Approximating the Generalized Terminal Backup Problem via Half-integral Multiflow Relaxation

Takuro Fukunaga

National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, Japan.
JST, ERATO, Kawarabayashi Large Graph Project, Japan.
`takuro@nii.ac.jp`

─── **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 implies 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 [3].

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, $c\colon E \to \mathbb{Q}_+$ be an edge cost function, and let $u\colon E \to \mathbb{Z}_+$ 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 [3], 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 (removing "general" from the problem names

makes no confusion because distinguishing edge-connectivity and node-connectivity implies that the connectivity requirement of a terminal is larger than one).

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 [3] 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 be 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 generalized terminal backup problem.*

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

- Bernáth and Kobayashi [3] 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 $f: 2^V \to \mathbb{Z}_+$ be a function such that $f(X) = r(t)$ if $X \cap T = \{t\}$, and $f(X) = 0$ otherwise. By the edge-connectivity version of Menger's theorem, F satisfies the edge-connectivity constraints if and only if $|\delta_F(X)| \geq f(X)$ for each $X \in 2^V$. Bernáth and Kobayashi [3] showed that the function f is skew supermodular (skew supermodularity

is defined in Section 2). For any skew supermodular set function h, Jain [9] proposed a seminal 2-approximation algorithm for computing a minimum-cost edge set F satisfying $|\delta_F(X)| \geq h(X)$, $X \in 2^V$. 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 [6]. 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 $\{e \in E \colon x(e) = 1/2\}$ 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 $j \equiv i \bmod 2$, or if $j < i$ and $j \equiv i+1 \bmod 2$, and $x_i'(e_j) = 0$ otherwise. 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 characterize 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 multiflow variants, we focus on the type sometimes called *free multiflows*. For $t, t' \in T$, $\mathcal{A}_{t,t'}$ denotes the set of paths that terminate at t and t'. Let $\mathcal{A}_t$ denote $\bigcup_{t' \in T \setminus \{t\}} \mathcal{A}_{t,t'}$, and $\mathcal{A}$ denote $\bigcup_{t \in T} \mathcal{A}_t$. $E(A)$ and $V(A)$ denote the sets of edges and nodes in $A \in \mathcal{A}$, respectively. We define a multiflow as a function $\psi \colon \mathcal{A} \to \mathbb{Q}_+$. In the edge-capacitated setting, an edge capacity $u(e) \in \mathbb{Z}_+$ is given, and we must satisfy $\sum \{\psi(A) \colon A \in \mathcal{A}, e \in E(A)\} \le u(e)$ for each $e \in E$. In the node-capacitated setting, a node capacity $u(v) \in \mathbb{Z}_+$ is given and $\sum \{\psi(A) \colon A \in \mathcal{A}, v \in V(A)\} \le u(v)$ is required for each $v \in V$. The multiflow ψ is called an *integral* multiflow if $\psi(A) \in \mathbb{Z}_+$ for each $A \in \mathcal{A}$, and is called a *half-integral* multiflow if $2\psi(A) \in \mathbb{Z}_+$ for each $A \in \mathcal{A}$. Let $c(A)$ denote $\sum_{e \in E(A)} c(e)$ for $A \in \mathcal{A}$. The cost of ψ is given by $\sum_{A \in \mathcal{A}} \psi(A) c(A)$.

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 ψ such that $\sum_{A \in \mathcal{A}_t} \psi(A) \ge r(t)$. We note that (V, F) with unit edge-capacities admits a multiflow ψ if and only if the number of copies of $e \in E$ in F is at least $\sum_{A \in \mathcal{A} \colon e \in E(A)} \psi(A)$. These observations suggest a correspondence between the edge-connectivity terminal backup and the problem of finding a minimum cost multiflow ψ under the constraint that $\sum_{A \in \mathcal{A}_t} \psi(A) \ge r(t)$ 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 [12] and Cherkassky [5] 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 ψ. Using an identical objective function to ours, Karzanov [11, 10] 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 [7] 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 ψ^* be a minimum cost multiflow that satisfies the flow demands from terminals, and let $\nu(t) = \sum_{A \in \mathcal{A}_t} \psi^*(A)$ 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 ψ^* can be extended to the multiflow of maximum flow value for the terminal set T'. Applying the algorithm in [11] 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 [10, 11] 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 ψ^*. We only know 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 [3] 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-integral LP solutions.

1.3 Structure of This Paper

In this article, due to the space limitation, we only sketch how our 4/3-approximation algorithm works in the case of $r \equiv 1$. We also omit proofs of several lemmas. We recommend referring to the full version of the present paper for the full description of our results. Section 2 introduces notations and essential preliminary facts. Section 3 presents required properties of extreme point solutions of an LP relaxation to the edge-connectivity terminal backup. Section 4 introduces our 4/3-approximation algorithm, restricted to the case of $r \equiv 1$. Section 5 concludes the paper.

2   Preliminaries

For an edge set F and a node set $X \in 2^V$, let $\delta_F(X)$ denote the set of edges in F with one end node in X and the other in $V \setminus X$. We identify a node $v \in V$ with the node set $\{v\}$. Thereby $\delta_F(v)$ denotes the set of edges incident to v in F. For simplicity, we write $\delta_E(X)$ as $\delta(X)$ when the edge set is unambiguously E. If an edge e is in $\delta(X)$, we say that e is *incident* to X.

We say that $X \in 2^V$ and $Y \in 2^V$ are *noncrossing* when $X \cap Y = \emptyset$, $X \subseteq Y$, or when $Y \subseteq X$. Otherwise, X and Y are called *crossing*. A family of node sets is called *laminar* if each pair of node sets in the family is noncrossing. The laminarity naturally defines a child-parent relationship among those in the family; If $X, Y \in \mathcal{L}$, and if $Y \in \mathcal{L}$ is the minimal node set such that $X \subseteq Y$ in $\mathcal{L}$, then Y is defined as the *parent* of X, and X is a *child* of Y. This child-parent relationship naturally leads to terminologies such as "ancestor" and "descendant." For a node set Y in a laminar family $\mathcal{L}$ and an edge set F, we let $F_{\mathcal{L}}^+(Y)$ and $F_{\mathcal{L}}^-(Y)$ respectively denote $\delta_F(Y) \setminus \left(\bigcup_{X \in \mathcal{X}} \delta_F(X)\right)$ and $\left(\bigcup_{X \in \mathcal{X}} \delta_F(X)\right) \setminus \delta_F(Y)$, where $\mathcal{X}$ denotes the set of children of Y in $\mathcal{L}$. If Y has no child, $F_{\mathcal{L}}^+(Y) = \delta_F(Y)$ and $F_{\mathcal{L}}^-(Y) = \emptyset$.

For $t \in T$, let $\mathcal{C}(t) = \{X \in 2^V : X \cap T = \{t\}\}$. We denote $\bigcup_{t \in T} \mathcal{C}(t)$ by $\mathcal{C}$. For a vector $x \in \mathbb{Q}_+^E$ and $E' \subseteq E$, let $x(E')$ represent $\sum_{e \in E'} x(e)$. Recall that in Section 1, we defined the set function f representing the edge-connectivity constraints by

$$f(X) = \begin{cases} r(t), & \text{if } t \in T, X \in \mathcal{C}(t), \\ 0, & \text{otherwise} \end{cases}$$

for each $X \in 2^V$.

Given a function $h \colon 2^V \to \mathbb{Q}_+$ and an edge-capacity function $u \colon E \to \mathbb{Z}_+$, we define $P(h, u)$ as the set of $x \in \mathbb{Q}_+^E$ such that

$$x(\delta(X)) \geq h(X) \quad \text{for } X \in 2^V \tag{1}$$

and $x(e) \leq u(e)$ for $e \in E$.

Let F be a multiset of edges in E, and χ_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 $|\delta_F(X)| = \chi_F(\delta(X))$ for $X \in 2^V$. Hence, $\chi_F \in P(f, u)$ if and only if F is a feasible solution to the edge-connectivity terminal backup. These statements imply that the LP $\mathsf{LP}(h, u) = \min\left\{\sum_{e \in E} c(e)x(e) \colon x \in P(h, u)\right\}$ relaxes the edge-connectivity terminal backups when $h = f$.

A biset function h is called (*positively*) *skew supermodular* when, for any $X \in 2^V$ with $h(X) > 0$ and $Y \in 2^V$ with $h(Y) > 0$, h satisfies

$$h(X) + h(Y) \leq h(X \cap Y) + h(X \cup Y) \tag{2}$$

or

$$h(X) + h(Y) \leq h(X \setminus Y) + h(Y \setminus X). \tag{3}$$

For any function $h \colon 2^V \to \mathbb{Q}_+$ and a vector $x \colon E \to \mathbb{Q}_+$, we let h_x denote the function such that $h_x(X) = h(X) - x(\delta(X))$ for each $X \in 2^V$. Bernáth and Kobayashi [3] reported that f_x is skew supermodular for any $x \colon E \to \mathbb{Q}_+$.

3   Structure of Extreme Point Solutions

In this section, we present the properties of the extreme points of $P(f, u)$. More precisely, we prove that each extreme point of $P(f, u)$ is half-integral, and that the edges whose

corresponding variables are half-integral are characteristically structured. Note that f is an integer-valued skew supermodular function, and $f(X) = 0$ for any $X \notin \mathcal{C}$.

3.1 Half-Integrality

In the following, we denote an integer-valued skew supermodular function by h, and an extreme point of $P(h, u)$ by x. Given an edge set F on V and a node set $X \in 2^V$, let $\eta_{F,X}$ denote the characteristic vector of $\delta_F(X)$, i.e., an $|F|$-dimensional vector whose components are set to 1 if indexed by an edge in $\delta_F(X)$, and 0 otherwise. The following lemma has been previously proposed [9].

▶ **Lemma 3.** *Let $h: 2^V \to \mathbb{Q}_+$ be a skew supermodular 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 $\{X \in 2^V: x(\delta_F(X)) = h(X) - u(\delta_{E_1}(X)) > 0\}$ such that the vectors in $\{\eta_{F,X}: X \in \mathcal{L}\}$ are linearly independent. Then $|F| = |\mathcal{L}|$, and x is a unique vector that satisfies $x(\delta_F(X)) = h(X) - u(\delta_{E_1}(X)) > 0$ for each $X \in \mathcal{L}$, $x(e) = 0$ for each $e \in E_0$, and $x(e) = u(e)$ for each $e \in E_1$.*

We note that $\mathcal{L}$ in Lemma 3 can be constructed in a greedy way; initialize $\mathcal{L}$ to an empty set, and repeatedly add a biset X such that $x(\delta_F(X)) = h(X) - u(\delta_{E_1}(X)) > 0$ and $\eta_{F,X}$ is linearly independent of the characteristic vectors in the current $\mathcal{L}$. Hereafter, we assume that $\mathcal{L}$ is constructed as claimed in Lemma 3. Similarly, E_0, E_1, and F are defined from x as in Lemma 3.

Let $\bar{x}: E \to \mathbb{Z}_+$, and define a function $h_{\bar{x}}(X)$ as $h(X) - \bar{x}(\delta(X))$ for $X \in 2^V$. 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 5, we will show that this is sufficient for proving the half-integrality of $P(h, u)$. If $h(X) > 0$ holds only for $X \in \mathcal{C}$, we have $\mathcal{L} \subseteq \mathcal{C}$. In this case, no node set in $\mathcal{L}$ has more than one child, and x is characterized as follows.

▶ **Lemma 4.** *Suppose that $h: 2^V \to \mathbb{Z}_+$ is an integer-valued skew supermodular function such that $h(X) > 0$ only for $X \in \mathcal{C}$. Let $\bar{x}: E \to \mathbb{Z}_+$, 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:*
  (i) *$|F_{\mathcal{L}}^+(X)| + |F_{\mathcal{L}}^-(X)| = 2$ for each $X \in \mathcal{L}$;*
  (ii) *If $e \in F$ is incident to a maximal node set in $\mathcal{L}$, then it is incident to exactly two maximal node sets in $\mathcal{L}$;*
  (iii) *$x(e) = 1/2$ for each $e \in F$.*

▶ **Corollary 5.** *Suppose that $h: 2^V \to \mathbb{Q}_+$ is a skew supermodular function such that $h(X) > 0$ only if $X \in \mathcal{C}$. 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 $\bar{x}(e) = \lfloor x(e) \rfloor$ 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 $P(h_{\bar{x}}, \mathbf{1})$. Moreover, $P(h, u)$ is half-integral if h is integer-valued.*

3.2 Path decompositions of extreme point solutions

In this section, we consider $x \in P(f, u)$. 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 node set 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 $x(e)$. 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 $R_1^t, \ldots, R_{2r(t)}^t$ each of which accommodates a half unit of flow.

Let $X' \in \mathcal{L}(t)$. Each path between s and t passes through an edge in $\delta(X')$. Since $x(\delta(X')) = r(t)$, the edges in $\delta(X')$ are used to full capacity by the maximum flow, and each path R_i^t includes exactly one edge in $\delta(X')$.

Suppose that both R_i^t and R_j^t include a node $v \notin \{s, t\}$. Let e_i and e_i' be the edges incident to v on R_i^t, where e_i is near to s than e_i'. We define the edges e_j and e_j' incident to v on R_j^t, similarly. We assume that the following fact holds for any such paths R_i^t and R_j^t.

▶ **Assumption 1.** If $x(e_i)$ is half-integral and $x(e_j)$ is an integer, and if exactly one of $x(e_i')$ and $x(e_j')$ is half-integral, then $x(e_i')$ is half-integral.

Indeed, if Assumption 1 does not hold, then exchanging the subpaths between v and t makes them satisfy it.

In the following discussion, we consider a maximum flow between a terminal t' and $T \setminus \{t'\}$ in G, where t' may equal t. In such a flow, each edge e is capacitated by $x(e)$. The flow quantity is at least $r(t')$ if and only if x satisfies (1) with $h = f$. Let $\mathcal{S}$ be a path decomposition of the flow, in which each path in $\mathcal{S}$ accommodates a half unit of flow. Let $\mathcal{S}_t$ be the set of paths in $\mathcal{S}$ that contain nodes in X (recall that X is the maximal node set in $\mathcal{L}(t)$). Without loss of generality, we can state the following fact.

▶ **Assumption 2.** Each path in $\mathcal{S}_t$ ends at t. Moreover, $\{S' : S \in \mathcal{S}_t\} \subseteq \{R_1^t, \ldots, R_{2r(t)}^t\}$, where S' is the subpath of S between t and the nearest node in $V \setminus X$.

If Assumption 2 is not satisfied by $\mathcal{S}$, we can modify the flow between t' and $T \setminus \{t'\}$ by replacing the subpaths of those in $\mathcal{S}_t$ by appropriate paths in $R_1^t, \ldots, R_{2r(t)}^t$, without decreasing the amount of flow.

We say that x is *minimal in $P(f, u)$* if $x \in P(f, u)$ and no $y \in P(f, u)$ 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 $P(f, u)$, then $x(e') = |\{i = 1, \ldots, 2r(t) : e' \in E(R_i^t)\}|/2$; Otherwise, as $x(e)$ is decreased, it would remain in $P(f, u)$.

▶ **Lemma 6.** *Let x be an extreme minimal point in $P(f, u)$. Then $x(\delta(v))$ is an integer for each $v \in V$.*

4 LP-rounding $4/3$-Approximation Algorithm for Terminal Backup Problem

Our algorithm rounds a half-integral optimal solution to the LP relaxations into an integer solution. Let us assume that a minimal half-integral optimal solution x and a laminar family $\mathcal{L}$ in Lemma 3 are given. In what follows, we explain how to round x.

Let F denote $\{e \in E : x(e) = 1/2\}$. We call the edges in F *half-integral edges*. $|\delta_F(v)|$ is even for each $v \in V$ because $x(\delta(v))$ is an integer by Lemma 6. Hence F can be decomposed into an edge-disjoint set of cycles. Let H be a cycle in the decomposition.

For each $e \in F$, $\mathcal{L}$ contains a node set to which e is incident. Let $\mathcal{L}'$ be the subfamily of $\mathcal{L}$ that consists of the node sets to which edges in H are incident. Since $r \equiv 1$, exactly two edges in H are incident to each node set in $\mathcal{L}'$.

Let $t_1, \ldots, t_k$ be the terminals such that $\mathcal{L}(t_i) \cap \mathcal{L}' \neq \emptyset$ for each $i \in \{1, \ldots, k\}$. We can prove the following lemma.

▶ **Lemma 7.** *k is an odd integer larger than one.*
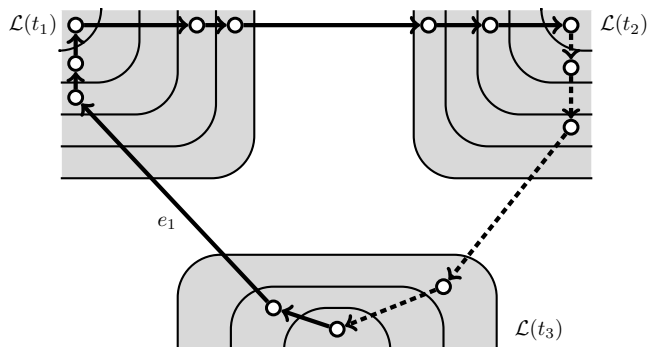
Figure 1 An example of a cycle of half-integral edges and the first assignment of lables to the edges. Edges drawn by solid and dashed lines are assigned "+" and "−," respectively. The edges are oriented in the direction of traverse. The areas surrounded by thin solid lines represent the node sets in $\mathcal{L}$.

For each $i \in \{1, \ldots, k\}$, let X_i denote the maximal node set in $\mathcal{L}(t_i) \cap \mathcal{L}'$, and let H_i be the subpath of H comprising of edges incident to node sets in $\mathcal{L}(t_i) \cap \mathcal{L}'$. Hence, if an edge are incident to both X_i and X_j, the edge is shared by H_i and H_j.

Let $e_1 = uv$ be an edge incident to X_1, where we assume without loss of generality that $u \in X_1$ and $v \notin X_1$. Consider traversing $E(H)$, starting from e_1 in the direction from v to u. We say that t_i *appears* when we traverse an edge incident to two node sets $X_i \in \mathcal{L}(t_i)$ and $X_j \in \mathcal{L}(t_j)$ with $i \neq j$ in the direction from the end node in X_j to the one in X_i. Without loss of generality, we assume that the terminals appear in the increasing order of subscripts. Therefore, during the traverse of H, we first visit edges in H_1, then those in H_2, and so on. Suppose that $X \in \mathcal{L}(t_i)$ and $e \in \delta_H(X)$. We say that e is *outward* with respect to t_i if e is traversed from the end node in X to the other. Otherwise, e is called *inward*. This implies that, during the traverse of H_i, we first traverse edges inward with respect to t_i, and then those outward with respect to t_i.

We define k assignments of labels to the edges in H, where each edge is labeled by either "+" or "−." Let us define the i-th assignment. If $e \in E(H_i)$, then e is labeled by "+." If $e \in E(H_j)$ for some $j \neq i$, then its label is decided by the following rules.

- If $j - i$ is odd and e is outward with respect to t_j, e is labeled by "+."
- If $j - i$ is odd and e is inward with respect to t_j, e is labeled by "−."
- If $j - i$ is even and e is outward with respect to t_j, e is labeled by "−."
- If $j - i$ is even and e is inward with respect to t_j, e is labeled by "+."

Note that this assignment is consistent; if e is included in both H_j and H_{j+1}, then e is outward with respect to t_j and inward with respect to t_{j+1}, and hence e is assigned the same label from j and $j + 1$; e_1 is shared by H_1 and H_k, and similarly it is assigned the same label because k is odd. Figure 1 shows an example of the cycle H, and the first assignment of labels to the edges on H.

Our algorithm rounds $x(e)$ into 1 if e is labeled by "+," and into 0 otherwise. Since we have k assignments of labels, we have k ways of rounding of edges in H. Our algorithm chooses the most cost-effective one among them.

Let us observe that this algorithm achieves 4/3-approximation. First, we prove that the above rounding increases the cost by a factor of at most 4/3. Let x' be the vector obtained from x by the rounding.

▶ **Lemma 8.**

$$\sum_{e \in E} c(e)x'(e) \leq \frac{4}{3} \sum_{e \in E} c(e)x(e).$$

Proof. Let H be a cycle of half-integral edges. We show that

$$\sum_{e \in H} c(e)x'(e) \leq \frac{4}{3} \sum_{e \in H} c(e)x(e) \,.$$

Applying this claim to all cycles in the decomposition of F, we can prove the lemma. We use the notations used in the definition of the rounding.

Let x_i denote the vector obtained by rounding $x(e)$, $e \in E(H)$ according to the i-th assignment of labels. 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 the k assignments, $e \in H$ is labeled "+" by the $(k+1)/2$ assignments. 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 $k \geq 3$. ◀

Next, let us prove the feasibility of x'. For a path P and nodes u, v on P, we denote the subpath of P between u and v by $P[u, v]$.

▶ **Lemma 9.** *x' is a feasible solution to the terminal backup problem.*

Proof. Obviously x' is an integer vector. Hence, to prove the feasibility of x', the graph with edge-capacites x' admits a unit of flow from each terminal t to the other terminals. Since $x(\delta(X)) \geq 1$ for each $X \in \mathcal{C}(t)$, the graph capacitated by x admits such a flow. Hence we show that a flow for x' can be obtained by modifying the flow for x. 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.

Recall the definitions of $R_1^t, \ldots, R_{2r(t)}^t$ in Section 3.2. Since we are considering $r \equiv 1$, we have two paths R_1^t and R_2^t for each terminal t with $\mathcal{L}(t) \neq \emptyset$. We assume these paths satisfy Assumption 1. Fix a terminal t, and suppose that the flow from t to the other terminals with edge-capacities x delivers a half unit of flow along a path P, and another half unit along a path Q. We assume that $\mathcal{S} = \{P, Q\}$ satisfies Assumption 2.

If both P and Q contains no half-integral edge (with respect to x) labeled by "−," the flow satisfies the capacity constraints defined from x'. Thus, let us consider the case where P includes a half-integral edge labeled by "−." Let e be the one nearest to t among such edges, and let v be the end node of e near to t.

We first show that there exists $X^* \in \mathcal{L}(t)$ such that $e \in \delta(X^*)$ and $v \in X^*$. For arriving at a contradiction, suppose that such X^* does not exist. e is incident to at least one node
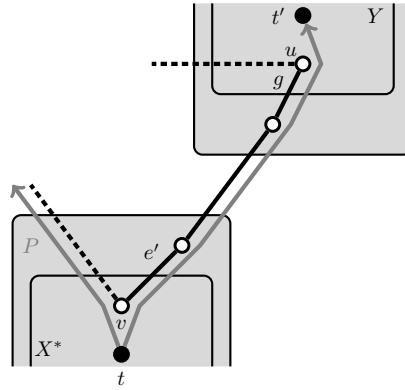
■ **Figure 2** The definitions in the proof of Lemma 9.

set in $\mathcal{L}$. In particular, Lemma 4(ii) implies that there exists a terminal $t' \in T$ and node set $X' \in \mathcal{L}(t')$ such that $e \in \delta(X')$ and $v \in X'$. However, this means that $t' \neq t$ and $P[t, v]$ enters X' when traversed from t to v. Assumption 2 indicates that the subpath of P between v and the end opposite to t is included by $R_1^{t'}$ or $R_2^{t'}$. Hence, the end of P opposite to t is t', and P does not include e, which is a contradiction. Therefore, there exists $X \in \mathcal{L}(t)$ such that $e \in \delta(X)$ and $v \in X$.

This fact indicates that Q contains no "$-$"-labeled half-integral edge because of the following reason. Let P' be the maximal subpath of P that consists of edges incident to node sets in $\mathcal{L}(t)$. Since $\mathcal{L}(t) \neq \emptyset$, there exists R_1^t and R_2^t. By Assumption 2, P' is equal to R_1^t or R_2^t. Without loss of generality, let P' be equal to R_1^t. Then, Assumption 1 indicates that all "$-$"-labeled half-integral edges incident to node sets in $\mathcal{L}(t)$ is included in R_1^t. Since P and Q share no half-integral edges, Q does not included these edges in R_1^t. Hence, if Q contains a "$-$"-labeled half-integral edge, its both end node is included by some node sets in $\mathcal{L} \setminus \mathcal{L}(t)$. However, we can derive a contradiction similarly for the above claim with P.

Since $x(\delta(X^*)) = 1$, the other edge e' incident to v on H is also incident to X^*. By the label-assignment rules, e' is labeled by "$+$." Let H' denote the subpath of H consisting of "$+$"-labeled edges and terminating at v. Let u be the other end node of H', and let g be the edge incident to u on H'. By Lemma 4, there exists $Y \in \mathcal{L}$ with $g \in \delta(Y)$ and $u \in Y$. Y belongs to $\mathcal{L}(t')$ for some $t' \neq t$. g is included in a path $R_1^{t'}$ or $R_2^{t'}$. Without loss of generality, we suppose that $R_1^{t'}$ includes g'. We replace P by the concatenate of $P[t, v]$, H', and $R_1^{t'}[u, t']$. See Figure 2 for illustration of this modification.

Let us observe that this modification preserves the capacity constraints. $P[t, v]$ was a part of P before the modification. The capacity of each edge on H' is increased by $1/2$ when x' replaces x. The capacity of each edge in $R_1^{t'}[u, t']$ is integer. Hence no capacity constraint is violated.   ◀

5   Conclusion

We have presented 4/3-approximation algorithms for the generalized terminal backup problem. Our result also implies that the integrality gaps of the LP relaxations 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 the LP relaxations because of their integrality gaps. For the capacitated b-edge cover problem, an LP relaxation of integrality gap one is known [13]. For obtaining an LP-based polynomial-time algorithm for the generalized terminal backup problem, we have to extend this LP relaxation for the capacitated b-edge cover problem.

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 [9] 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. Recently, Hirai [8] showed that $\mathsf{LP}(f, u)$ can be solved by a combinatorial algorithm. Indeed, he also showed that his algorithm can be used to implement our 4/3-approximation algorithm for the edge-connectivity terminal backup without generic LP solvers.

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 [4] solved a similar problem related to integral multiflows in the edge-capacitated setting. Their problem differs from ours in the fact that $\sum_{A \in \mathcal{A}_t} \psi(A)$ is required to match the specified value for each terminal t.

References

1. Elliot Anshelevich and Adriana Karagiozova. Terminal backup, 3D matching, and covering cubic graphs. *SIAM Journal on Computing*, 40(3):678–708, 2011.
2. Maxim A. Babenko and Alexander V. Karzanov. Min-cost multiflows in node-capacitated undirected networks. *Journal of Combinatorial Optimization*, 24(3):202–228, 2012.
3. Attila Bernáth and Yusuke Kobayashi. The generalized terminal backup problem. In *SODA*, pages 1678–1686, 2014.
4. Michel Burlet and Alexander V. Karzanov. Minimum weight (T, d)-joins and multi-joins. *Discrete Mathematics*, 181(1-3):65–76, 1998.
5. Boris V. Cherkassky. A solution of a problem on multicommodity flows in a network. *Ekonomika i Matematicheskie Metody*, 13(1):143–151, 1977.
6. Lisa Fleischer, Kamal Jain, and David P. Williamson. Iterative rounding 2-approximation algorithms for minimum-cost vertex connectivity problems. *Journal of Computer and System Sciences*, 72(5):838–867, 2006.
7. Hiroshi Hirai. Half-integrality of node-capacitated multiflows and tree-shaped facility locations on trees. *Mathematical Programming*, 137(1-2):503–530, 2013.

8   Hiroshi Hirai. L-extendable functions and a proximity scaling algorithm for minimum cost
         multiflow problem. *ArXiv e-prints*, November 2014.

9   Kamal Jain. A factor 2 approximation algorithm for the generalized Steiner network prob-
         lem. *Combinatorica*, 21(1):39–60, 2001.

10   Alexander V. Karzanov. A problem on maximum multifow of minimum cost. *Combinatorial
         Methods for Flow Problems*, pages 138–156, 1979. in Russian.

11   Alexander V. Karzanov. Minimum cost multifows in undirected networks. *Mathematical
         Programming*, 66(3):313–325, 1994.

12   László Lovász. On some connectivity properties of Eulerian graphs. *Acta Mathematica
         Hungarica*, 28(1):129–138, 1976.

13   Alexander Schrijver. *Combinatorial Optimization – Polyhedra and Efficiency.* Springer,
         2003.