Egerváry Research Group
on Combinatorial Optimization

Technical reportS

TR-2013-07. Published by the Egerváry Research Group, Pázmány P. sétány 1/C, H–1117, Budapest, Hungary. Web site: www.cs.elte.hu/egres . ISSN 1587–4451.

The Generalized Terminal Backup Problem

Attila Bernáth and Yusuke Kobayashi

December 2013

The Generalized Terminal Backup Problem

Attila Bernáth* and Yusuke Kobayashi**

Abstract

We consider the following network design problem, that we call the **Generalized Terminal Backup Problem**. Given a graph (or a hypergraph) $G_0 = (V, E_0)$, a set of (at least 2) terminals $T \subseteq V$ and a requirement $r(t)$ for every $t \in T$, find a multigraph $G = (V, E)$ such that $\lambda_{G_0+G}(t, T - t) \geq r(t)$ for any $t \in T$. In the **minimum cost** version the objective is to find G minimizing the total cost $c(E) = \sum_{uv \in E} c(uv)$, given also costs $c(uv) \geq 0$ for every pair $u, v \in V$. In the **degree-specified version** the question is to decide whether such a G exists, satisfying that the number of edges is a prescribed value $g(v)$ at each node $v \in V$. The **Terminal Backup Problem** solved in [1] is the special case where G_0 is the empty graph and $r(t) = 1$ for every terminal $t \in T$. We solve the Generalized Terminal Backup Problem in the following two cases.

1. In the first case we start with the minimum cost version for $c \equiv 1$, which helps solving the degree-specified version by a splitting-off theorem. This splitting-off theorem in turn provides the solution for the minimum cost version in the case when c is **node-induced**, that is $c(uv) = w(u) + w(v)$ for some node weights $w : V \to \mathbb{R}_+$. The algorithm for this case is polynomial.
2. In the second solved case we turn to the general minimum cost version, and we are able to solve it when G_0 is the empty graph. This includes the **Terminal Backup Problem** [1] ($r \equiv 1$) and the **Maximum-Weight b-matching Problem** ($T = V$). The solution depends on an interesting new variant of a theorem of Lovász and Cherkassky, and on the solution of the so-called **Simplex Matching** problem [1]. Our algorithm is polynomial in $|V|$ and $\max\{r(t) : t \in T\}$.

1 Introduction

Edge-connectivity augmentation problems usually mean the following: find a graph satisfying certain edge-connectivity requirement, and any number of parallel

*Institute of Informatics, University of Warsaw, Poland. Research supported by the ERC StG project PAAl no. 259515. E-mail: athos@cs.elte.hu

**University of Tokyo, Tokyo 113-8656, Japan. Supported by JST, ERATO, Kawarabayashi Large Graph Project and by Grant-in-Aid for Scientific Research. E-mail: kobayashi@mist.i.u-tokyo.ac.jp

edges is allowed between any pair of the nodes. The objective function is usually to minimize the number of edges in the graph found, while the edge-connectivity requirements can vary from problem to problem. The classical result of edge-connectivity augmentation is the theorem of Watanabe and Nakamura [23], who determined the minimum number of edges of a graph $G = (V, E)$ which gives a k-edge-connected graph when added to the input graph $G_0 = (V, E_0)$. This was generalized by Bang-Jensen and Jackson [3] who solved the same problem in the case when G_0 can even be a hypergraph. Another generalization is the **local edge-connectivity augmentation problem** solved by Frank [7], which is the following. Given a graph $G_0 = (V, E_0)$ and requirement $r(u, v) \in \mathbb{Z}_+$ for every pair of nodes $u, v \in V$, find the minimum number of edges of a multigraph G satisfying $\lambda_{G_0+G}(u, v) \geq r(u, v)$ for every pair $u, v \in V$. Here, the edge-connectivity between u and v is denoted by $\lambda(u, v)$ (see Section 2.1 for definition). Note that the same problem becomes NP-complete, if G_0 can be a hypergraph [14]. Ishii and Hagiwara [11] solved the so-called **node-to-area edge-connectivity augmentation problem** which is the following. Given a graph $G_0 = (V, E_0)$, a collection of subsets $\mathcal{W}$ of V (called **areas**) and a function $r : \mathcal{W} \to \mathbb{Z}_+$, find a graph $G = (V, E)$ with smallest possible number of edges such that $\lambda_{G_0+G}(x, W) \geq r(W)$ for any $W \in \mathcal{W}$ and $x \in V$. It is shown in [18] that this problem is NP-complete, however, the authors of [11] have given a polynomial algorithm solving it if $r(W) \geq 2$ for every $W \in \mathcal{W}$ (see also [10]). More generalizations, abstract versions and related results were given by [4, 5, 13, 21], good surveys can be found in [9, 22].

Weighted versions of edge-connectivity augmentation problems are often called **survivable network design problems**. Here we want to find a minimum-cost subgraph of a given supply graph so that the edge-connectivity requirements are satisfied. Parallel copies of the edges might or might not be allowed. These problems are usually NP-hard already in very simple cases, as an example consider the minimum-cost 2-edge-connected subgraph problem. In the Steiner Tree Problem we want to find a minimum cost set of edges that connects every pair of a set of terminals (clearly, the optimum solution can be chosen to be a tree). In its generalization, the Generalized Steiner Network Problem we have a requirement $r(u, v)$ for every pair of nodes $u, v \in V$ and the question is to find a minimum cost graph G so that $\lambda_G(u, v) \geq r(u, v)$ for every pair $u, v \in V$. Jain [12] has given a framework of 2-approximation algorithms that includes many different survivable network design problems (for example, the Generalized Steiner Network Problem). A polynomially solvable survivable network design problem is the **Terminal Backup Problem**, defined as follows. Given a set of terminals $T \subseteq V$ and costs $c(uv) \geq 0$ for every pair $u, v \in V$, find a minimum cost set of edges in which every terminal is connected to *some other terminal*. Clearly, the optimum solution of this problem can always be chosen to be a forest. The Terminal Backup Problem was introduced and solved in [1]. Note the similarity of this problem with the Steiner Tree Problem: here we want that every terminal is connected to *some other* terminal, while the Steiner Tree Problem requires that every terminal is connected to *all other* terminals.

In this paper we consider the following uncapacitated network design problem, which generalizes the Terminal Backup Problem.

Problem 1. *(Generalized Terminal Backup Problem,* ***Problem GTBP****) Given a graph (or a hypergraph) $G_0 = (V, E_0)$, a set of (at least 2) terminals $T \subseteq V$, and a requirement $r(t)$ for every $t \in T$, find a multigraph $G = (V, E)$ such that $\lambda_{G_0+G}(t, T - t) \geq r(t)$ for any $t \in T$.*

Note that G_0 can be a hypergraph, but G has to be a graph here, in which we can include any number of parallel edges between any pair of nodes.

In the **minimum cost version** of Problem 1 (Problem **MC-GTBP**) we want to minimize the total cost $c(E) = \sum_{uv \in E} c(uv)$ of the solution found, given also costs $c(uv) \geq 0$ for every pair $u, v \in V$. In the **degree-specified version** of Problem 1 (Problem **DS-GTBP**) we want to decide whether such a graph G exists, satisfying that the number of edges is a prescribed value $g(v)$ at each node $v \in V$.

In this paper we solve the following special cases of Problem GTBP.

1. An **edge-connectivity augmentation type problem**: here we start with the minimum cost version for $c \equiv 1$, which helps solving the degree-specified version by a splitting-off theorem. This splitting-off theorem in turn provides the solution for the minimum cost version in the case when c is **node-induced**. Here, the cost function c is said to be **node-induced** if there exists a weight function $w : V \to \mathbb{R}_+$ such that $c(uv) = w(u) + w(v)$ for every pair $u, v \in V$.

2. A **survivable network design problem**: we turn to the general minimum cost version, and we are able to solve it when G_0 is the empty graph. The solution depends on Lemma 4, a variant of Theorem 1, which is of independent interest. The second ingredient of the solution is the algorithm given by Anshelevich and Karagiozova [1] for the problem called **simplex matching problem**.

Problem GTBP is a new network design problem. It includes the Terminal Backup Problem [1] (by letting G_0 to be an empty graph and $r \equiv 1$) and the Maximum-Weight b-matching Problem ($T = V$), but it seems that this particular problem was not considered before, we have not found this type of question in the literature. A special case of this problem (the degree-specified version) was raised by András Frank (private communication). The following, somewhat related theorem of Lovász and Cherkassky can be considered as a motivation for our problem.

Theorem 1 (Lovász [15] and Cherkassky [6])**.** *Let $G = (V, E)$ be an undirected graph and $T \subseteq V$ a set of terminals so that the degree of v is even for every $v \in V - T$. Then there is a set F of edge-disjoint paths such that each path has its endnodes in T and for each element $t \in T$, the paths in F ending at t form a maximum set of edge-disjoint $(t, T - t)$-paths.*

We give an interesting variant of this theorem (see Lemma 4). Theorem 1 was generalized in many directions, for example Mader [16] determined the maximum number of edge-disjoint T-paths in a graph G in which the degree of v is not necessarily even for every $v \in V - T$ (where a path is called a **T-path** if both its endnodes are in T, see also [20, Corollary 73.2b]). We could not see our Lemma 4 as an easy corollary of these results.

The paper is organized as follows. In Section 2 we give the necessary definitions and results. In Section 3 we solve the edge-connectivity augmentation problem by first solving the minimum cardinality case in subsection 3.1, and the proving the splitting-off theorem and exploring its consequences in subsection 3.2. In Section 4 we solve the survivable network design problem: in subsection 4.1 we give the algorithm, and in subsection 4.2 we prove the main ingredient of our solution, Lemma 4. We close the paper with some concluding remarks in Section 5.

2 Preliminaries

2.1 Hypergraphs and edge-connectivity

For general graph theoretic notations we will follow [8]. For subsets X, Y of a ground set V let $X - Y = \{v \in X : v \notin Y\}$; sometimes we will also use $X + Y$ to mean $X \cup Y$. A hypergraph is a pair $H = (V, \mathcal{E})$ where V is some finite set of nodes and $\mathcal{E}$ is a multiset of subsets of V. The members of $\mathcal{E}$ are called **hyperedges**, a hyperedge of size at most 2 is called a **graph edge** (or simply **edge**), and a hyperedge of size 1 is called a **loop**. A graph is a special hypergraph containing only edges. If H and G are hypergraphs on the same node set V then $H + G$ is the hypergraph on node set V in which the multiplicity of a hyperedge is the sum of its multiplicities in H and in G. For a hypergraph $H = (V, \mathcal{E})$ and a set $X \subseteq V$ we say that a hyperedge $e \in \mathcal{E}$ enters X if neither $e \cap X$ nor $e \cap (V - X)$ is empty, and we define $d_H(X) = |\{e \in \mathcal{E} : e \text{ enters } X\}|$. If a set contains only one element v then we will write v instead of $\{v\}$; thus $d_H(v)$ means $d_H(\{v\})$, etc.

A **path** between nodes s and t of a hypergraph H is an alternating sequence of distinct nodes and hyperedges $(s = v_0, e_1, v_1, e_2, \dots, e_k, v_k = t)$, such that $v_{i-1}, v_i \in e_i$ for all i between 1 and k. For a path $P = (v_0, e_1, v_1, e_2, \dots, e_k, v_k)$, its subsequence $(v_i, e_{i+1}, v_{i+1}, e_{i+2}, \dots, e_j, v_j)$ between v_i and v_j $(0 \le i < j \le k)$ is called a **subpath** of P and denoted by $P[v_i, v_j]$. For sets $S, T \subseteq V$ of nodes in a hypergraph $H = (V, \mathcal{E})$, the **edge-connectivity** $\lambda_H(S, T)$ between S and T in H is defined as the maximum number of pairwise hyperedge-disjoint paths, where each path has one endnode in S, and the other in T (where we understand $\lambda_H(S, T) = \infty$ if $S \cap T \neq \emptyset$). The following theorem of Menger shows that this value coincides with the size of a minimum S-T cut.

Theorem 2 (Menger's Theorem [17])**.** *Let $H = (V, \mathcal{E})$ be a hypergraph, and $S, T \subseteq V$. Then*
$$\lambda_H(S, T) = \min\{d_H(X) : T \subseteq X \subseteq V - S\}.$$

2.2 Skew-supermodular functions

We say that a graph G **covers** a set function p if $d_G(X) \ge p(X)$ holds for every $X \subseteq V$. In our proof of the first result, we regard the problem as a covering problem of a skew-supermodular set function. In this subsection, we describe some notations and properties of skew-supermodular functions.

A set function $p : 2^V \to \mathbb{Z} \cup \{-\infty\}$ is called **skew-supermodular** if at least one of the following two inequalities holds for every $X, Y \subseteq V$:

$$p(X) + p(Y) \leq p(X \cap Y) + p(X \cup Y), \qquad (\cap\cup)$$
$$p(X) + p(Y) \leq p(X - Y) + p(Y - X). \qquad (-)$$

A set function is **symmetric** if $p(X) = p(V - X)$ for every $X \subseteq V$. For a hypergraph H, we can easily see that $p = -d_H$ is symmetric and satisfies both $(\cap\cup)$ and $(-)$ for any $X, Y \subseteq V$. Let the **symmetrized** p^s of a set function p be defined with the formula $p^s(X) = \max(p(X), p(V - X))$ for every $X \subseteq V$. We can see that a graph G covers p if and only if it covers p^s. We can also see the following claim.

Claim 2.1 ([5]). *The symmetrized of a skew-supermodular function is (symmetric and) skew supermodular.*

For a function $g : V \to \mathbb{R}$ or a vector $g \in \mathbb{R}^V$, we denote $g(X) = \sum_{v \in X} g(v)$ for $X \subseteq V$. For a set function $p : 2^V \to \mathbb{Z} \cup \{-\infty\}$ we introduce the polyhedron

$$C(p) = \{x \in \mathbb{R}^V : x(Z) \geq p(Z) \ \forall Z \subseteq V, x \geq 0\}.$$

This polyhedron will be used to characterize the feasibility of the degree-specified version of Problem GTBP (see Theorem 5). An important property of $C(p)$ is the following.

Theorem 3 ([2]). *If $p : 2^V \to \mathbb{Z} \cup \{-\infty\}$ is a skew supermodular function with $p(\emptyset) \leq 0$ then $C(p)$ is an integer polyhedron (namely an **integer contrapolymatroid**).*

A **subpartition of** V is a family of disjoint subsets of V. We say that an $x \in C(p)$ is **minimal** if we cannot decrease $x(v)$ at any v without violating some condition in the definition of $C(p)$. The properties of contrapolymatroids relevant for us are formulated in the following corollary of Theorem 3. See details about contrapolymatroids in [20].

Corollary 1. *If p is as in Theorem 3 then we have the following.*

- $\max\{\sum_{X \in \mathcal{X}} p(X) : \mathcal{X}$ *is a subpartition of* $V\} = \min\{1 \cdot x : x \in C(p)\}$.
- *Any minimal $m \in C(p)$ achieves $m(V) = \min\{1 \cdot x : x \in C(p)\}$.*
- *Given any $w : V \to \mathbb{R}_+$, an (integer) optimal solution of $\min\{w \cdot x : x \in C(p)\}$ can be found in polynomial time (with a simple greedy algorithm), assuming that we can test membership in $C(p)$.*

2.3 The splitting-off operation

Let $p : 2^V \to \mathbb{Z} \cup \{-\infty\}$ be a symmetric, skew-supermodular function that satisfies $p(\emptyset) \leq 0$ and let $m : V \to \mathbb{Z}$ be a nonnegative function satisfying $m(X) \geq p(X)$ for any $X \subseteq V$ (i.e. an integer element of $C(p)$). We would like to decide whether there is a graph (or possibly hypergraph) G covering p that satisfies $d_G(v) = m(v)$

for every $v \in V$. Let $u, v \in V$ be two nodes with $m(u), m(v) > 0$. The operation **splitting-off (at u and v)** is the following: we substitute m and p with m' and p' where $m'(x) = m(x)$ if $x \in V - \{u, v\}$ and $m'(x) = m(x) - 1$ if $x \in \{u, v\}$ and $p' = p - d_{(V,\{(uv)\})}$ (where $(V, \{(uv)\})$ is a graph having only one edge: note that p' is symmetric and skew-supermodular). If $m'(X) \geq p'(X)$ holds for any $X \subseteq V$, then we say that the splitting off is **admissible**. A set X is **dangerous** if $m(X) - p(X) \leq 1$. The following claim is well known.

Claim 2.2 (see e.g. [5]). *The splitting off at u and v is admissible if and only if there is no dangerous set containing both u and v.*

We will use the following lemma.

Lemma 1 ([5, 19]). *Let $p : 2^V \to \mathbb{Z} \cup \{-\infty\}$ be a symmetric skew-supermodular function and $m \in C(p) \cap \mathbb{Z}^V$. If $\max\{p(X) : X \subseteq V\} > 1$, then there is an admissible splitting-off.*

3 Solution of the edge-connectivity augmentation problem

In this section we solve the following variants of Problem GTBP. We start with the **minimum cardinality version**, in which the number of edges $|E|$ of G is to be minimized (that is, the minimum cost version with cost function $c \equiv 1$). Then we prove a splitting-off theorem that solves the **degree-specified version**. Unlike other edge-connectivity augmentation problems, here the minimum cardinality version of the problem is easier than the degree-specified version, and it helps proving the splitting-off theorem. The splitting-off theorem gives rise to the solution of the minimum cost version for **node-induced cost function** (that is, we find a graph G minimizing $\sum_{v \in V} w(v) d_G(v)$), given some node-weights $w(v) \geq 0$ for every $v \in V$).

Consider Problem GTBP above. To simplify the discussion, let $T = \{t_1, t_2, \ldots, t_k\}$ and let $r_i = r(t_i)$ for every i. For any terminal t_i let $d_i = \min\{d_{G_0}(X) : X \cap T = \{t_i\}\}$. By Menger's theorem $d_i = \lambda_{G_0}(t_i, T - t_i)$. Let furthermore X_i be an inclusionwise minimal subset with $X_i \cap T = \{t_i\}$ and $d_{G_0}(X_i) = d_i$.

Lemma 2. *For different indices $i \neq j$ we have $Y_i \cap X_j = \emptyset$, where Y_i is a set with $Y_i \cap T = \{t_i\}$ and $d_{G_0}(Y_i) = d_i$. Consequently, $X_1, X_2, \ldots, X_k$ is a subpartition of V.*

Proof. Assume $Y_i \cap X_j \neq \emptyset$. Since

$$\begin{aligned} d_i + d_j &= d_{G_0}(Y_i) + d_{G_0}(X_j) \\ &\geq d_{G_0}(Y_i - X_j) + d_{G_0}(X_j - Y_i) \geq d_i + d_j, \end{aligned}$$

we have $d_{G_0}(X_j - Y_i) = d_j$, which contradicts the minimality of X_j. □

Let us define a set function $R : 2^V \to \mathbb{Z} \cup \{-\infty\}$ by

$$R(X) = \begin{cases} r_i & \text{if } X \cap T = \{t_i\}, \\ -\infty & \text{otherwise.} \end{cases}$$

It is clear that a graph G is feasible for Problem GTBP if and only if $d_G(X) \geq R(X) - d_{G_0}(X)$ holds for every subset $X \subseteq V$ (i.e., G covers $R - d_{G_0}$).

Claim 3.1. *The function R is skew-supermodular (and then so is the function $R - d_{G_0}$).*

Proof. Let $X, Y \subseteq V$. We can assume that $R(X)$ and $R(Y)$ are both finite, otherwise there is nothing to prove. If $X \cap T = Y \cap T$ then $(\cap\cup)$ holds for R (with equality), otherwise $(-)$ holds for R (again, with equality). The skew-supermodularity of R implies the skew-supermodularity of $R - d_{G_0}$. □

Let $R^s(X) = \max\{R(X), R(V - X)\}$ for any $X \subseteq V$ (the **symmetrized of** R): it is a symmetric and skew supermodular function by Claim 2.1. Let finally $p(X) = R^s(X) - d_{G_0}(X)$ for any $X \subseteq V$, which is symmetric and skew-supermodular. Note that G covers $R - d_{G_0}$ if and only if G covers p.

Membership oracle for $C(p)$. In order to turn our proofs into polynomial algorithms, we describe a membership oracle for $C(p)$, where $p = R^s - d_{G_0}$. This oracle is needed in Corollary 1, and in our Splitting-off Theorem 5; note that this implies a membership oracle for $C(p - d_G)$ for any graph G, since we can add G to G_0. Given some $x : V \to \mathbb{Z}_+$, we want to decide whether $x \in C(p)$ or not. This is done as follows. Add a new node s to G_0 and an edge with multiplicity $x(v)$ between s and every $v \in V$. Denote the resulting hypergraph by H. We claim that $x \in C(p)$ if and only if $\lambda_H(t, T - t) \geq r(t)$ holds for every $t \in T$, which can be checked with maximum flow computations. We prove this claim. If $x \notin C(p)$ then $x(Z) < R^s(Z) - d_{G_0}(Z)$ for some $Z \subseteq V$. By the definition of the function R, there exists some $t \in T$ so that $Z \cap T = \{t\}$ or $Z \cap T = T - \{t\}$: for this t we have $\lambda_H(t, T - t) < r(t)$. On the other hand, if $\lambda_H(t, T - t) < r(t)$ for some $t \in T$ then $d_H(Z) < r(t)$ for some set $Z \subseteq V + s$ separating t and $T - t$. We can assume that $s \notin Z$ and then for this set we have $x(Z) < p(Z)$.

3.1 Solution of the minimum cardinality version

Let us introduce $r'_i = \max\{r_i - d_i, 0\}$ for every $i = 1, 2, \dots, k$. Assume without loss of generality that $r'_1 \geq r'_2 \geq \cdots \geq r'_k$. Note that $r'_i = \max\{R(X_i) - d_{G_0}(X_i), 0\}$ for every i and $r'_1 = \max\{R(X) - d_{G_0}(X) : X \subseteq V\} = \max\{p(X) : X \subseteq V\}$ (by assuming that $r'_1 > 0$).

Theorem 4. *The minimum number of edges of a graph G that satisfies the requirements of Problem GTBP is equal to $\gamma = \max\{r'_1, \lceil \frac{\sum_i r'_i}{2} \rceil\}$.*

Proof. It is clear from Lemma 2 that $\max\{r'_1, \lceil \frac{\sum_i r'_i}{2} \rceil\}$ is a lower bound. On the other hand, let us find an arbitrary loopless graph G on nodeset T such that $d_G(t_i) \geq r'_i$ for every i and $|E(G)| = \gamma$. Such a graph exists and satisfies our requirements, since $\lambda_G(t_i, T - t_i) \geq r'_i$ for every i. □

Corollary 2. $\max\{\lceil \frac{1}{2} \sum_{X \in \mathcal{X}} p(X) \rceil$: *$\mathcal{X}$ is a subpartition of V*$\} = \gamma$.

Proof. It is clear that $\lceil \frac{1}{2}\sum_{X\in\mathcal{X}} p(X)\rceil$ is a lower bound of γ. The other direction follows from Lemma 2 and Theorem 4: if $\gamma = r'_1$ then take $\mathcal{X} = \{X_1, V - X_1\}$, otherwise take $\mathcal{X} = \{X_i : r'_i > 0\}$. □

3.2 The splitting-off theorem and its consequences

Next we solve the degree-specified version of Problem GTBP. If a specified degree of some vertex is too large compared to other degrees (i.e., $g(v) > g(V - v)$ for some $v \in V$), then we need to care about loops. For a node $v \in V$ in a graph $G = (V, E)$ let $d^+_G(v)$ be $d_G(v)$ plus 2 times the number of loops at v, which is a standard definition of the degree of v in a graph with loops. Recall that $p : 2^V \to \mathbb{Z} \cup \{-\infty\}$ is defined by $p(X) = R^s(X) - d_{G_0}(X)$ for $X \subseteq V$.

Theorem 5. *Given values $g(v) \in \mathbb{Z}_+$ for every node $v \in V$, there exists a graph G with $d^+_G(v) = g(v)$ at every $v \in V$ satisfying the requirements of Problem GTBP if and only if $g(V)$ is even and $g(Z) \geq p(Z)$ holds for every $Z \subseteq V$.*

Proof. To prove necessity of the conditions, assume that such a graph G exists. Summing $d^+_G(v) = g(v)$ for every $v \in Z$ gives that $g(Z) \geq d_G(Z)$, therefore the condition $g(Z) \geq p(Z)$ is necessary for any Z. Similarly, summing $d^+_G(v) = g(v)$ for every $v \in V$ gives $g(V) = 2|E(G)|$, therefore $g(V)$ has to be even.

To prove sufficiency, let us assume that $m : V \to \mathbb{Z}_+$ is such that $m(v) \leq g(v)$ for every $v \in V$, $m(Z) \geq p(Z)$ holds for any $Z \subseteq V$, but we cannot decrease any $m(v)$ without violating this condition (such an m can be found greedily, starting from $m = g$). By Corollary 2 and Corollary 1, we know that $m(V)$ is either 2γ or $2\gamma - 1$: in the latter case let us increase $m(v)$ by one for an arbitrary v with $m(v) \leq g(v) - 1$. If we show that there exists a graph G satisfying $d_G(v) = m(v)$ for every v, that satisfies the requirements of Problem GTBP, then the theorem is proved (because we can add more edges, possibly loops, to achieve that $d^+_G(v) = g(v)$ at every $v \in V$: note however that we can avoid loops unless $g(v) > g(V - v)$ for some node $v \in V$). In order to prove this we only need to show that an **admissible splitting-off** exists: that is, we can find nodes x, y such that $m(x) > 0, m(y) > 0$ and any set X containing x, y has $m(X) \geq p(X) + 2$ (and then the proof is ready by induction).

If $r'_1 = \max\{p(Z) : Z \subseteq V\} > 1$ then there exists an admissible splitting-off by Lemma 1. So we can assume that $r'_1 = 1$. We can also assume that $m(V) \geq 4$, implying $r'_2 = r'_3 = 1$, otherwise there trivially exists an admissible splitting-off. Choose an arbitrary $x \in X_1$ and $y \in X_2$ with $m(x) > 0, m(y) > 0$ (such nodes exist, since $m(X_i) \geq p(X_i)$ for $i = 1, 2$), and assume that the splitting-off at x and y is not admissible. This means that there exists a set X containing x, y with $m(X) \leq p(X) + 1$. Since $m(X) \geq 2$, this means that $p(X) = 1$ and $m(X) = 2$, implying that $X_3 - X \neq \emptyset$. Since the role of t_1 and t_2 is symmetric here, we can assume that either $X \cap T = \{t_1\}$ or $T - X = \{t_1\}$. In both cases $d_{G_0}(X) = d_1$ must hold. In the first case $X \cap X_2 \neq \emptyset$, contradicting Lemma 2. In the second case $d_{G_0}(V - X) = d_1$ and $(V - X) \cap X_3 \neq \emptyset$ contradicts Lemma 2. □

Using Corollary 1 and our splitting-off Theorem 5 above we obtain the solution of the node-weighted version of Problem GTBP.

Theorem 6. *Given Problem GTBP and node weights $w(v)$ for every node $v \in V$, we can find a solution G minimizing $\sum_{v\in V} w(v)d_G(v)$ in polynomial time.*

Proof. By Corollary 1, we can find a vector $g : V \to \mathbb{Z}_+$ minimizing $\sum_{v\in V} w(v)g(v)$. If $g(V)$ is odd then increase $g(v)$ by one for the node v that has smallest weight. The theorem is proved by our splitting-off Theorem 5. □

We mention the following related result. In our problem setting (Problem 1) we insist that G has to be a graph. If we allow hyperedges in G then we arrive at a different problem, but it is not clear how to choose the objective function. A natural candidate is to minimize the **total size** of G (where the total size of a hypergraph is the sum of the sizes of its hyperedges: note that this is twice the number of edges, if the hypergraph is in fact a graph). A more general version would consider a **node-induced cost function**, as in Theorem 6: given node weights $w(v)$ for every node $v \in V$, and the cost of choosing a hyperedge is the sum of the weights of the nodes contained in that hyperedge. This general problem is solved by Szigeti in [21], as it is contained in the framework of covering a skew-supermodular function by hyperedges.

4 Solution of the survivable network design problem

In this section we solve minimum cost version of Problem GTBP in the special case when G_0 is the empty graph. Let us formulate this problem separately.

Problem 2. *What is the minimum cost of a multigraph $G = (V, E)$ such that $\lambda_G(t, T-t) \geq r(t)$ for any $t \in T$, given a terminal set $T \subseteq V$ ($|T| \geq 2$), a requirement $r(t) \in \mathbb{Z}_+$ for every $t \in T$, and a cost $c(uv) \geq 0$ for every pair $u, v \in V$.*

We observe that Problem 2 is polynomially solvable if $T = V$, because now the question is to find a smallest cost graph $G = (V, E)$ so that the degree $d_G(v)$ of each node v is at least $r(v)$. This is a minimum-cost b-edge cover problem [20, Section 21.7] (which is equivalent to the maximum-weight b-matching problem with a simple reduction).

We also note that the special case $r \equiv 1$ of Problem 2 is known as the **Terminal Backup Problem**, and is shown to be polynomially solvable in [1]. It seems that the methods of [1] also apply to the case when G_0 is not an empty graph (and $r(t) = 1$ for every $t \in T$), but the details need to be clarified.

The algorithm for the Terminal Backup Problem in [1] is based on a polynomial-time algorithm for the **simplex matching problem**. In an instance of the simplex matching problem, we are given a hypergraph $H = (V, \mathcal{E})$ that has hyperedges of sizes 2 and 3 with edge costs $\gamma(e)$, and the objective is to find a perfect matching of H with minimum total cost. Since this problem is NP-hard in general, we consider instances with the **simplex condition**, which states that for any hyperedge $\{u_1, u_2, u_3\} \in \mathcal{E}$ of size 3, $\{u_1, u_2\}, \{u_2, u_3\}, \{u_3, u_1\} \in \mathcal{E}$ and

$$\gamma(\{u_1, u_2\}) + \gamma(\{u_2, u_3\}) + \gamma(\{u_3, u_1\}) \leq 2\gamma(\{u_1, u_2, u_3\}).$$

The main theorem in [1] is as follows.

Theorem 7 (Anshelevich and Karagiozova [1])**.** *There is a polynomial-time algorithm for the simplex matching problem with the simplex condition.*

4.1 Algorithm

We will give an algorithm for the general case of Problem 2. In order to solve this problem, we investigate the structure of the optimal solution. For a given instance of Problem 2, define a family $\mathcal{E} = \binom{T}{2} \cup \binom{T}{3} \subseteq 2^T$, where $\binom{T}{2} = \{\{t_1, t_2\} \mid t_1, t_2 \in T,\ t_1 \neq t_2\}$, and $\binom{T}{3} = \{\{t_1, t_2, t_3\} \mid t_1, t_2, t_3 \in T,\ t_1 \neq t_2 \neq t_3 \neq t_1\}$, and let $\gamma : \mathcal{E} \to \mathbb{R}_+$ be the cost function such that $\gamma(\{t_1, t_2\})$ is the minimum cost of a t_1-t_2 path (with respect to the cost function c) and $\gamma(\{t_1, t_2, t_3\})$ is the minimum cost of a Steiner tree spanning t_1, t_2 and t_3 (with respect to the cost function c).

Consider the following problem.

Problem 3. *Suppose $\mathcal{E}$ and γ are defined as above. Find a minimum cost multi-hypergraph $H = (T, \mathcal{F})$ such that $\mathcal{F}$ is a multiset of $\mathcal{E}$ and $d_H(t) \geq r(t)$ for any $t \in T$.*

We can show the following lemma, whose proof is given in Section 4.2.

Lemma 3. *The optimal value of Problem 3 is equal to the the optimal value of Problem 2.*

Furthermore, the optimal solutions correspond to each other, i.e., an optimal solution of Problem 2 can be decomposed into paths and Steiner trees with three leafs.

Note that the corresponding result is given in [24] for the special case $r \equiv 1$. Based on this lemma showing the correspondence between Problem 2 and Problem 3, we propose the following algorithm for Problem 2.

Algorithm for Problem 2

Step 1 Construct the family $\mathcal{E} = \binom{T}{2} \cup \binom{T}{3}$ and compute the cost $\gamma(e)$ for each $e \in \mathcal{E}$ defined as above.

Step 2 Let $R := \max_{t \in T}\{r(t)\}$. Construct a simplex matching instance consisting of hypergraph $(T^+, \mathcal{E}_+ \cup \mathcal{E}_0)$ and costs as follows.

Step 2-1. The ground set is $T^+ = \{t^{(1)}, t^{(2)}, \ldots, t^{(R+2)} : t \in T\}$, that is we introduce $R + 2$ copies of each node of T.

Step 2-2 The hyperedges in $\mathcal{E}_+$ and their costs are the following. For each $\{t_1, t_2\} \in \mathcal{E}$, add edges $\{t_1^{(i)}, t_2^{(j)}\}$ with cost $\gamma(\{t_1, t_2\})$ for all $i, j \in \{1, \ldots, R+2\}$. Similarly, for each $\{t_1, t_2, t_3\} \in \mathcal{E}$, add edges $\{t_1^{(i)}, t_2^{(j)}, t_3^{(k)}\}$ with cost $\gamma(\{t_1, t_2, t_3\})$ for all $i, j, k \in \{1, \ldots, R + 2\}$.

Step 2-3 The hyperedges in $\mathcal{E}_0$ and their costs are the following. For each $t \in T$, add edges $\{t^{(i)}, t^{(j)}\}$ with cost 0 for $r(t) + 1 \leq i < j \leq R + 2$, and add edges $\{t^{(i)}, t^{(j)}, t^{(k)}\}$ with cost 0 for $r(t) + 1 \leq i < j < k \leq R + 2$.

Step 3 Solve the obtained simplex matching instance using Theorem 7. Then, from the optimal solution of the simplex matching problem, we can construct a solution of Problem 3 by ignoring the hyperedges in $\mathcal{E}_0$ and contracting $t^{(1)}, \ldots, t^{(R+2)}$ to a single vertex for each $t \in T$.

Step 4 Output a solution $G = (V, E)$ of Problem 2 that consists of paths and Steiner trees corresponding to the solution of Problem 3 .

Before proving the correctness of this algorithm, we give a small claim on the optimal solutions of Problem 3.

Claim 4.1. *Problem 3 always has an optimal solution $H = (T, \mathcal{F}')$ such that $d_H(t) \in \{r(t), r(t)+1, \ldots, R\}$ for any $t \in T$, where $R := \max_{t \in T}\{r(t)\}$.*

Proof. Assume that H is an optimum solution and $d_H(t) > R$ for some $t \in T$. We replace H with another optimum solution H' having $d_{H'}(t) < d_H(t)$, and then the proof is ready by induction.

Assume first that a hyperedge $\{t, t', t''\}$ is in $\mathcal{F}'$: replace it with the edge $\{t', t''\}$ (i.e. decrease the multiplicity of $\{t, t', t''\}$ and increase that of $\{t', t''\}$ to get H'). It is easy to see that H' is feasible, and since $\gamma(\{t, t', t''\}) \geq \gamma(\{t', t''\})$, H' is also optimal.

If there exist 2 edges $\{t, t'\}, \{t, t''\} \in \mathcal{F}'$ for which $t' \neq t''$ then replace them with $\{t, t', t''\}$ to get H'. It is again clear that the new solution is feasible, and $\gamma(\{t, t'\}) + \gamma(\{t, t''\}) \geq \gamma(\{t, t', t''\})$ shows that it is also optimal.

If none of the above can be applied then t is incident with more then R copies of an edge $\{t, t'\}$ for some $t' \in T - t$. In this case simply delete a copy of this edge: since $d_H(t') \geq R + 1$, the obtained hypergraph H' is still feasible, and thus it is also optimal. □

Our main result is stated as follows.

Theorem 8. *Our algorithm solves Problem 2 in polynomial time in $|V|$ and $R = \max_{t \in T}\{r(t)\}$.*

Proof. First, an easy but important observation is that a minimum cost Steiner tree spanning t_1, t_2 and t_3 consists of (at most) three paths each connecting a hub vertex $v \in V$ and each t_i. This shows that the simplex condition holds when we apply Theorem 7 in Step 3. Furthermore, based on this observation, in Step 1, $\gamma(\{t_1, t_2, t_3\})$ can be computed in polynomial time by guessing the hub vertex v and using a shortest path algorithm.

Next, we show the optimality of the output. Without the set of hyperedges $\mathcal{E}_0$ added in Step 2-3 in our algorithm, Step 3 would find a minimum cost multihypergraph $H = (T, \mathcal{F})$ such that $\mathcal{F}$ is a multiset of $\mathcal{E}$ and $d_H(t) = R + 2$ for any $t \in T$. By using edges in $\mathcal{E}_0$, we can cover k vertices in $t^{(r(t)+1)}, t^{(r(t)+2)}, \ldots, t^{(R+2)}$, where k can be $0, 2, 3, 4, \ldots, R + 2 - r(t)$ (note that we cannot cover exactly one vertex with a zero cost hyperedge). Therefore, in Step 3 of our algorithm, we obtain a minimum cost multihypergraph $H = (T, \mathcal{F})$ such that $\mathcal{F}$ is a multiset of $\mathcal{E}$ and $d_H(t) \in \{r(t), r(t)+1, \ldots, R\}$ for any $t \in T$, which is an optimal solution of Problem 3 by the above argument and Claim 4.1. Therefore, by Lemma 3, we obtain an optimal solution of Problem 2 in Step 4.

Finally, we note that since we introduced $R+2$ vertices for each vertex $u \in T$ in Step 2-1, the running time of our algorithm is polynomial in $|V|$ and $R = \max_{t\in T}\{r(t)\}$. $\square$

We remark that using $R+2$ copies of each node $t \in T$ in Step 2-1 of our algorithm was only needed because we wanted to use the Simplex Matching Algorithm of Anshelevich and Karagiozova [1] as a black box. We believe that the Simplex Matching Algorithm of [1] can be modified to solve directly Problem 3, which would imply an algorithm for our Problem 2 whose running time is not pseudo-polynomial but polynomial.

4.2 Proof of Lemma 3

In this section, we give a proof of Lemma 3. To show this lemma, it suffices to prove the following lemma, which can be seen as a variant of Theorem 1.

Lemma 4. *Suppose we have a multigraph $G = (V, E)$ with a set of (at least two) terminals $T \subseteq V$. Then there is a set F of mutually edge-disjoint trees in G, so that each tree has at most 3 leafs, all these leafs are in T and for each element $t \in T$, $\lambda_G(t, T-t)$ trees in F are incident to t.*

Proof. Define $r(t) := \lambda_G(t, T-t)$ for each $t \in T$. We use induction on $\sum_{t\in T} r(t)$.

We take one terminal $t_0 \in T$. By Menger's theorem, we have $r(t_0)$ edge-disjoint paths $P_1, \dots, P_{r(t_0)}$ from t_0 to $T - t_0$. Let $P_1 = (v_0 = t_0, e_1, v_1, e_2, v_2, \dots, v_{l-1}, e_l, v_l \in T - t_0)$. For $i = 1, 2, \dots, l$, we define $G_i := G - \{e_1, \dots e_i\}$. Let $i \in \{1, 2, \dots, l\}$ be the integer satisfying the following condition:

- $\lambda_{G_{i-1}}(t, T-t) = r(t)$ for each $t \in T - t_0$ and
- $\lambda_{G_i}(t, T-t) = r(t) - 1$ for some $t \in T - t_0$,

where $G_0 := G$. Since $\lambda_{G_l}(v_l, T - v_l) \le r(v_l) - 1$, such an integer i must exist.

In the graph G_i, for each $t \in T$, let $X_t \subseteq V$ be the minimum vertex set such that $t \in X_t \subseteq V - (T - t)$ and $d_{G_i}(X_t) = \lambda_{G_i}(t, T-t)$. By the minimality of X_t (and standard uncrossing techniques, see Lemma 2), we can see that X_t's are mutually disjoint. Let $X_0 := V \setminus \bigcup_{t\in T} X_t$. By the choice of i, we may assume that $\lambda_{G_i}(t', T-t') = r(t') - 1$ for some $t' \in T - t_0$. Since e_i connects $X_{t'}$ and $V - X_{t'}$, we consider the following two cases.

Case 1: e_i connects $X_{t'}$ and $X_{t_0} \cup X_0$.

By Menger's theorem, in G_{i-1}, we can take edge-disjoint paths $Q_1, \dots, Q_{r(t')}$ from t' to $T - t'$. Since each path contains exactly one edge connecting $X_{t'}$ and $V - X_{t'}$ in G_{i-1}, without loss of generality, we may assume that Q_1 contains e_i. By concatenating the subpath of Q_1 between t' and e_i and the subpath $(v_0, e_1, v_1, e_2, \dots, v_{i-1})$ of P_1, we obtain a path P from t_0 and t', that is, $P := P_1[t_0, v_{i-1}] + Q_1[t', v_{i-1}]$. (Note that if $v_{i-1} \in X_{t'}$ and $v_i \notin X_{t'}$, then P does not contain e_i.)

Let $E(P)$ be the set of edges in P, and let G' be a new graph obtained from $G - E(P) - \{e_i\}$ by shrinking $X_{t'}$ to a single vertex (and the shrunk vertex is regarded as the terminal t'). Then,

- it is clear that $\lambda_{G'}(t, T - t) = r(t)$ for each $t \in T - t_0 - t'$,
- $\lambda_{G'}(t_0, T - t_0) = r(t_0) - 1$ by the existence of $P_2, \ldots, P_{r(t_0)}$, and
- $\lambda_{G'}(t', T - t') = r(t') - 1$ by the existence of $Q_2, \ldots, Q_{r(t')}$.

Note that $Q_2, \ldots, Q_{r(t')}$ do not share an edge with $P_1[t_0, v_{i-1}]$, since they are paths in G_{i-1}. By induction hypothesis, G' contains a set F' of trees of the required form, and F' can be extended in $G - E(P)$ by using subpaths of $Q_2, \ldots, Q_{r(t')}$. Therefore, these objects together with P form a desired set of trees in G.

Case 2: e_i connects $X_{t'}$ and $X_{t''}$ for some $t'' \in T - t_0 - t'$.
By Menger's theorem, in G_{i-1}, we can take edge-disjoint paths $Q_1, \ldots, Q_{r(t')}$ from t' to $T - t'$. Since each path contains exactly one edge connecting $X_{t'}$ and $V - X_{t'}$ in G_{i-1} without loss of generality, we may assume that Q_1 contains e_i. Similarly, we take edge-disjoint paths $R_1, \ldots, R_{r(t'')}$ from t'' to $T - t''$ in G_{i-1} and we may assume that R_1 contains e_i.

By concatenating the subpath of Q_1 between t' and e_i, the subpath of R_1 between t'' and e_i, and the subpath $(v_0, e_1, v_1, e_2, \ldots, v_{i-1})$ of P_1, we obtain a tree P connecting t_0, t', and t'', that is, $P := P_1[t_0, v_{i-1}] + Q_1[t', v_{i-1}] + R_1[t'', v_{i-1}]$. (Note that even if $v_{i-1} \in X_{t'}$ and $v_i \notin X_{t'}$, the subpath $R_1[t'', v_{i-1}]$ must contain e_i.)

Let $E(P)$ be the set of edges in P, and let G' be a new graph obtained from $G - E(P)$ by shrinking $X_{t'}$ and $X_{t''}$ to single vertices (and the shrunk vertices are regarded as the terminals t' and t''). Then,

- it is clear that $\lambda_{G'}(t, T - t) = r(t)$ for each $t \in T - t_0 - t' - t''$,
- $\lambda_{G'}(t_0, T - t_0) = r(t_0) - 1$ by the existence of $P_2, \ldots, P_{r(t_0)}$,
- $\lambda_{G'}(t', T - t') = r(t') - 1$ by the existence of $Q_2, \ldots, Q_{r(t')}$, and
- $\lambda_{G'}(t'', T - t'') = r(t'') - 1$ by the existence of $R_2, \ldots, R_{r(t'')}$.

Note that $Q_2, \ldots, Q_{r(t')}, R_2, \ldots, R_{r(t'')}$ do not share an edge with $P_1[t_0, v_{i-1}]$, since they are paths in G_{i-1}. By induction hypothesis, G' contains a set F' of trees, and F' can be extended in $G - E(P)$ by using subpaths of $Q_2, \ldots, Q_{r(t')}$ and $R_2, \ldots, R_{r(t'')}$. Therefore, these objects together with P form a desired set of trees in G. $\square$

5 Concluding remarks

Note that in Problem GTBP we allow an arbitrary number of parallel copies of any edge in G, therefore our problem is an **uncapacitated network design problem**. A natural capacitated extension of our problem would be the following (we only formulate the minimum cost version here).

Problem 4. *In the minimum cost version of Problem 1, find a graph $G = (V, E)$ also satisfying that the number of parallel copies of an edge $e \in E$ is at most some capacity $cap(e) \in \mathbb{Z}_+$, that is given in advance.*

This problem can also be seen as a **minimum cost subgraph problem** by introducing a supply graph with edge-multiplicities $cap(uv)$ for every $u, v \in V$. Note that Problem 1 is the special case of this problem by setting $cap(uv) = \max\{r(t) : t \in T\}$ for every pair $u, v \in V$. We could not extend our results to Problem 4. The problem is open even if G_0 is the empty graph. Note that Jain's framework implies a 2-approximation algorithm for this problem in the case when the capacities do not exceed some fixed constant (that is not part of the input).

Acknowledgement

We would like to thank András Frank for telling us about this problem and to Gyula Pap for his helpful comments.

References

[1] Elliot Anshelevich and Adriana Karagiozova, *Terminal backup, 3d matching, and covering cubic graphs*, Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing, ACM, 2007, pp. 391–400.

[2] Jørgen Bang-Jensen, András Frank, and Bill Jackson, *Preserving and increasing local edge-connectivity in mixed graphs*, SIAM J. Discrete Math. **8** (1995), no. 2, 155–178.

[3] Jørgen Bang-Jensen and Bill Jackson, *Augmenting hypergraphs by edges of size two*, Math. Program. **84** (1999), no. 3, 467–481.

[4] András A. Benczúr and András Frank, *Covering symmetric supermodular functions by graphs*, Math. Program. **84** (1999), no. 3, 483–503.

[5] Attila Bernáth and Tamás Király, *A unifying approach to splitting-off*, Combinatorica **32** (2012), 373–401.

[6] Boris V. Cherkassky, *A solution of a problem on multicommodity flows in a network*, Ekonomika i Matematicheskie Metody **13** (1977), no. 1, 143–151.

[7] András Frank, *Augmenting graphs to meet edge-connectivity requirements*, SIAM J. Discrete Math. **5** (1992), no. 1, 25–53.

[8] ———, *Connections in combinatorial optimization*, Oxford University Press, 2011.

[9] András Frank and Tamás Király, *A survey on covering supermodular functions*, Research Trends in Combinatorial Optimization (W.J. Cook, L. Lovász, and J. Vygen, eds.), Springer, 2009, pp. 87–126.

[10] Roland Grappe and Zoltán Szigeti, *Note: Covering symmetric semi-monotone functions*, Discrete Appl. Math. **156** (2008), no. 1, 138–144.

[11] Toshimasa Ishii and Masayuki Hagiwara, *Minimum augmentation of local edge-connectivity between vertices and vertex subsets in undirected graphs*, Discrete Appl. Math. **154** (2006), no. 16, 2307–2329.

[12] Kamal Jain, *A factor 2 approximation algorithm for the generalized steiner network problem*, Combinatorica **21** (2001), no. 1, 39–60.

[13] Tamás Király, *Covering symmetric supermodular functions by uniform hypergraphs*, J. Combin. Theory Ser. B **91** (2004), no. 2, 185–200.

[14] Zoltán Király, Ben Cosh, and Bill Jackson, *Local edge-connectivity augmentation in hypergraphs is NP-complete*, Discrete Applied Mathematics **158** (2010), no. 6, 723–727.

[15] László Lovász, *On some connectivity properties of Eulerian graphs*, Acta Mathematica Hungarica **28** (1976), no. 1, 129–138.

[16] W. Mader, *Über die Maximalzahl kantendisjunkter A-Wege*, Arch. Math. **30** (1978), no. 3, 325–336.

[17] Karl Menger, *Zur allgemeinen Kurventheorie*, Fundamenta Mathematicae **10** (1927), 96–115.

[18] Hiroyoshi Miwa and Hiro Ito, *NA-edge-connectivity augmentation problems by adding edges*, J. Oper. Res. Soc. Japan **47** (2004), no. 4, 224–243.

[19] Zeev Nutov, *Approximating connectivity augmentation problems*, Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, 2005, pp. 176–185.

[20] Alexander Schrijver, *Combinatorial optimization. Polyhedra and efficiency.*, Algorithms and Combinatorics, vol. 24, Springer-Verlag, Berlin, 2003.

[21] Zoltán Szigeti, *Hypergraph connectivity augmentation*, Math. Program. **84** (1999), no. 3, 519–527.

[22] ______, *Edge-connectivity augmentations of graphs and hypergraphs*, Research Trends in Combinatorial Optimization (W.J. Cook, L. Lovász, and J. Vygen, eds.), Springer, 2009, pp. 483–521.

[23] Toshimasa Watanabe and Akira Nakamura, *Edge-connectivity augmentation problems*, J. Comput. System Sci. **35** (1987), no. 1, 96–144.

[24] Dahai Xu, Elliot Anshelevich, and Mung Chiang, *On survivable access network design: Complexity and algorithms*, Proceedings of the Twenty-seventh Conference on Computer Communications, IEEE, 2008, pp. 186–190.