( |\delta_F(\hat{X})| \geq f^\kappa(\hat{X}) \) for all \( \hat{X} \in \mathcal{V} \).

In Section 1, we defined the set function \( f^\lambda \) representing the edge-connectivity constraints. For treating both node-connectivity and edge-connectivity simultaneously, we sometimes extend \( f^\lambda \) to a biset function by identifying \( X \subseteq V \) with the biset \( (X, X) \). Specifically, the biset function \( f^\lambda \) is defined by

\[
f^\lambda(\hat{X}) = \begin{cases} r(t), & \text{if } t \in T, \hat{X} \in \mathcal{C}(t), \Gamma(\hat{X}) = \emptyset, \\ 0, & \text{otherwise} \end{cases}
\]

for each \( \hat{X} \in \mathcal{V} \).

Given a biset function \( h \) and an edge-capacity function \( u : E \rightarrow \mathbb{Z}_+ \), we define \( P(h, u) \) as the set of \( x \in \mathcal{Q}_{\mathcal{H}}^E \) such that

\[
x(\delta(\hat{X})) \geq h(\hat{X}) \quad \text{for } \hat{X} \in \mathcal{V} \tag{1}
\]

and

\[
x(e) \leq u(e) \quad \text{for } e \in E.
\]

Let \( F \) be a multiset of edges in \( E \), and \( \chi_F \) denote the characteristic vector of \( F \) (i.e., \( \chi_F \in \mathbb{Z}_+^E \) and \( F \) contains \( \chi_F(e) \) copies of \( e \) for each \( e \in E \)). Note that \( |\delta_F(\hat{X})| = \chi_F(\delta(\hat{X})) \) for \( \hat{X} \in \mathcal{V} \). Hence, \( \chi_F \in P(f^\kappa, u) \) if and only if \( F \) is a feasible solution to the node-connectivity terminal backup. Similarly, \( \chi_F \in P(f^\lambda, u) \) if and only if \( F \) is a feasible solution to the edge-connectivity terminal backup. These statements imply that the following LP relaxes the node-connectivity and the edge-connectivity terminal backups when \( h = f^\kappa \) and \( h = f^\lambda \), respectively:

\[
\text{LP}(h, u) = \min \left\{ \sum_{e \in E} c(e) x(e) : x \in P(h, u) \right\}.
\]
A biset function $h$ is called (positively) skew supermodular when, for any $\hat{X} \in \mathcal{V}$ with $h(\hat{X}) > 0$ and $\hat{Y} \in \mathcal{V}$ with $h(\hat{Y}) > 0$, $h$ satisfies
\[
 h(\hat{X}) + h(\hat{Y}) \leq h(\hat{X} \cap \hat{Y}) + h(\hat{X} \cup \hat{Y}) \quad (2)
\]
or
\[
 h(\hat{X}) + h(\hat{Y}) \leq h(\hat{X} \setminus \hat{Y}) + h(\hat{Y} \setminus \hat{X}). \quad (3)
\]
For any biset function $h$ and a vector $x : E \to \mathbb{Q}_+$, we let $h_x$ denote the biset function such that $h_x(\hat{X}) = h(\hat{X}) - x(\delta(\hat{X}))$ for each $\hat{X} \in \mathcal{V}$. The skew supermodularity of $f^\kappa_x$ was reported by Bernáth and Kobayashi [4]. Here, we prove that $f^\kappa_x$ is also skew supermodular.

**Theorem 3.** The biset function $f^\kappa_x$ is skew supermodular for any $x : E \to \mathbb{Q}_+$.

**Proof.** Let $\hat{X}$ and $\hat{Y}$ be two bisets. $\hat{X}$ and $\hat{Y}$ are known to always satisfy $|\Gamma(\hat{X})| + |\Gamma(\hat{Y})| \geq |\Gamma(\hat{X} \cap \hat{Y})| + |\Gamma(\hat{X} \cup \hat{Y})|, |\Gamma(\hat{X})| + |\Gamma(\hat{Y})| \geq |\Gamma(\hat{X} \setminus \hat{Y})| + |\Gamma(\hat{Y} \setminus \hat{X})|, x(\delta(\hat{X})) + x(\delta(\hat{Y})) \geq x(\delta(\hat{X} \cap \hat{Y})) + x(\delta(\hat{X} \cup \hat{Y}))$, and $x(\delta(\hat{X})) + x(\delta(\hat{Y})) \geq x(\delta(\hat{X} \setminus \hat{Y})) + x(\delta(\hat{Y} \setminus \hat{X}))$. These inequalities can be proven by counting contributions of edges on both sides.

Suppose that $f^\kappa_x(\hat{X}) > 0$ and $f^\kappa_x(\hat{Y}) > 0$. Then $\hat{X}, \hat{Y} \in \mathcal{C}$ if $\hat{X}, \hat{Y} \in \mathcal{C}(t)$ for some $t \in T$, then both $\hat{X} \cap \hat{Y}$ and $\hat{X} \cup \hat{Y}$ belong to $\mathcal{C}(t)$. From this statement and the above inequalities, we have $f^\kappa_x(\hat{X}) + f^\kappa_x(\hat{Y}) \leq f^\kappa_x(\hat{X} \cap \hat{Y}) + f^\kappa_x(\hat{X} \cup \hat{Y})$ in this case. If $\hat{X} \in \mathcal{C}(t)$ and $\hat{Y} \in \mathcal{C}(t')$ for some $t, t' \in T$ with $t \neq t'$, then $\hat{X} \setminus \hat{Y} \in \mathcal{C}(t)$ and $\hat{Y} \setminus \hat{X} \in \mathcal{C}(t')$. In this case, we have $f^\kappa_x(\hat{X}) + f^\kappa_x(\hat{Y}) \leq f^\kappa_x(\hat{X} \setminus \hat{Y}) + f^\kappa_x(\hat{Y} \setminus \hat{X})$. \hfill \(\square\)

### 3 Structure of extreme point solutions

In this section, we present the properties of the extreme points of $P(f^\kappa, u)$ and $P(f^\lambda, u)$. More precisely, we prove that each extreme point of $P(f^\kappa, u)$ and $P(f^\lambda, u)$ is half-integral, and that the edges whose corresponding variables are half-integral are characteristically structured. Note that both $f^\kappa$ and $f^\lambda$ are integer-valued skew supermodular functions, and $f^\kappa(\hat{X}) = f^\lambda(\hat{X}) = 0$ for any $\hat{X} \notin \mathcal{C}$. In the following, we denote an integer-valued skew supermodular function by $h$, and an extreme point of $P(h, u)$ by $x$.

#### 3.1 Half-integrality

Given an edge set $F$ on $\mathcal{V}$ and $\hat{X} \in \mathcal{V}$, let $\eta_{F,\hat{X}}$ denote the characteristic vector of $\delta_F(\hat{X})$, i.e., an $|F|$-dimensional vector whose components are set to 1 if indexed by an edge in $\delta_F(\hat{X})$, and 0 otherwise. The following lemma has been previously proposed [6] [8].

**Lemma 1.** Let $h$ be a skew supermodular biset function, and $x$ be an extreme point of $P(h, u)$. Let $E_0 = \{e \in E : x(e) = 0\}$, $E_1 = \{e \in E : x(e) = u(e)\}$, and $F = E \setminus (E_0 \cup E_1)$. Let $\mathcal{L}$ be an inclusion-wise maximal laminar subfamily of $\{\hat{X} \in \mathcal{V} : x(\delta_F(\hat{X})) = h(\hat{X}) - u(\delta_{E_1}(\hat{X})) > 0\}$ such that the vectors in $\{\eta_{F,\hat{X}} : \hat{X} \in \mathcal{L}\}$ are linearly independent. Then $|F| = |\mathcal{L}|$, and $x$ is a unique vector that satisfies $x(\delta_F(\hat{X})) = h(\hat{X}) - u(\delta_{E_1}(\hat{X})) > 0$ for each $\hat{X} \in \mathcal{L}$, $x(e) = 0$ for each $e \in E_0$, and $x(e) = u(e)$ for each $e \in E_1$.\[8\]
We note that \( \mathcal{L} \) in Lemma 1 can be constructed in a greedy way: initialize \( \mathcal{L} \) to an empty set, and repeatedly add a biset \( \hat{X} \) such that \( x(\delta_F(\hat{X})) = h(\hat{X}) - u(\delta_{E_1}(\hat{X})) > 0 \) and \( \eta_{F,\hat{X}} \) is linearly independent of the characteristic vectors in the current \( \mathcal{L} \). Hereafter, we assume that \( \mathcal{L} \) is constructed as claimed in Lemma 1. Similarly, \( L \) is linearly independent of the characteristic vectors in the current \( \mathcal{L} \).

Let \( \bar{x} : E \rightarrow \mathbb{Z}_+ \), and define a biset function \( h_{\bar{x}}(\hat{X}) = h(\hat{X}) - \bar{x}(\delta(\hat{X})) \) for \( \hat{X} \in \mathcal{V} \). Let \( \mathbf{1} \) denote the \(|E|\)-dimensional all-one vector. The following lemma relates only to the extreme points of \( P(h_{\bar{x}}, \mathbf{1}) \). In Corollary 1 we will show that this is sufficient for proving the half-integrality of \( P(h, u) \). If \( h(\hat{X}) > 0 \) holds only for \( X \in \mathcal{C} \), we have \( \mathcal{L} \subseteq \mathcal{C} \). In this case, no biset in \( \mathcal{L} \) has more than one child, and \( x \) is characterized as follows.

**Lemma 2.** Suppose that \( h \) is an integer-valued skew supermodular biset function such that \( h(\hat{X}) > 0 \) only for \( \hat{X} \in \mathcal{C} \). Let \( \bar{x} : E \rightarrow \mathbb{Z}_+ \), and let \( x \) be an extreme point of \( P(h_{\bar{x}}, \mathbf{1}) \). Let \( F \) denote \( \{e \in E : 0 < x(e) < 1\} \). Then the following conditions hold:

1. \( |F^+_L(\hat{X})| + |F^-_L(\hat{X})| = 2 \) for each \( \hat{X} \in \mathcal{L} \);
2. If \( e \in F \) is incident to a maximal biset in \( \mathcal{L} \), then it is incident to exactly two maximal bisets in \( \mathcal{L} \);
3. \( x(e) = 1/2 \) for each \( e \in F \).

**Proof.** We first prove (i) and (ii) by contradiction. Let us assume that not all of the above conditions hold. For each pair of \( e \in F \) and its end node \( v \), we distribute a token to a biset in \( \mathcal{L} \). The biset that obtains the token corresponding to \((e, v)\) is decided as follows:

- If there exist one or more bisets \( \hat{X} \in \mathcal{L} \) such that \( e \in \delta_F(\hat{X}) \) and \( v \in X \), the token is assigned to the minimal of these bisets.
- Otherwise, the token is assigned to the minimal biset \( \hat{Y} \) that includes both end nodes of \( e \) in its outer part (if such a biset exists). Notice that such a minimal biset is unique because \( \mathcal{L} \) is laminar and \( e \) is incident to at least one biset in \( \mathcal{L} \).

The total number of tokens is at most \( 2|F| \). In the following, we prove that tokens may be rearranged so that each biset in \( \mathcal{L} \) receives at least two tokens and at least one biset receives three tokens. This rearrangement implies that the number of tokens exceeds \( 2|\mathcal{L}| \), contradicting our requirement that \(|\mathcal{L}| = |F|\).

Recall that \( E_1 = \{e \in E : x(e) = 1\} \). Let \( \bar{x}' \) denote \( \bar{x} + \chi_{E_1} \), and let \( \hat{X} \) be a minimal biset in \( \mathcal{L} \). The minimality of \( \hat{X} \) implies \( F^-_L(\hat{X}) = \emptyset \) and \( F^+_L(\hat{X}) = \delta_F(\hat{X}) \). Since \( x(\delta_F(\hat{X})) = h_{\bar{x}'}(\hat{X}) > 0 \) and \( x(e) < 1 \) for each \( e \in \delta_F(\hat{X}) \), we have \( |F^+_L(\hat{X})| = |\delta_F(\hat{X})| \geq 2 \). Since each edge in \( \delta_F(\hat{X}) \) allocates one token to \( \hat{X} \), \( \hat{X} \) obtains at least two tokens. If \( \hat{X} \) violates (i), then \( |F^+_L(\hat{X})| = |\delta_F(\hat{X})| \geq 3 \), and \( \hat{X} \) obtains at least three tokens.

Next, let \( \hat{X} \) be a biset in \( \mathcal{L} \) that admits a child \( \hat{Y} \in \mathcal{L} \). Since \( \eta_{F,\hat{X}} \) and \( \eta_{F,\hat{Y}} \) are linearly independent, \( |F^+_L(\hat{X})| + |F^-_L(\hat{X})| > 0 \). Therefore, if \( h_{\bar{x}'}(\hat{X}) = h_{\bar{x}'}(\hat{Y}) \), then \( |F^+_L(\hat{X})| \geq 1 \) and \( |F^-_L(\hat{X})| \geq 1 \). If \( h_{\bar{x}'}(\hat{X}) > h_{\bar{x}'}(\hat{Y}) \), then \( |F^+_L(\hat{X})| \geq 2 \) because \( x(e) < 1 \), \( e \in F^+_L(\hat{X}) \). Similarly, if \( h_{\bar{x}'}(\hat{X}) < h_{\bar{x}'}(\hat{Y}) \), then \( |F^-_L(\hat{X})| \geq 2 \). In summary, either case yields \( |F^+_L(\hat{X})| + |F^-_L(\hat{X})| \geq 2 \). Since \( \hat{X} \) receives a token from each edge in \( F^+_L(\hat{X}) \cup F^-_L(\hat{X}) \), it obtains at least two tokens and at least three tokens if condition (i) is violated.
Extending the above discussion, each biset in $\mathcal{L}$ obtains at least two tokens, implying that the number of tokens is at least $2|\mathcal{L}|$. If (i) is violated for any biset in $\mathcal{L}$, that biset receives more than two tokens. Now suppose that (ii) is violated. Then there exists an edge $e \in F$ incident to exactly one maximal biset $X$ in $\mathcal{L}$. The relation $e \in \delta_F(X)$ indicates that $e$ has an end node $v \in V \setminus X^+$, and the token corresponding to $(e, v)$ is assigned to no biset in $\mathcal{L}$. Therefore, if either (i) or (ii) is violated, the number of tokens exceeds the required $2|\mathcal{L}|$

Let $y \in \mathbb{Q}_+^E$ be a vector with components $y(e) = 1/2$ for each $e \in F$, and $y(e) = x(e)$ for each $e \in E \setminus F$. Let $X \in \mathcal{L}$, and denote the child of $X$ (if it exists) by $Y$. From the above discussion, we obtain the following statements:

- $h_{\gamma'}(X) = 1$ and $|\delta_F(X)| = 2$ if $X$ is minimal;
- $|F_\infty^+(X)| = |F_\infty^-(X)| = 1$ if $X$ is not minimal and $h_{\gamma'}(X) = h_{\gamma'}(Y)$;
- $|F_\infty^+(X)| = 2$, $|F_\infty^-(Y)| = 0$ and $h_{\gamma'}(X) = h_{\gamma'}(Y) + 1$ if $X$ is not minimal and $h_{\gamma'}(X) > h_{\gamma'}(Y)$;
- $|F_\infty^+(X)| = 0$, $|F_\infty^-(Y)| = 2$, and $h_{\gamma'}(X) + 1 = h_{\gamma'}(Y)$ if $X$ is not minimal and $h_{\gamma'}(X) < h_{\gamma'}(Y)$.

Therefore, $y$ satisfies $y(\delta(X)) = h_{\gamma'}(X)$ for each $X \in \mathcal{L}$. Since this condition is also uniquely satisfied by vector $x$, we have $x = y$, which proves (iii).

\begin{corollary}
Suppose that $h$ is a skew supermodular biset function such that $h(X) > 0$ only if $X \in \mathcal{L}$. Let $u: E \to \mathbb{Z}_+$. Given $x \in P(h, u)$, we define $\bar{x}: E \to \mathbb{Z}_+$ and $x': E \to \mathbb{Q}_+$ by $\bar{x}(e) = |x(e)|$ and $x'(e) = x(e) - \bar{x}(e)$, respectively for each $e \in E$. If $x$ is an extreme point of $P(h, u)$, then $x'$ is an extreme point of $P(h_{\bar{x}}, 1)$. Moreover, $P(h, u)$ is half-integral if $h$ is integer-valued.
\end{corollary}

\begin{proof}
Note that $0 \leq x'(e) < 1$ for $e \in E$ and $x'(\delta(X)) = x(\delta(X)) - \bar{x}(\delta(X)) \geq h(X) - \bar{x}(\delta(X)) = h_{\bar{x}}(X)$ for $X \in \mathcal{V}$. Hence, $x' \in P(h_{\bar{x}}, 1)$. In the following, we show that $x'$ is an extreme point of $P(h_{\bar{x}}, 1)$ if $x$ is an extreme point of $P(h, u)$. This proves that $x$ is half-integral because $P(h_{\bar{x}}, 1)$ is half-integral by Lemma 2.

If $x'$ is not an extreme point of $P(h_{\bar{x}}, 1)$, there exist $y, y' \in P(h_{\bar{x}}, 1)$ and a real number $\alpha$ such that $x' = \alpha y + (1 - \alpha)y'$ and $0 < \alpha < 1$. Then, $x = x' + \bar{x} = \alpha(y + \bar{x}) + (1 - \alpha)(y' + \bar{x})$. Note that both of $y + \bar{x}$ and $y' + \bar{x}$ are contained in $P(h, u)$, implying that $x$ is not an extreme point of $P(h, u)$.
\end{proof}

\subsection{Path decompositions of extreme point solutions}

We denote \{ $X \in \mathcal{L}; t \in X$ \} by $\mathcal{L}(t)$ for each $t \in T$. Let $t \in T$ with $\mathcal{L}(t) \neq \emptyset$, and let $X$ be the maximal biset in $\mathcal{L}(t)$. We obtain a graph $G^s[X]$ from $G$ by shrinking all the nodes in $V \setminus X^+$ into a single node $s$. Removing $s$ from $G^s[X]$, we obtain another graph $G[X]$ (i.e., $G[X]$ is the subgraph of $G$ induced by $X^+$). We suppose that each edge $e$ in $G^s[X]$ or in $G[X]$ is capacitated by $x(e)$.

If $h = f^s$, each node $v$ in $G^s[X]$ except $s$ and $t$ has unit capacity; conversely, when $h = f^\lambda$, each node has unbounded capacity. The capacities of $s$ and $t$ are always unbounded. Since all capacities are half-integral, the maximum flow between $s$ and $t$ in $G^s[X]$ can be decomposed into a set of paths $R_{1}, \ldots, R_{2r(t)}$ so that a half unit of flow runs along each of them. We specify $X' \in \mathcal{L}(t)$, and note that each path between $s$ and $t$ passes through an edge in $\delta(X')$ or a node in $\Gamma(X')$. Since
by replacing the subpaths of those in $S$ the amount of flow. $\delta$ is the subpath of $t$.

Suppose that $\epsilon$ permits two cases: (i) the existence of maximal bisets $\hat{X}$ Lemma 2 implies the existence of another biset $\hat{X}$ such that $\hat{X}$ retains $x$ and does not remain in $X$. Each edge $e$ is assigned the unit capacity or an unbounded capacity if $h = f^\kappa$ or $h = f^\lambda$, respectively. The capacities of the terminals are assumed as unbounded. For each $t'$, the flow quantity is at least $r(t')$ if and only if $x$ satisfies $[\Pi]$. Let $S$ be a path decomposition of the flow, in which each path in $S$ accommodates a half unit of flow. Let $S_t$ be the set of paths in $S$ that contain nodes in $X^+$ (recall that $\hat{X}$ is the maximal biset in $L(t)$). Without loss of generality, we can state the following fact.

**Assumption 1.** Each path in $S_t$ ends at $t$. Moreover, $\{S': S \in S_t\} \subseteq \{R_{t_1}^1, \ldots, R_{t_2}^l\}$, where $S'$ is the subpath of $S$ between $t$ and the nearest node in $V \setminus X$.

Indeed, if Assumption $[\Pi]$ is not satisfied by $S$, we can modify the flow between $t'$ and $T \setminus \{t'\}$ by replacing the subpaths of those in $S_t$ by appropriate paths in $R_{t_1}^1, \ldots, R_{t_2}^l$, without decreasing the amount of flow.

We say that $x$ is minimal in $P(h, u)$ if $x \in P(h, u)$ and no $y \in P(h, u)$ exists such that $x \neq y$ and $x(e) \geq y(e)$ for any $e \in E$. Let edge $e'$ be incident to a node in $X$. If $x$ is minimal in $P(h, u)$, then $x(e') = \max\{|i = 1, \ldots, 2r(t): e' \in E(R_i)|\}/2$; Otherwise, as $x(e)$ is decreased, it would remain in $P(h, u)$.

**Lemma 3.** Suppose that $h = f^\kappa$ or $h = f^\lambda$, and let $x$ be an extreme minimal point in $P(h, u)$. Then $x(\delta(v))$ is an integer for each $v \in V$.

**Proof.** Define $\hat{x}$ and $x'$ from $x$ as in Corollary $[\Pi]$ and define sets $F$ and $L$ for $x'$ and $P(h, 1)$ as in Lemma $[\Pi]$. In other words, $F = \{e : x'(e) = 1/2\}$, and $L$ is a maximal laminar subfamily of $\{X \in V : x'(X) = h_\kappa(X) > 0\}$ (because $x'(e) = 1/2$ for $e \in E$) such that the vectors in $\{\eta_{F, X} : \hat{X} \in L\}$ are linearly independent. It suffices to show that $|\delta_F(v)|$ is even for each $v \in V$.

Let $v$ be a node with $\delta_F(v) \neq \emptyset$. We first observe that $v$ is included by the outer part of some biset in $L$. Let $e \in \delta_F(v)$. There exists some $\hat{X} \in L$ with $e \in \delta_F(\hat{X}')$; otherwise a slight decrease in $x$ retains $x$ in $P(h, u)$. Let $\hat{X}$ be the maximal biset such that $\hat{X}' \subseteq \hat{X} \in L$. If $v \notin X^+$, then (ii) of Lemma $[\Pi]$ implies the existence of another biset $\hat{Y} \in L$ with $e \in \delta_F(\hat{Y})$, where $\hat{Y}$ satisfies $v \in Y^+$. We now prove that $|\delta_F(v)|$ is even. First, we consider the case of $h = f^\kappa$. The laminarity of $L$ permits two cases: (i) the existence of maximal bisets $\hat{X}_1, \ldots, \hat{X}_l$ with $v \in \Gamma(\hat{X}_1) \cap \cdots \cap \Gamma(\hat{X}_l)$, and (ii) the existence of exactly one maximal biset $\hat{X} \in L$ with $v \in X$.

First, we consider the case (i). In the following discussion, we show that an even number of edges in $\delta_F(v)$ remains in $G[\hat{X}_i]$ for each $i \in \{1, \ldots, l\}$. Each edge $e \in \delta_F(v)$ is associated with exactly one biset $\hat{X}_i$ that includes the both end nodes of $e$ in its outer part. $e$ remains in $G[\hat{X}_i]$, and does not remain in $G[\hat{X}_v]$ for any $i' \in \{1, \ldots, l\}$ with $i' \neq i$. Therefore the claim proves that $|\delta_F(v)|$ is even. Denote by $t_i$ the terminal with $\hat{X}_i \in L(t_i)$. Note that $v$ is included in exactly two paths in $\{R_{t_1}^1, \ldots, R_{t_2}^l\}$ (say $R_{t_1}^l$ and $R_{t_2}^l$). $v$ is adjacent to $s$ in $R_{t_1}^l$ and $R_{t_2}^l$. For $j \in \{1, 2\}$, let $e_j$ be the edge that joins $v$ to the neighbor opposite $s$ in $R_{t_j}^l$. If $e_1 = e_2$, then $x(e_1) = 1$, and $v$ has no incident edge in $F$ remaining in $G[\hat{X}_i]$. If $e_1 \neq e_2$, then $x(e_1) = x(e_2) = 1/2$. Among the edges in $F$ remaining in $G[\hat{X}_i]$, these edges alone are incident to $v$. Hence, the number of edges in $F$ remaining in $G[\hat{X}_i]$ is zero or two.
We now discuss case (ii). Let \( t \) be the terminal with \( \hat{X} \in \mathcal{L}(t) \). By laminarity of \( \mathcal{L} \), no biset in \( \mathcal{L} \setminus \mathcal{L}(t) \) includes \( v \) in its outer part. Hence, it suffices to show that an even number of edges in \( \delta_F(v) \) remains in \( G^s[\hat{X}] \). At most two paths in \( R_1^t, \ldots, R_{2r(t)}^t \) pass through \( v \), but if no biset in \( \mathcal{L}(t) \) includes \( v \) in its neighbor, \( v \) may not be used to full capacity. However, each edge in \( \delta(v) \) is used to full capacity by the minimality of \( x \). If \( v \neq t \), then \( x(\delta(v)) = |\{ i : v \in V(R_i^t) \}| \), and \( x(\delta(v)) \) is an integer. If \( v = t \), then \( x(\delta(v)) = r(t) \), and \( x(\delta(v)) \) is again an integer. In either case, \( |\delta_F(v)| \) is even, which completes the proof for \( \hat{h} = f^\kappa \).

The lemma can be similarly proven for \( \hat{h} = f^\lambda \). Case (i) does not occur because \( \Gamma(\hat{X}) = \emptyset \) for each \( \hat{X} \in \mathcal{L} \).

4 4/3-approximation algorithm for the generalized terminal backup problem

In this section, we prove Theorem 1 by presenting a 4/3-approximation algorithm for the generalized terminal backup problem. Our algorithm rounds a half-integral optimal solution to \( \text{LP}(f^\kappa, u) \) or to \( \text{LP}(f^\lambda, u) \) into an integer solution. In the following discussion, \( h \) denotes a skew supermodular function such that \( h(\hat{X}) > 0 \) only when \( \hat{X} \in \mathcal{L} \).

Solving the LP relaxation

We assume a positive optimal value of \( \text{LP}(h, u) \) (the problem is trivial otherwise). We wish to ensure that any optimal solution \( x \) to \( \text{LP}(h, u) \) is minimal in \( \text{LP}(h, u) \). Clearly, this condition holds when \( c(e) > 0 \) for each \( e \in E \). If \( c(e) = 0 \) for some \( e \in E \), the condition is ensured by perturbing \( c \). Since we can restrict our attention to half-integral solutions, it is sufficient to reset \( c(e) \) to a positive number smaller than \( 2/\theta|E| \) for each \( e \) with \( c(e) = 0 \), where \( \theta \) is the maximum denominator of the edge costs.

The number of constraints of \( \text{LP}(h, u) \) is exponential; hence, whether a polynomial-time algorithm exists for solving \( \text{LP}(h, u) \) is unclear. If \( h = f^\kappa \) or \( h = f^\lambda \), the separation is reducible to a maximum flow computation, and \( \text{LP}(h, u) \) can be solved by the ellipsoid method. Alternatively, the constraints can be written in a compact form by introducing flow variables for each terminal, as implemented in Jain [10]. Hence, if \( h = f^\kappa \) or \( h = f^\lambda \), there are two ways of solving \( \text{LP}(h, u) \) in polynomial time. However, Theorem 1 claims a strongly polynomial-time algorithm. All coefficients in the constraints of \( \text{LP}(h, u) \) are one. Accordingly, Tardos’ algorithm [10] computes an optimal solution to \( \text{LP}(h, u) \) in strongly polynomial time, but does not guarantee an extreme point solution.

Our algorithm first finds an optimal solution to \( \text{LP}(h, u) \) by Tardos’ algorithm. The obtained solution is denoted by \( x^* \). Defining \( \bar{x}^* : E \to \mathbb{Z}_+ \) by \( \bar{x}^*(e) = \lfloor x^*(e) \rfloor \) for \( e \in E \), we then compute an extreme point optimal solution \( x \) to \( \text{LP}(h_{\bar{x}^*}, 1) \). \( \bar{x}^* + x \) is not necessarily an extreme point of \( \mathcal{P}(h, u) \), but is a half-integral optimal solution to \( \text{LP}(h, u) \). The following lemma shows that \( x \) can be computed by iterating Tardos’ algorithm.

Lemma 4. An extreme point optimal solution to \( \text{LP}(h_{\bar{x}^*}, 1) \) can be computed in strongly polynomial time.
Proof. As noted above, an optimal solution to LP\((h_{x*}, 1)\) can be computed in strongly polynomial time. Moreover, whether fixing a variable \(x(e)\) to a specific value \(\tau\) increases the optimal value is also testable in strongly polynomial time by solving LP\((h_{x*}, 1)\) with an additional constraint \(x(e) = \tau\). We sequentially test fixing the variables \(x(e)\) to 0 or 1, and if the fix does not increase the optimal value, the variable is set to the fixed value. If \(x(e)\) is not fixed to 0 or 1, it is set to 1/2.

Optimality of the above-constructed solution \(x\) follows from the existence of a half-integral optimal solution (see Lemma 2). We must now prove that the obtained solution \(x\) is an extreme point. If not, \(x\) can be represented by \(\sum_{i=1}^{l} \alpha_i y_i\), where \(l \geq 2\), \(y_1, \ldots, y_l\) are extreme points of \(P(h_{x*}, 1)\), and \(\alpha_1, \ldots, \alpha_l\) are positive real numbers with \(\sum_{i=1}^{l} \alpha_i = 1\). Let \(i \in \{1, \ldots, l\}\). The optimality of \(x\) indicates that \(y_i\) is an optimal solution to LP\((h_{x*}, 1)\). Moreover, \(y_i(e) = x(e)\) holds if \(x(e) \in \{0, 1\}\). Therefore, there exists some \(e \in E\) such that \(x(e) = 1/2\) and \(y_i(e) \in \{0, 1\}\), which contradicts the way of constructing \(x\). \(\square\)

Rounding half-integral solutions to 4/3-approximate solutions

Our algorithm rounds \(x\), the extreme point optimal solution to LP\((h_{x*}, 1)\), to an integer vector \(x' \in P(h_{x*}, 1)\) subject to \(\sum_{e \in E} c(e)x'(e) \leq 4/3 \cdot \sum_{e \in E} c(e)x(e)\). It then outputs \(x^* + x'\).

We now explain how \(x'\) is computed from \(x\). Recall that \(F\) denotes \(\{e \in E : x(e) = 1/2\}\). We call the edges in \(F\) half-integral edges. By Lemma 3, \(|\delta_F(v)|\) is even for each \(v \in V\) because \(x^* + x\) is minimal in \(P(h, u)\). Therefore, \(F\) can be decomposed into a set of cycles.

Let \(H\) be a cycle in the decomposition. Starting from an arbitrary node \(v_0\) on \(H\), we traverse \(H\). We say that a terminal \(t\) appears when we traverse an edge incident to both bisets \(\bar{X} \in L(t)\) and \(X' \in L(t')\), in the direction from the end node in \(X'\) to the one in \(X\). Let \(t_1, \ldots, t_{k+1}\) be the sequence of terminals, where \(t_1\) is defined such that a biset in \(L(t_1)\) includes \(v_0\) in its inner part, terminal \(t_{i+1}\) appears immediately after \(t_i\) for each \(i = 1, \ldots, k\). \(t_{k+1}\) and \(t_1\) indicate the same terminal. Note that a terminal can appear more than once; hence, \(t_i\) and \(t_j\) may indicate the same terminal even if \(i \neq j\) (unless \(j \in \{i - 1, i + 1\}\)). For \(i \in \{1, \ldots, k\}\), let \(H_i\) be the subpath of \(H\) comprising edges traversed between the appearances of \(t_i\) and \(t_{i+1}\), where \(H_i\) and \(H_{i+1}\) share an edge \(e\) incident to bisets in \(L(t_i)\) and \(L(t_{i+1})\). We also define \(H_{k+1}\) as the subpath of \(H\) comprising edges traversed after the appearance of \(t_{k+1}\). Here, we abuse the notation of \(t_i\)'s as in the case when each terminal appears once only on \(H\); even if \(t_i\) and \(t_j\) denote the same terminal for some \(i, j\) with \(i < j - 1\), \(H_i\) and \(H_j\) share no common edge.

We label each edge \(e \in E(H)\) by “+” or “−” as follows. Suppose that \(e \in E(H_i)\) for some \(i \in \{2, \ldots, k\}\). Let \(X\) be a biset in \(L(t_i)\) to which \(e\) is incident. We call \(e\) outward with respect to \(t_i\) if it is traversed from the end node in \(X\) to the end node in \(V \setminus X^+\). Otherwise, \(e\) is called inward. The label assigned to \(e\) is defined from the parity of \(i\) and the direction of \(e\) as follows:

- If \(i\) is an odd number and \(e\) is outward with respect to \(t_i\), \(e\) is labeled by “+.”
- If \(i\) is an odd number and \(e\) is inward with respect to \(t_i\), \(e\) is labeled by “−.”
- If \(i\) is an even number and \(e\) is outward with respect to \(t_i\), \(e\) is labeled by “−.”
- If \(i\) is an even number and \(e\) is outward with respect to \(t_i\), \(e\) is labeled by “+.”
Figure 2: An example of a cycle of half-integral edges and labels assigned to the edges. Edges drawn by solid and dashed lines are assigned “+” and “−,” respectively. The edges are oriented in the direction of traverse. The areas surrounded by thin solid lines represent the outer parts of bisets in \( L \), and gray areas indicate their neighbors. In this figure, neighbors of bisets in \( L \) are disjoint for visibility, but neighbors can overlap in the general case.

Note that this assignment is consistent; if \( e \) is included in both \( H_i \) and \( H_{i+1} \), then \( e \) is outward with respect to \( t_i \) and inward with respect to \( t_{i+1} \), or vice versa.

If \( e \) is included in \( H_1 \) or \( H_{k+1} \), it is labeled by “+.” If \( e \) is included in both \( H_k \) and \( H_{k+1} \) and \( k \) is even, an inconsistency occurs because \( e \) becomes outward with respect to \( t_k \), and \( e \) is then labeled by “−” by the above rules. The following lemma shows that this situation does not occur.

**Lemma 5.** A cycle such that \( k \) is one or an even number does not exist.

**Proof.** Suppose that \( k \) is one or an even number for a cycle \( H \). Following the above rules, let us assign labels to each edge in \( H \). We also assign labels to those in \( H_1 \) and \( H_{k+1} \) following the rules. Then, for each \( X_1 \in L \), exactly half of the half-integral edges in \( \delta H(X) \) are labeled by “+.”

Let \( \epsilon \) be a constant. For each edge \( e \) in \( H \), if \( e \) is labeled by “+”, the corresponding variable \( x(e) \) is updated to \( x(e) + \epsilon \); otherwise, it is updated to \( x(e) - \epsilon \). Let \( x_\epsilon \) denote the vector after the update. The number of labels assigned indicates that both \( x_{\epsilon} \) and \( x_{-\epsilon} \) belong to \( P(h_{x^*}, 1) \) for a sufficiently small positive number \( \epsilon \), contradicting that \( x \) is an extreme point of \( P(h_{x^*}, 1) \).

Figure 2 illustrates a cycle of half-integral edges and the labels assigned to its edges. In this example, \( k = 5 \), and \( t_3 \) and \( t_5 \) indicate the same terminal.

Our algorithm rounds \( x(e) \) up to 1 if labeled by “+,” and down to 0 if labeled by “−.” The label assignment depends on the choice of \( t_1 \); Another assignment would result if \( H \) is traversed from a node included in the inner part of some biset in \( L(t_i) \) for some \( 1 < i \leq k \) (even if \( t_1 \) and \( t_i \) indicate the same terminal). Hence, we have \( k \) ways of assigning labels to edges in \( H \), all of which allow a feasible rounding of \( x \). However, to achieve a 4/3-approximation, we must select the best assignment by computing and comparing the cost increases of each choice.

In summary, our algorithm computes an integer vector \( x' \) from \( x \) as follows. For each cycle \( H \) of half-integral edges, the algorithm selects the best choice from \( t_1, \ldots, t_k \), and accordingly assigns
labels to each edge. Based on the labels, \( x \) is rounded to obtain the vector \( x' \). Recall that the algorithm outputs \( \bar{x}^* + x' \).

**Performance guarantee**
We first analyze the cost of \( x' \).

**Lemma 6.**
\[
\sum_{e \in E} c(e)x'(e) \leq \frac{4}{3} \sum_{e \in E} c(e)x(e).
\]

**Proof.** Let \( H \) be a cycle of half-integral edges, and \( t_1, \ldots, t_k \) be the sequence of terminals that appear when traversing \( H \). Let \( x_i \) denote the vector obtained by rounding variables corresponding to the edges on \( H \) according to the labels decided by \( t_i \). We note that
\[
\sum_{e \in H} c(e)x'(e) = \min_{1 \leq i \leq k} \sum_{e \in H} c(e)x_i(e) \leq \frac{1}{k} \sum_{i=1}^{k} \sum_{e \in H} c(e)x_i(e).
\]

Recall that \( k \) is an odd number larger than one. In \( k \) assignments, “+” is assigned to an edge \( e \) on \( H \) \((k + 1)/2 \) times. Thus
\[
\sum_{i=1}^{k} \sum_{e \in H} c(e)x_i(e) = \frac{k + 1}{2} \sum_{e \in H} c(e).
\]

Note that \( \sum_{e \in H} c(e)x(e) = \sum_{e \in H} c(e)/2 \). Therefore,
\[
\frac{\sum_{e \in H} c(e)x'(e)}{\sum_{e \in H} c(e)x(e)} \leq \frac{k + 1}{k} \leq \frac{4}{3},
\]
where the last inequality follows from \( k \geq 3 \).

**Lemma 7.** \( x' \in P(h_{\bar{x}^*}, 1) \) when \( h = f^* \) or \( h = f^\lambda \).

**Proof.** Consider the case of \( h = f^* \). Assume that nodes in \( V \setminus T \) have unit capacity and nodes in \( T \) have unbounded capacity. We also regard \( \bar{x}^* + x \) and \( \bar{x}^* + x' \) as edge capacities. To prove that \( x' \in P(h_{\bar{x}^*}, 1) \), it suffices to show that, for each \( t \in T \), the graph capacitated by \( \bar{x}^* + x' \) admits a flow of amount \( r(t) \) between \( t \) and \( T \setminus \{t\} \).

Now consider a maximum flow between \( t \) and \( T \setminus \{t\} \) in the graph capacitated by \( \bar{x}^* + x \). Suppose that the maximum flow is decomposed into a set \( S \) of paths, each running a half unit of flow from \( t \) to another terminal. Since \( x \) satisfies \( x(\delta(X)) \geq f_{\bar{x}^*}(\hat{X}) \) for each \( \hat{X} \in \mathcal{V} \), the flow amount is at least \( r(t) \) (i.e., \( |S| \geq 2r(t) \)). Recall that we are assuming Assumption \( \square \). We now modify \( S \) to satisfy the capacity constraints when the capacity of \( e \in E \) is changed from \( \bar{x}^*(e) + x(e) \) to \( \bar{x}^*(e) + x'(e) \). In the following, we assume that \( x' \) is obtained by rounding variables corresponding to the half-integral edges in a cycle \( H \). If required, the modification is repeated for each cycle of half-integral edges. Let \( t_1, \ldots, t_{k+1} \) be the sequence of terminals appearing when \( H \) is traversed, where \( t_1 \) and \( t_{k+1} \) indicate the same terminal. Labels are assigned to the edges on \( H \), assuming that \( H \) is traversed from \( t_1 \).
First, we consider the case of \( t \not\in \{ t_2, t_k \} \). We traverse \( S \in \mathcal{S} \) from \( t \) to the other end. When arriving at an edge \( e \in E(H) \) labeled by \( "-" \), we reroute the flow along \( S \) as follows. Let \( v \) be the end node of \( e \) near to \( t \). By Assumption 1 and the label-assignment rules, \( e \) shares node \( v \) with an edge labeled \( "+" \) on \( H \). Let \( H' \) denote the subpath of \( H \) consisting of \( "+" \)-labeled edges and terminating at \( v \). We follow \( H' \) instead of \( e \). Let \( u \) be the other end node of \( H' \), and let \( e' \) be the edge incident to \( u \) on \( H' \). By Lemma 2 there exists \( \hat{X} \in \mathcal{L} \) with \( u \in X^+ \).

Suppose that \( \hat{X} \in \mathcal{L}(t) \). Let \( \hat{X} \) be the minimal biset such that \( \hat{X} \in \mathcal{L}(t) \) and \( u \in X^+ \), and let \( \hat{Y} \) be the child of \( \hat{X} \). Then, \( e' \in \delta(\hat{Y}) \), and \( u \in X^+ \setminus Y^+ \). Moreover, another half-integral edge \( e'' \in \delta(\hat{Y}) \), labeled \( "-" \), is incident to \( u \). Edge \( e'' \) is included in another path \( S' \in \mathcal{S} \). Let \( t' \) be the terminal such that \( t \neq t' \) and \( S' \in \mathcal{S}' \). After reaching \( u \), we move to \( t' \) along the path \( S' \). In other words, path \( S \) is replaced by the concatenate of \( S[t, v], H' \), and \( S'[u, t'] \). If \( S'[u, t'] \) contains a half-integral edge labeled by \( "-" \), we modified it recursively. These definitions are illustrated in the left panel of Figure 3. Observe that this modification does not violate the capacity constraints when the edges are capacitated by \( \bar{x}^* + x' \); The capacity of each edge on \( H' \) increases by \( 1/2 \), exactly counterbalancing the unused half capacity of each inner node on \( H' \) prior to the modification (since only two half-integral edges are incident to each inner node), and \( S' \) is modified such that \( S'[u, t'] \) is unused.

Next, suppose that \( \hat{X} \in \mathcal{L}(t') \) for some \( t' \) with \( t \neq t' \). It follows from \( t \not\in \{ t_2, t_k \} \) that \( t' \not\in \{ t_1, t_{k+1} \} \). Since \( H' \) enters \( X \) from \( V \setminus X^+ \) when traversed from \( v \) to \( u \), we have \( u \in X \). Assume that \( \hat{X} \) is the minimal among such bisets. Another half-integral edge \( e'' \in \delta(\hat{X}) \), labeled by \( "-" \), is incident to \( u \), and is included in a path in \( \{ R_{t_1}', \ldots, R_{2r(t')}'' \} \). Without loss of generality, we suppose that that \( R_{t_1}'' \) is such a path. After arriving at \( u \), we reach \( t' \) along \( R_{t_1}''[u, t'] \), as shown in the right panel of Figure 3. Again, this modification preserves the capacity constraints. To see this, suppose that another path \( S' \in \mathcal{S} \setminus \{ S \} \) includes \( R_{t_1}'' \). Then, \( S' \) includes a \( "-" \)-labeled edge before reaching \( u \) when traversed from the other end to \( t' \). \( S' \) will be diverted to another route, and half of the
edge and node capacity on $R'_1[u, t']$ will be no longer used. Prior to modification, half of the inner node capacity of $H'$ was unused because the nodes were incident to exactly two half-integral edges.

We next discuss the case of $t \in \{t_2, t_k\}$. Recall that all edges in $H_1$ and $H_{k+1}$ are labeled by “+.” $H_1$ and $H_2$, as well as $H_k$ and $H_{k+1}$, share exactly one edge. We denote these shared edges as $e_1$ and $e_k$, respectively. Let $H''$ be the subpath of $H$ consisting of edges in $H_1$ and $H_{k+1}$, and consider traversing $H''$ from $e_1$ to $e_k$. With respect to $t_1$, $e_1$ and $e_k$ are traversed inward and outward, respectively. If $t = t_2$, we modify each path in $S_{t_1}$ as when each outward-traversed edge in $H''$ is labeled “−,” whereas other edges are labeled “+.” If $t = t_k$, we perform the converse operation, implemented when each outward-traversed edge in $H''$ is labeled “+,” whereas other edges are labeled “−.” The modification when $t = t_2$ is illustrated in Figure 4. The capacity constraints are preserved because no path in $S$ includes $e_k$ when $t = t_2$, and no path in $S$ includes $e_1$ if $t = t_k$ before the modification.

These transformations generate a flow of amount $r(t)$ from $t$ to $T \setminus \{t\}$ in the graph capacitated by $\bar{x}^* + x'$. This indicates that $x' \in P(f^\kappa, 1)$. Assigning unbounded capacity to each node in $V \setminus T$, a similar proof can be derived for $h = f^\lambda$.

Theorem 1 is immediately proven from Lemmas 4, 6, and 7.

5 Relationship between terminal backup and multiflow

In this section, we limit the constraints on the generalized terminal backup problem to the edge-connectivity constraints, unless otherwise stated. Furthermore, our discussion of multiflows assumes that edges alone are capacitated. Let $\mathcal{A}$ denote the set of paths connecting distinct terminals, and assume that the capacity constraints and flow demands are satisfied by the multiflow $\psi: \mathcal{A} \to \mathbb{Q}_+$.
i.e., \( \sum_{A \in \mathcal{A}}: e \in E(A) \psi(A) \leq u(e) \) for each \( e \in E \) and \( \sum_{A \in \mathcal{A}}: e \in E(A) \psi(A) \geq r(t) \) for each \( t \in T \). We call a vector (or a function) 1/\( k \)-\textit{fractional} if each entry multiplied by \( k \) is an integer.

In this section, we answer the question: to what extent the edge-connectivity terminal backup differs from the minimum cost multiflow problem in the edge-capacitated setting? The differences are small, as demonstrated below.

\textbf{Lemma 8.} For each 1/\( k \)-\textit{fractional} multiflow, there exists a 1/\( k \)-\textit{fractional} vector of the same cost in \( P(f^\lambda, u) \). For each 1/2\( k \)-\textit{fractional} vector \( x \), where \( x \) is minimal in \( P(f^\lambda, u) \) and \( x(\delta(v)) \) is 1/\( k \)-\textit{fractional} for each \( v \in V \setminus T \), there exists a 1/2\( k \)-\textit{fractional} multiflow \( \psi \) such that \( x(e) = \sum_{A \in \mathcal{A}}: e \in E(A) \psi(A) \).

The former part of Lemma \([8]\) is straightforward to prove; if \( \psi \) is a 1/\( k \)-\textit{fractional} multiflow, then \( x: E \to \mathbb{Q}_+ \) defined by \( x(e) = \sum_{A \in \mathcal{A}}: e \in E(A) \psi(A) \) is 1/\( k \)-\textit{fractional} and belongs to \( P(g, u) \).

To prove the latter part, we use a graph operation called \textit{splitting off}. Let \( e = uv \) and \( e' = u'v \) be two edges incident to the same node \( v \). \textit{Splitting off} \( e \) and \( e' \) replaces both \( e \) and \( e' \) by a new edge \( uu' \). In this section, we regard \( f^\lambda \) as a set function. To avoid confusion, we denote \( f^\lambda \) defined from \( r: T \to \mathbb{Z}_+ \) by \( f^\lambda_r \). Let \( J \) be an edge set on \( V \) such that

\[ |\delta_J(X)| \geq f^\lambda_r(X) \text{ for each } X \in 2^V. \tag{4} \]

We say that a pair of edges in \( J \) incident to the same node is \textit{admissible} (with respect to \( f^\lambda_r \)) when \([4]\) holds after splitting off the edges.

\textbf{Lemma 9.} Let \( J \) be an edge set on \( V \) that satisfies \([4]\), and let \( v \) be a node in \( V \setminus T \) with \( |\delta_J(v)| \neq 3 \). Then \( \delta_J(v) \) includes an admissible pair with respect to \( f^\lambda_r \) or \([4]\) holds even after an edge is removed from \( \delta_J(v) \).

Lemma \([8]\) derives from a theorem in \([14, 3]\), which gave a condition for admissible pairs in a more general setting. Bernáth and Kobayashi \([4]\) proved an almost identical claim when discussing the degree-specified version of the edge-connectivity terminal backup, but did not explicitly specify the condition under which admissible pairs can exist. For completeness, we provide a proof of Lemma \([9]\) in the Appendix.

\textbf{Proof of Lemma \([8]\)} The former part of Lemma \([8]\) has been proven above. Here, we concentrate on the latter part. Since \( x \) is 1/2\( k \)-\textit{fractional}, 2\( kx(e) \in \mathbb{Z}_+ \) for each \( e \in E \). Let \( J \) be the set of 2\( kx(e) \) edges parallel to \( e \) for each \( e \in E \). Since \( x(\delta(X)) \geq f^\lambda_r(X) \) for each \( X \in 2^V \), \( J \) satisfies

\[ |\delta_J(X)| \geq 2k f^\lambda_r(X) = f^\lambda_{2k}(X) \text{ for each } X \in 2^V. \tag{5} \]

Let \( v \in V \setminus T \). Since \( x(\delta(v)) \) is 1/\( k \)-\textit{fractional}, \( |\delta_J(v)| \) is an even integer. By the minimality of \( x \), no edge can be removed from \( \delta_J(v) \) without violating \([5]\). Hence, by Lemma \([9]\) \( \delta_J(v) \) includes an admissible pair with respect to \( f^\lambda_{2k} \). For each \( v \in V \setminus T \), we repeatedly split off admissible pairs of edges incident to \( v \) until no edge is incident to \( v \). The graph at the end of this process is denoted by \((V, J')\). In \( J' \), no edge is incident to nodes in \( V \setminus T \), and at least \( 2kr(t) \) edges join \( t \in T \) to other terminals. An edge joining terminals \( t \) and \( t' \) in \( J' \) is generated by splitting off edges on a path between \( t \) and \( t' \) in \( J \). In other words, edges in \( J' \) correspond to edge-disjoint \( T \)-paths in \( J \). By pushing a 1/2\( k \) unit of flow along each of these \( T \)-paths in \( G \), we obtain the required multiflow.
We see that Theorem 2 follows from Lemma 8 and the properties of $P(f^\lambda, u)$ described in Section 3.

**Proof of Theorem 2.** The former part of Lemma 8 implies that $LP(f^\lambda, u)$ relaxes the minimum cost multiflow problem. As proven in Corollary 1, $LP(f^\lambda, u)$ admits a half-integral optimal solution $x$. This solution can be computed in strongly polynomial time and is guaranteed minimal in $P(f^\lambda, u)$, as shown in Section 4. By Lemma 3, $x(\delta(v))$ is integer-valued for each $v \in V$. Hence, by the latter part of Lemma 8, there exists a half-integral multiflow $\psi$ such that $x(e) = \sum_{A \in A, e \in E(A)} \psi(A)$. Note that $\sum_{e \in E} c(e) x(e) = \sum_{A \in A} c(A) \psi(A)$, and therefore $\psi$ minimizes the cost among all feasible multiflows.

How $\psi$ should be computed from $x$ in strongly polynomial time is unknown. However, because $\sum_{A \in A} \psi(A) = x(\delta(v)), \nu(t) = \sum_{A \in A} \psi(A)$ can be computed for each $t \in T$. Moreover, $\nu(t)$ is an integer for each $t \in T$. Therefore, as explained in Section 1.2, this problem reduces to minimizing the cost of maximum multiflow, for which a strongly polynomial-time algorithm is known [12]. 

Each vector $x \in P(f^\kappa, u)$ belongs to $P(f^\lambda, u)$. Hence, we can show that each minimal extreme point of $P(f^\kappa, u)$ admits a half-integral multiflow of the same cost which is feasible in the edge-capacitated setting. However we cannot relate extreme points of $P(f^\kappa, u)$ to feasible multiflows in the node-capacitated setting as we observed for star graphs in Section 1.2.

6 Conclusion

We have presented $4/3$-approximation algorithms for the generalized terminal backup problem. Our result also implies that the integrality gaps of $LP(f^\kappa, u)$ and $LP(f^\lambda, u)$ are at most $4/3$. These gaps are tight even in the edge cover problem (i.e., $T = V$ and $r \equiv 1$): Consider an instance in which $G$ is a triangle with unit edge costs; The half-integral solution $x$ with $x(e) = 1/2$ for all $e \in E$ is feasible to the LPs, and its cost is $3/2$; On the other hand, any integer solution chooses at least two edges from the triangle; Since the costs of these integer solutions are at least 2, the integrality gap is not smaller than $4/3$ in this instance.

An obvious open problem is whether the generalized terminal backup problem admits polynomial-time exact algorithms or not. It seems hard to obtain such an algorithm by rounding solutions of $LP(f^\kappa, u)$ or $LP(f^\lambda, u)$ because of their integrality gaps. For the capacitated $b$-edge cover problem, an LP relaxation of integrality gap one is known [15]. Although this LP relaxation has an exponential number of constraints, the separation can be done in polynomial time. Hence, solving the LP by the ellipsoid method gives an exact polynomial-time algorithm for the capacitated $b$-edge cover problem. For obtaining an LP-rounding polynomial-time algorithm for the generalized terminal backup problem, we have to extend these LP relaxation and polynomial-time separation algorithm.

Another interesting approach is offered by combinatorial approximation algorithms because it is currently a major open problem to find a combinatorial constant-factor approximation algorithm for the survivable network design problem, for which the Jain’s iterative rounding algorithm [10] achieves $2$-approximation. The survivable network design problem involves more complicated connectivity constraints than the generalized terminal backup problem. Hence, study on combinatorial algorithms for the latter problem may give useful insights for the former problem.
Many problems related to multiflows also remain open. We have shown that an LP solution provides a minimum cost half-integral multiflow that satisfies the flow demand from each terminal in the edge-capacitated setting. However, how the computation should proceed in the node-capacitated setting remains elusive. Computing a minimum cost integral multiflow under the same constraints is yet another problem worth investigating. We note that Burlet and Karzanov [5] solved a similar problem related to integral multiflows in the edge-capacitated setting. Their problem differs from ours in the fact that \( \sum_{A \in A_t} \psi(A) \) is required to match the specified value for each terminal \( t \).

Acknowledgements

This work was partially supported by Japan Society for the Promotion of Science (JSPS), Grants-in-Aid for Young Scientists (B) 25730008. The author thanks Hiroshi Hirai for sharing information on multiflows.

References

[1] E. Anshelevich and A. Karagiozova. Terminal backup, 3D matching, and covering cubic graphs. *SIAM Journal on Computing*, 40(3):678–708, 2011.

[2] M. A. Babenko and A. V. Karzanov. Min-cost multiflows in node-capacitated undirected networks. *Journal of Combinatorial Optimization*, 24(3):202–228, 2012.

[3] A. Bernáth and T. Király. A unifying approach to splitting-off. *Combinatorica*, 32:373–401, 2012.

[4] A. Bernáth and Y. Kobayashi. The generalized terminal backup problem. In *SODA*, pages 1678–1686, 2014.

[5] M. Burlet and A. V. Karzanov. Minimum weight \((T,d)\)-joins and multi-joins. *Discrete Mathematics*, 181(1-3):65–76, 1998.

[6] J. Cheriyan, S. Vempala, and A. Vetta. Network design via iterative rounding of setpair relaxations. *Combinatorica*, 26:255–275, 2006.

[7] B. V. Cherkassky. A solution of a problem on multicommodity flows in a network. *Ekonomika i Matematicheskie Metody*, 13(1):143–151, 1977.

[8] L. Fleischer, K. Jain, and D. P. Williamson. Iterative rounding 2-approximation algorithms for minimum-cost vertex connectivity problems. *Journal of Computer and System Sciences*, 72(5):838–867, 2006.

[9] H. Hirai. Half-integrality of node-capacitated multiflows and tree-shaped facility locations on trees. *Mathematical Programming*, 137(1-2):503–530, 2013.

[10] K. Jain. A factor 2 approximation algorithm for the generalized Steiner network problem. *Combinatorica*, 21(1):39–60, 2001.
A proof of Lemma 9

Since Lemma 9 is trivial when $|\delta_f(v)| \leq 2$, we here suppose that $|\delta_f(v)| \geq 4$. Assuming that no edge in $\delta_f(v)$ can be removed without violating (4), we prove that an admissible pair exists in $\delta_f(v)$.

We denote $V \setminus \{v\}$ by $V'$, $\delta_f(v)$ by $A$, and $J \setminus A$ by $J'$. For each $X \subseteq V'$, define $p(X)$ as \( \max\{f^\lambda_r(X), f^\lambda_r(V \setminus X)\} - |\delta_{J'}(X)| \). Note that $p$ is a symmetric skew supermodular function on $V'$. $J$ satisfies (4) if and only if $|\delta_A(X)| \geq p(X)$ for each $X \subseteq V'$. This assumption implies that each $e \in A$ is incident to some $X \subseteq V'$ such that $|\delta_A(X)| = p(X) > 0$. A pair of $uv, u'v \in J$ is admissible if and only if no subset $X \subseteq V'$ satisfies $u, u' \in X$ and $|\delta_A(X)| - p(X) \leq 1$. We call $X$ a dangerous set if $X \subseteq V'$ and $|\delta_A(X)| - p(X) \leq 1$.

If $X$ is a dangerous set, then $p(X) > 0$, implying that $|X \cap T| = 1$ or $|T \setminus X| = 1$. Without loss of generality, we assume that each $t \in T$ admits $X \subseteq V'$ with $t \in X$ and $p(X) > 0$ (otherwise $t$ can be removed from $T$). We denote $\{X \subseteq V': X \cap T = \{t\}\}$ by $C'(t)$, and the set of $X \in \mathcal{C}'(t)$ attaining $\min_{X \in \mathcal{C}'(t)} |\delta_{J'}(X)|$ by $\mathcal{M}(t)$. Since $\max\{f^\lambda_r(X), f^\lambda_r(V \setminus X)\}$ is identical for all $X \in \mathcal{C}'(t)$, we have $p(Y) > 0$ for each $Y \in \mathcal{M}(t)$. Since $J$ satisfies (4), $|\delta_A(Y)| \geq 1$ for each $Y \in \mathcal{M}(t)$. From the submodularity and posimodularity of graph cut functions, we have $X \cup Y, X \cap Y \in \mathcal{M}(t)$ for any $X, Y \in \mathcal{M}(t)$, and also $X \setminus Y \in \mathcal{M}(t)$ and $Y \setminus X \in \mathcal{M}(t')$ for any $X \in \mathcal{M}(t)$ and $Y \in \mathcal{M}(t')$ with $t \neq t'$. $\mathcal{M}(t)$ has a unique minimal set $Z_t$ and a unique maximal set $W_t$. Moreover, $Z_t \cap Y = \emptyset$ for any $Y \in \mathcal{M}(t')$ with $t' \neq t$.

In previous work [14 18], it was shown that $A$ includes an admissible pair if there exists some $X \subseteq V'$ with $p(X) \geq 2$. Hence, in the following discussion, we assume that $p(X) \leq 1$ for each $X \subseteq V'$. By this assumption, $p(X) = 1$ holds for each $X \in \bigcup_{t \in T} \mathcal{M}(t)$. Moreover, $X$ is a dangerous set if and only if $|\delta_A(X)| = 2$, and $X$ or $V' \setminus X$ belongs to $\bigcup_{t \in T} \mathcal{M}(t)$.

First, let us prove by contradiction that $|T| \geq 4$. For this purpose, we suppose that $|T| \leq 3$. Because $|A| \geq 4$, there exist $t_1 \in T, e_1, e'_1 \in A$, and $X \subseteq V'$ such that $\{e_1\} = \delta_A(Z_{t_1}), \{e'_1\} = \delta_A(X)$, and such that $X \in \mathcal{M}(t_1)$ or $V' \setminus X \in \mathcal{M}(t_1)$. If $X \in \mathcal{M}(t_1)$, then $Z_{t_1} \subseteq X$, which contradicts $\{e'_1\} = \delta_A(X)$. Hence $V' \setminus X \in \mathcal{M}(t_1)$. Note that $V' \setminus W_{t_1} \subseteq V' \setminus X$. Each edge in $A \setminus \{e_1, e'_1\}$ is
incident to $V' \setminus X$. Hence there exist $e_2' \in A \setminus \{e_1, e_1'\}$, $t_2 \in T$, and $Y \subseteq V'$ such that \(\{e_2'\} = \delta_A(Y)\), and $Y \in \mathcal{M}(t_2)$ or $V' \setminus Y \in \mathcal{M}(t_2)$. Existence of $e_1$ and $e_1'$ implies that $Z_{t_1} \not\subseteq Y$ and $V' \setminus Y \not\subseteq W_{t_1}$, respectively. $t_1 \neq t_2$ follows from $Z_{t_1} \not\subseteq Y$ if $Y \in \mathcal{M}(t_2)$, and from $V' \setminus Y \not\subseteq W_{t_1}$ if $V' \setminus Y \in \mathcal{M}(t_2)$. If $e_2'$ is not incident to $Z_{t_2}$, another edge $e_2 \in A \setminus \{e_1, e_1', e_2'\}$ is incident to $Z_{t_2}$. This same edge is also incident to $V' \setminus X$, implying that $Z_{t_2} \cap (V \setminus X) \neq \emptyset$ and contradicting $V \setminus X \in \mathcal{M}(t_1)$. Therefore $|T| \geq 4$.

Let $t_1, t_2 \in T$, $e_1 \in \delta_A(Z_{t_1})$, and $e_2 \in \delta_A(Z_{t_2})$. Suppose that the pair of $e_1$ and $e_2$ is nonadmissible. Then, there exists a dangerous set $Y$ with $\delta_A(Y) = \{e_1, e_2\}$. $Y \in \mathcal{C}(t_3)$ or $V' \setminus Y \in \mathcal{C}(t_3)$ for some $t_3 \in T$. In the former case, if $t_3 \neq t_1$, the existence of $e_1 \in \delta_A(Y) \cap \delta_A(Z_{t_1})$ contradicts $Y \cap Z_{t_1} = \emptyset$, and if $t_3 = t_1$, the existence of $e_2 \in \delta_A(Y) \cap \delta_A(Z_{t_2})$ contradicts $Y \cap Z_{t_2} = \emptyset$. Hence, $V' \setminus Y \in \mathcal{C}(t_3)$. Existence of $e_1$ and $e_2$ implies that $Z_{t_1} \setminus (V' \setminus Y) \neq \emptyset$ and $Z_{t_2} \setminus (V' \setminus Y) \neq \emptyset$. If $t_3 \in \{t_1, t_2\}$, the minimality of $Z_{t_1}$ or $Z_{t_2}$ is violated. Hence, $t_3 \not\in \{t_1, t_2\}$. Now, let $t_4 \in T \setminus \{t_1, t_2, t_3\}$, and $e_4 \in \delta_A(Z_{t_4})$. Since the end node of $e_4$ in $Z_{t_4}$ is included in $V' \setminus Y$, we obtain $(V' \setminus Y) \cap Z_{t_4} \neq \emptyset$, which also presents a contradiction.