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, Yusuke Kobayashi, and
Tatsuya Matsuoka

December 2013
Revised June 2014

The Generalized Terminal Backup Problem

Attila Bernáth*, Yusuke Kobayashi**, and Tatsuya Matsuoka***

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 $m(v)$ 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.

In the first case we solve 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 $w : V \to \mathbb{R}_+$.

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 algorithms run in strongly polynomial time for both problems.

1 Introduction

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

*Hungarian Academy of Sciences, Institute for Computer Science and Control (MTA-SZTAKI), Budapest, Hungary. Research supported by the ERC StG project PAAl no. 259515. E-mail: bernath@cs.elte.hu

**University of Tokyo, Tokyo 113-8656, Japan. Supported by JST, ERATO, Kawarabayashi Large Graph Project. Supported by by KAKENHI Grant Number 24106002, 24700004. E-mail: kobayashi@mist.i.u-tokyo.ac.jp

***University of Tokyo, Tokyo 113-8656, Japan. Supported by JST, ERATO, Kawarabayashi Large Graph Project. E-mail: tatsuya_matsuoka@mist.i.u-tokyo.ac.jp

December 2013. Revised June 2014

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 [25], 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 [8], 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 [15]. Ishii and Hagiwara [12] 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 [19] that this problem is NP-complete, however, the authors of [12] have given a polynomial algorithm solving it if $r(W) \geq 2$ for every $W \in \mathcal{W}$ (see also [11]). More generalizations, abstract versions and related results were given by [4, 5, 14, 23], good surveys can be found in [10, 24].

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 [13] 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 $m(v)$ 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**: we start with the minimum cost version for $c \equiv 1$ and solve 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 24, a variant of Theorem 2, 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 2 (Lovász [16] and Cherkassky [7]). *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 24). Theorem 2 was generalized in many directions, for example Mader [17] 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 [22, Corollary 73.2b]). We could not see our Lemma 24 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 then proving the splitting-off theorem and exploring its consequences in subsection 3.3. In Section 4 we solve the survivable network design problem: in subsection 4.1 we prove the main ingredient of our solution, Lemma 24, and reduce the problem to a generalization of the simplex matching problem, in subsection 4.2 we give a pesudo-polynomial time algorithm, and in subsection 4.3 we improve the running time to strongly polynomial. 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 [9]. 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** (or sometimes **multihypergraph**) 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** and the multiplicity of a hyperedge is represented as a binary number. 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 (the term **multigraph** is used as a synonym of the term graph). A **simple hypergraph** is a hypergraph in which every hyperedge has multiplicity 1. 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 $(s = v_0, e_1, v_1, e_2, \ldots, e_k, v_k = t)$, such that $v_{i-1}, v_i \in e_i$ for all i between 1 and k. 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 3 (Menger's Theorem for hypergraphs [18]). *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

A function $p : 2^V \to \mathbb{R} \cup \{-\infty\}$ is called a **set function**. We say that a graph G **covers** a set function p if $d_G(X) \geq p(X)$ holds for every $X \subseteq V$. Problem GTBP can be formulated as the problem of covering a skew-supermodular set function by a graph, as will be shown in Section 3. 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 4 ([5]). *The symmetrized of a skew-supermodular function is (symmetric and) skew supermodular.*

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

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

This polyhedron will be used to characterize the feasibility of the degree-specified version of Problem GTBP (see Section 3.3). An important property of $C(p)$ is the following.

Theorem 5 ([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 5. See details about contrapolymatroids in [22].

Corollary 6. *If p is as in Theorem 5 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 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). One can observe that this is indeed the usual notion of splitting-off: if we introduce a graph $H = (V + s, E)$ with a new node s, every edge of E incident to s and $m(v)$ parallel edges between s and v for any $v \in V$, then we are back at the well-known (undirected) splitting-off operation (as introduced in Section 8.1 of [9]). If $m'(X) \geq p'(X)$ holds for any $X \subseteq V$, then we say that the splitting off is (p, m)-**admissible**. A set X is called (p, m)-**tight**, if $m(X) = p(X)$, and it is called (p, m)-**dangerous** if $m(X) - p(X) \leq 1$. We will only say admissible, tight and dangerous, if p and m are clear from the context. The following claim is well-known.

Claim 7 (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.*

Contraction of tight sets is a standard technique in splitting-off proofs (see for example [5], where contraction is explained in detail).

Lemma 8 (see e.g. [5]). *Let $u, v \in V$ with $m(u), m(v) > 0$. If we contract a tight set $X \subseteq V$, then the splitting at u' and v' is admissible in the contracted instance if and only if the splitting at u and v is admissible in the original instance (where u' (v') is the contracted image of u (v, respectively)).*

We will also use the following lemma.

Lemma 9 ([5, 20]). *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. 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 $v \in V$).

3.1 Notation and the minimum cardinality version

Consider Problem GTBP above. We introduce the following notation. For any terminal $t \in T$ let $d_t = \min\{d_{G_0}(X) : X \cap T = \{t\}\}$ and we say that $X \subseteq V$ is a **t-mincut (in G_0)** if $d_{G_0}(X) = d_t$ and $X \cap T = \{t\}$. We can easily see the following.

Claim 10. *The intersection and the union of two t-mincuts are again t-mincuts.*

For a terminal $t \in T$ let X_t (Y_t) denote the inclusionwise minimal (maximal, respectively) t-mincut. By Claim 10, the sets X_t and Y_t are well defined.

Lemma 11. *For two different terminals $t, t' \in T$ we have $X_t \cap Y = \emptyset$, where Y is an arbitrary t'-mincut (and X_t is defined above). Consequently, $\{X_t : t \in T\}$ is a subpartition of V.*

Proof. Assume $X_t \cap Y \neq \emptyset$. Since

$$d_t + d_{t'} = d_{G_0}(X_t) + d_{G_0}(Y) \geq d_{G_0}(X_t - Y) + d_{G_0}(Y - X_t) \geq d_t + d_{t'},$$

we have $d_{G_0}(X_t - Y) = d_t$, which contradicts the minimality of X_t. □

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

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

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 12. *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 4. Observe that $R(X) = R^s(X)$, unless $|T - X| = 1$. Let finally $p(X) = R^s(X) - d_{G_0}(X)$ for any $X \subseteq V$ (called the **deficiency function** for this instance of Problem GTBP), which is symmetric and skew-supermodular. Note that G covers $R - d_{G_0}$ if and only if G covers p. Notice that $p(X) = r(t) - d_t$ for any t-mincut X if $|T| \geq 3$.

By using these notations, we can solve the minimum cardinality version.

Theorem 13. *Suppose that $p(X_{t_1}) = \max_{t \in T} p(X_t)$ for some $t_1 \in T$. The minimum number of edges of a graph G that satisfies the requirements of Problem GTBP is equal to $\gamma = \max\{p(X_{t_1}), \lceil \frac{1}{2} \sum\{p(X_t) : t \in T, p(X_t) > 0\} \rceil\}$.*

Proof. It is clear from Lemma 11 that γ is a lower bound. On the other hand, let us find an arbitrary loopless graph G on nodeset T such that $d_G(t) \geq p(X_t)$ for every $t \in T$ and $|E(G)| = \gamma$. Such a graph exists and satisfies our requirements, since $\lambda_G(t, T - t) \geq p(X_t)$ for every $t \in T$. □

3.2 Properties of the contrapolymatroid

In this subsection, we show some properties of the contrapolymatroid $C(p)$, where $p = R^s - d_{G_0}$ is defined as in the previous subsection.

Lemma 14. *Suppose that $p(X_{t_1}) = \max_{t\in T} p(X_t)$ for some $t_1 \in T$. Then, we have $\min\{1 \cdot x : x \in C(p)\} = \max\{p(X_{t_1}) + p(V - X_{t_1}), \sum\{p(X_t) : t \in T, p(X_t) > 0\}\}$.*

Proof. Clearly, $\max\{p(X) : X \cap T = \{t\}\} = p(X_t)$ and $\max\{p(X) : |X \cap T| = 1\} = p(X_{t_1})$. Let $Z_1, Z_2, \dots, Z_k$ be a subpartition attaining $\sum_{i=1}^k p(Z_i) = \max\{\sum_{X\in\mathcal{X}} p(X) : \mathcal{X}$ is a subpartition of $V\} = \min\{1 \cdot x : x \in C(p)\}$. By the definition of the function p, for every $i = 1, 2, \dots, k$ either $|T \cap Z_i| = 1$, or $|T - Z_i| = 1$. Assume first that there exists an i such that $|T - Z_i| = 1$: say this holds for $i = 1$. In this case $k \le 2$ and by the symmetry of p we have $p(Z_1) = p(Z_2)$, and the best value we can get for $p(Z_2)$ is $p(X_{t_1})$, that is $\sum_{i=1}^k p(Z_i) = p(X_{t_1}) + p(V - X_{t_1})$ in this case. The other case is when $|T \cap Z_i| = 1$ for every $i = 1, 2, \dots, k$. In this case $\sum_{i=1}^k p(Z_i) = \sum\{p(X_t) : t \in T, p(X_t) > 0\}$, using that $\{X_t : t \in T\}$ is a subpartition. □

By observing that $p(X_{t_1}) = p(V - X_{t_1})$, we have the following as a corollary.

Corollary 15. *If $\min\{1 \cdot x : x \in C(p)\}$ is odd, then $p(X_{t'}) < \sum\{p(X_t) : t \in T - t', p(X_t) > 0\}$ for any $t' \in T$ and $\min\{1\cdot x : x \in C(p)\} = \sum\{p(X_t) : t \in T, p(X_t) > 0\}$.*

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 6, and in our Splitting-off Theorem (see Section 3.3); 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 $\lambda_H(t, T - t) \ge r(t)$ 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 $x(Z) < p(Z)$.

3.3 The splitting-off theorem and its consequences

In this section we solve the degree-specified version of Problem GTBP. For an instance of this problem, recall that $p : 2^V \to \mathbb{Z} \cup \{-\infty\}$ is defined by $p(X) = R^s(X) - d_{G_0}(X)$ for $X \subseteq V$. We start with the following splitting-off result.

Theorem 16. *If $m \in C(p) \cap \mathbb{Z}^V$ is minimal and $0 < m(V) \neq 3$ then there exists an admissible splitting-off.*

Proof. If $m(V) = 1$ then m cannot be minimal, if $m(V) = 2$ then clearly there exists an admissible splitting-off, so we can assume that $m(V) \geq 4$.

If $\max\{p(X) : X \subseteq V\} > 1$ then there exists an admissible splitting-off by Lemma 9. So we can assume that $p \leq 1$. By Corollary 6, there exists a subpartition $\mathcal{X}$ of V such that X is tight and $p(X) > 0$ for each $X \in \mathcal{X}$. Since $p \leq 1$ and $m(V) \geq 4$, we can assume that there exist tight sets $X_1, X_2, X_3, X_4 \in \mathcal{X}$ with $p(X_i) = m(X_i) = 1$ ($i = 1, 2, 3, 4$). Choose an arbitrary $x \in X_1$ and $y \in X_2$ with $m(x) > 0, m(y) > 0$, 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 $m(X) \leq p(X) + 1$. By Lemma 8, we can assume that $X_i = \{t_i\}$ for $i = 1, 2, 3, 4$, implying that $t_1, t_2 \in X$. This implies (by the definition of the function p) that $|T - X| = 1$, so we can assume that $t_3 \in X$ holds, too. But then $m(X) \geq 3$, so X cannot be dangerous, since $p(X) \leq 1$, a contradiction. □

Corollary 17. *If $m \in \mathbb{Z}^V$ is a minimal member of $C(p)$, and $m(V)$ is even then there exists a graph G with $d_G(v) = m(v)$ at every $v \in V$ satisfying the requirements of Problem GTBP.*

Proof. Clearly follows from Theorem 16 by induction. □

Now we are ready to give a solution of Problem DS-GTBP. If a specified degree of some vertex is too large compared to other degrees (i.e., $m(v) > m(V - v)$ 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, for a hypergraph $G_0 = (V, \mathcal{E}_0)$ and a set $T \subseteq V$ with $|T| \geq 2$, $X \subseteq V$ is a t-mincut (in G_0) if $d_{G_0}(X) = d_t := \min\{d_{G_0}(X) : X \cap T = \{t\}\}$ and $X \cap T = \{t\}$. The following theorem gives a solution of Problem DS-GTBP.[1]

Theorem 18. *Assume we are given an instance of Problem DS-GTBP (that is, a hypergraph $G_0 = (V, \mathcal{E}_0)$, a set of at least two terminals $T \subseteq V$, requirements $r : T \to \mathbb{Z}_+$, and degree-specifications $m : V \to \mathbb{Z}_+$), there exists a solution of this problem (that is a multigraph $G = (V, E)$ with $d_G^+(v) = m(v)$ at every $v \in V$ and $\lambda_{G_0+G}(t, T - t) \geq r(t)$ for every $t \in T$) if and only if*

1. *$m \in C(p) \cap \mathbb{Z}_+^V$, $m(V)$ is even, and*
2. *at least one of the following holds:*
   - *(a) $\min\{1 \cdot x : x \in C(p)\}$ is even, or*
   - *(b) there exists a $t_0 \in T$ such that $m(X_{t_0}) > \max\{p(X_{t_0}), 0\}$, or*
   - *(c) there exist a $y \in V - \bigcup_{t \in T} X_t$ and a $t_0 \in T$ such that $m(y) > 0, p(X_{t_0}) > 0$, and any t_0-mincut X containing y satisfies $m(X) > p(X) + 2$.*

[1] We have to mention that in the SODA version of this paper there was an error: unfortunately Theorem 3.2 in [6] is not true. The correct splitting-off statement is given in Theorem 18.

Proof. Notice that the intersection of two t_0-mincuts is again a t_0-mincut, so condition (2c) can be reformulated as follows: either there is no t_0-mincut containing y, or $m(X_0) > p(X_0) + 2$ for the inclusionwise minimal t_0-mincut X_0 containing y.

Necessity: Assume that the required solution G exists but the conditions are not met. Clearly, the existence of G implies that $m \in C(p)$ and that $m(V)$ is even, therefore $\min\{1 \cdot x : x \in C(p)\}$ is odd, $m(X_t) = \max\{p(X_t), 0\}$ for every $t \in T$, and for every $y \in V - \bigcup_{t \in T} X_t$ and $t \in T$ such that $m(y) > 0, p(X_t) > 0$, there exists a t-mincut X containing y such that $m(X) \leq p(X) + 2$. We get a contradiction by induction on the number of edges in G. The base case $E(G) = \emptyset$ is obvious, so assume that $E(G) \neq \emptyset$. Since $\min\{1 \cdot x : x \in C(p)\}$ is odd and $m(X_t) = \max\{p(X_t), 0\}$ for every $t \in T$, there exist a $t' \in T$ and a $y \in V - \bigcup_{t \in T} X_t$ such that there exists an edge xy in G for some $x \in X_{t'}$. Let X_0 be the inclusionwise minimal t'-mincut in G_0 containing y: by our conditions, $m(X_0) = p(X_0) + 2$ must hold (we use that the splitting-off at x and y must be admissible by the existence of G, therefore $m(X_0) \leq p(X_0) + 1$ cannot be the case).

Consider the following modified instance of Problem DS-GTBP: let $G_0' = G_0 + xy$, $m' = m - \chi_{\{x,y\}}$ and every other parameter is as in the original instance (that is, we start with a splitting-off at x and y), and let $G' = G - xy$. Let p' be the deficiency function for this modified instance (that is, $p' = R^s - d_{G_0'}$). We show that the conditions fail for this instance, and G' is a valid solution for this instance, leading to a contradiction by induction. Notice that the existence of G' implies that $m' \in C(p') \cap \mathbb{Z}_+^V$ (and clearly, $m'(V)$ is even). The inclusionwise minimal t'-mincut in G_0' is X_0, and $p'(X_0) = m'(X_0)$. Furthermore, X_t is the inclusionwise minimal t-mincut in G' for every $t \in T - t'$, and $p'(X_t) = p(X_t)$ and $m(X_t) = m'(X_t)$ for every $t \in T - t'$. This shows that condition (2b) fails also in the obtained instance. In what follows, we show that conditions (2a) and (2c) fail, respectively.

- **Condition (2a).** Since $p \geq p'$, $C(p) \subseteq C(p')$, therefore $\min\{1 \cdot x : x \in C(p)\} \geq \min\{1 \cdot x : x \in C(p')\}$. On the other hand, by Lemma 14, $\min\{1 \cdot x : x \in C(p')\} \geq \sum\{p'(X_t) : t \in T - t_0, p'(X_t) > 0\} + p'(X_0) = \sum\{p(X_t) : t \in T, p(X_t) > 0\} = \min\{1 \cdot x : x \in C(p)\}$, therefore equality has to hold here, so $\min\{1 \cdot x : x \in C(p')\}$ is odd.

- **Condition (2c).** Let $z \in V - (X_0 \cup \bigcup_{t \in T - t_0} X_t)$ with $m'(z) > 0$.

  If Z_0 is the smallest t_0-mincut containing z in G_0, then $m(Z_0) \leq p(Z_0) + 2$, since the original instance does not satisfy condition (2c). Furthermore, we have $m(X_0) = p(X_0) + 2$, $m(X_0 \cap Z_0) \geq m(X_t) = p(X_t)$, and $p(Z_0) = p(X_0) = p(X_t)$. By combining these inequalities, we have $m(X_0 \cup Z_0) \leq p(X_t) + 4$. Since $X_0 \cup Z_0$ is a t_0-mincut containing z in G_0' and $\{x, y\} \subseteq X_0 \cup Z_0$, we obtain $m'(X_0 \cup Z_0) \leq p'(X_0 \cup Z_0) + 2$.

  Let $t \in T - t_0$ with $p'(X_t) = p(X_t) > 0$ and let Z be the smallest t-mincut containing z in G_0. Observe that Z is disjoint from X_0 (use that $d_{G_0}(X_0) + d_{G_0}(Z) \geq d_{G_0}(X_0 - Z) + d_{G_0}(Z - X_0)$, and that $z \in Z - X_0$). This gives that $m'(Z) = m(Z) \leq p(Z) + 2 = p'(Z) + 2$.

By the above arguments, the conditions fail in the obtained instance, which completes the proof of necessity.

Sufficiency: Assume that the conditions hold. If $\min\{1 \cdot x : x \in C(p)\}$ is even then we are done by Corollary 17, so assume that this is not the case. If there exists a $t_0 \in T$ such that $m(X_{t_0}) > \max\{p(X_{t_0}), 0\}$ then we can modify the instance at hand as follows: we increase $r(t_0)$ by $\max\{1, 1 - p(X_{t_0})\}$ (and we leave every other parameter unchanged), and we arrive at the previous case for this modified instance. Finally, if $\min\{1 \cdot x : x \in C(p)\}$ is odd and $m(X_t) = \max\{p(X_t), 0\}$ for every $t \in T$ then, by Condition (2c), there exists a $y \in V - \bigcup_{t \in T} X_t$ and a $t_0 \in T$ such that $m(y) > 0, p(X_{t_0}) > 0$ and any t_0-mincut X containing y satisfies $m(X) > p(X) + 2$. Choose an arbitrary $x \in X_{t_0}$ with $m(x) > 0$ and consider the following modified instance of Problem DS-GTBP: let $G'_0 = G_0 + xy$, $m' = m - \chi_{\{x,y\}}$ and every other parameter is as in the original instance (that is, we start with a splitting-off at x and y). Let p' be the deficiency function for this modified instance.

Claim 19. *This is an admissible splitting-off, that is $m' \in C(p')$.*

Proof. Assume indirectly that there exists a set $X \subseteq V$ with $x, y \in X$ and $m(X) \leq p(X) + 1$. By Lemma 8, we can assume that $X_t = \{t\}$ is a singleton for every $t \in T$. Since X contains t_0, we have either $T - X = \{t'\}$ for some $t' \in T - t_0$ or $X \cap T = \{t_0\}$ by the definition of p.

First, suppose that $T - X = \{t'\}$ for some $t' \in T$. Since $p(X) = r(t') - d_{G_0}(X) \leq p(X_{t'})$ and $m(X) \geq \sum\{m(X_t) : t \in T - t'\} + 1 = \sum\{p(X_t) : t \in T - t', p(X_t) > 0\} + 1 > p(X_{t'}) + 1$, where the last inequality follows from Corollary 15, X cannot be dangerous.

Second, suppose that $X \cap T = \{t_0\}$, which implies that $p(X) = r(t_0) - d_{G_0}(X) \leq p(X_{t_0})$. Since $m(X) \geq m(X_{t_0}) + 1 = p(X_{t_0}) + 1 \geq p(X) + 1$, the only way X can be dangerous is that X is a t_0-mincut in G_0 containing y with $m(X) = p(X) + 1$, contradicting condition (2c). □

We now finish the proof of the sufficiency by distinguishing the following two cases.

- **Case 1.** There exists no t_0-mincut in G_0 containing y. In this case, X_{t_0} is a t_0-mincut in G'_0 and $\min\{1 \cdot x : x \in C(p')\}$ is even, so we are done by induction.

- **Case 2.** There exists a t_0-mincut in G_0 containing y: let X_0 be the inclusionwise minimal t_0-mincut in G_0 containing y. In this case the inclusionwise minimal t_0-mincut in G'_0 is X_0 and $m'(X_0) = m(X_0) - 2 > p(X_0) = p'(X_0) > 0$ by our conditions, so we are again done by induction.

□

By using this theorem, we can solve Problem MC-GTBP in polynomial time if the weight function is node-induced. Recall that Y_t is defined as the inclusionwise maximal vertex set with $Y_t \cap T = \{t\}$ and $d_{G_0}(Y_t) = d_t$.

Theorem 20. *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 6, we can find a vector $m \in C(p) \cap \mathbb{Z}_+^V$ minimizing $\sum_{v \in V} w(v)m(v)$. If $m(V)$ is even, then there is an optimal solution G with $d_G = m$ by Theorem 18. Otherwise, by the conditions (2b) and (2c) of Theorem 18, there exists an optimal solution G of the problem such that either

- there exists $v \in \bigcup_{t \in T} X_t$ such that $d_G = m + \chi_v$,
- there exists $v \in V - Y_t$ for some $t \in T$ with $p(X_t) > 0$ such that $d_G = m + \chi_v$, or
- there exists $v \in Y_t - X_t$ for some $t \in T$ with $p(X_t) > 0$ such that $d_G = m + 3\chi_v$.

Note that the first case corresponds to the condition (2b) and the second and third cases correspond to the condition (2c). With this observation, when $m(V)$ is odd, we can solve the problem by the following algorithm. Let $V_1 = \bigcap\{Y_t : t \in T, p(Y_t) > 0\}$. Note that there exist at least two terminals $t \in T$ with $p(Y_t) > 0$, since $m(V)$ is odd, therefore $V_1 \cap X_t = \emptyset$ for every $t \in T$ (that is, $V - V_1 = (\bigcup_{t \in T} X_t) \cup (\bigcup_{t \in T, p(X_t) > 0}(V - Y_t))$). Let x be the vertex in $V - V_1$ with the smallest weight, and let y be the vertex in V_1 with the smallest weight. Define $m' \in \mathbb{Z}_+^V$ by $m' := m + \chi_x$ if $w(x) \leq 3w(y)$ and $m' := m + 3\chi_y$ otherwise. By Theorem 18, we can find a graph G with $d_G(v) = m'(v)$ at every $v \in V$ satisfying the requirements of Problem GTBP, which is a desired graph. □

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 20: given node weights $w(v)$ for every node $v \in V$, 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 [23], 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 the minimum cost version of Problem GTBP in the special case when G_0 is the empty graph. Let us formulate this problem separately.

Problem 21. *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 21 is polynomially solvable if $T = V$, because now the question is to find a smallest cost multigraph $G = (V, E)$ 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 with

$b = r$ (which is equivalent to the maximum-weight b-matching problem with a simple reduction, see [22, Section 34.4]).

We also note that the special case $r \equiv 1$ of Problem 21 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 solution for the Terminal Backup Problem in [1] is based on a polynomial time algorithm for the **Simplex Matching Problem**. To formulate this problem let us give some definitions. A hypergraph that only has hyperedges of size 2 and 3 is called a **2-3 hypergraph**. A **perfect matching** in a hypergraph $H = (U, \mathcal{E})$ is a subset of hyperedges $\mathcal{F} \subseteq \mathcal{E}$ so that each node is contained in exactly one member of $\mathcal{F}$. In an instance of the Simplex Matching Problem we are given a simple 2-3 hypergraph $H = (U, \mathcal{E})$ with edge costs $\gamma : \mathcal{E} \to \mathbb{R}_+$, 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

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

To simplify the terminology, the Simplex Matching Problem is meant as a problem **with the simplex condition**. The main theorem in [1] is as follows.

Theorem 22 (Anshelevich and Karagiozova [1], see also [21])**.** *There is a polynomial time algorithm for the Simplex Matching Problem.*

In this paper we consider and solve the following generalization of the Simplex Matching Problem that we call the **Simplex b-Edge-Cover Problem**.

Problem 23. *Let $H = (T, \mathcal{E})$ be a simple 2-3 hypergraph, let $\gamma : \mathcal{E} \to \mathbb{R}_+$ be a cost function satisfying the simplex condition, and let $b(t) \in \mathbb{Z}_+$ be a requirement for $t \in T$. Find a minimum cost multihypergraph $H' = (T, \mathcal{F})$ such that $\mathcal{F}$ is a multiset of $\mathcal{E}$ and $d_{H'}(t) \geq b(t)$ for any $t \in T$.*

4.1 Reduction of Problem 21 to the Simplex b-Edge-Cover Problem

In this section, we show how to reduce Problem 21 to the simplex b-edge-cover problem. We start with the following lemma which will be used in solving the survivable network design problem in the next section. This lemma is a variant of Theorem 2 and we think that it is of independent interest. Given a graph $G = (V, E)$ and some $T \subseteq V$, a **T-path** is a path $P \subseteq E$ so that its endpoints are distinct nodes of T. Similarly, a **T-3-tree** is a tree $P \subseteq E$ that does not necessarily span V, has exactly 3 leaves, these leaves are all in T, and P is not incident with other nodes in T. The unique node with degree 3 in a T-3-tree is called the **hub-node** of the T-3-tree: by the previous definition, this node is not in T.

Lemma 24. *Given a graph $G = (V, E)$ and a set $T \subseteq V$, we can find in polynomial time a set F of mutually edge-disjoint T-paths and T-3-trees so that each $t \in T$ is incident with $\lambda_G(t, T - t)$ members of F and each $v \in V - T$ is the hub node of at most one T-3-tree in F.*

Proof. If there is an edge e in G so that

$$\lambda_{G'}(t, T - t) = \lambda_G(t, T - t) \text{ for every } t \in T \tag{1}$$

holds for $G' = G - e$, then the proof is ready by induction. Similarly, if there exist a pair of edges vx, vy incident to some node $v \in V - T$ so that (1) holds for the graph $G' = (V, E - \{vx, vy\} + \{xy\})$ that we obtain after splitting off the pair vx, vy then the proof is ready by induction. So assume that neither a removable edge, nor an admissible splitting-off exists. For every $v \in V - T$, by applying Theorem 16 for the graph $G - v$, in which $m(x) = d_G(x, v)$ for every $x \in V - v$ and $r(t) = \lambda_G(t, T - t)$ for every $t \in T$, we have $d_G(v) \in \{0, 3\}$.

We claim that there is no edge between two vertices $u, v \in V - T$: this claim clearly finishes the induction. Assume indirectly that uv is such an edge, so $d_G(u) = d_G(v) = 3$. Consider an instance of Problem DS-GTBP defined by the graph $G_0 = G - u$, $m(x) = d_G(x, u)$ for every $x \in V - u$, and $r(t) = \lambda_G(t, T - t)$ for every $t \in T$, and let p be the deficiency function defined by this instance. By Lemma 14 (applied for this instance), we get that there exists a set $X_0 \subseteq V - u$ such that $v \in X_0$, $X_0 \cap T = \{t_0\}$ for some $t_0 \in T$, and $d_G(X_0) = \lambda_G(t_0, T - t_0)$ (the (p, m)-tight set containing v). Now consider the following instance of Problem DS-GTBP. Let $G'_0 = G - v$, $m'(x) = d_G(x, v)$ for every $x \in V - v$, and $r(t) = \lambda_G(t, T - t)$ for every $t \in T$. Let p' be deficiency function defined by this instance: since there is no (p', m')-admissible splitting-off, $p' \leq 1$ by Lemma 9. This together with $d_G(V - X_0) = \lambda_G(t_0, T - t_0)$ implies that $m'(V - X_0) = p'(V - X_0) = 1$ (that is, the only G-neighbour of v in $V - X_0$ is u). By our assumptions, $m'(V - v) = 3$, so let $m'(x_1) = m'(x_2) = 1$ for some distinct $x_1, x_2 \notin V - X_0$, and let X_i be the smallest (p', m')-tight sets containing x_i for $i = 1, 2$. Note that $V - X_0, X_1, X_2$ are mutually disjoint, contradicting that $(V - X_0) \cap T = T - \{t_0\}$ and $|X_i \cap T| \geq 1$ for $i = 1, 2$. □

We note that another proof of this lemma is given in the conference version [6, Lemma 4.2]. We also note that we will not utilize below the fact that every node of $V - T$ is the hub-node of at most one T-3-tree in the decomposition given by Lemma 24. Now we show the reduction of Problem 21 to the simplex b-edge-cover problem.

Lemma 25. *We can reduce Problem 21 to the simplex b-edge-cover problem (Problem 23) in polynomial time.*

Proof. For a given instance I of Problem 21, define the 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 $\binom{T}{3} = \{\{t_1, t_2, t_3\} \mid t_1, t_2, t_3 \in T,\ t_1 \neq t_2 \neq t_3 \neq t_1\}$, let $b = r$, 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). Since 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, it can be computed in polynomial time by guessing the hub vertex v and using a shortest path algorithm. The family $\mathcal{E}$ and the cost function γ define an instance I' of the simplex b-edge-cover problem (note that the simplex condition naturally holds).

Clearly, any solution of instance I' of the simplex b-edge-cover problem gives rise to a solution of instance I of Problem 21 with the same cost. On the other hand, if we have a solution of instance I of Problem 21, then by Lemma 24 it defines edge-disjoint T-paths and T-3-trees: substitute each T-path with a minimum c-cost T-path between the same nodes, and substitute each T-3-tree with a minimum c-cost Steiner tree with the same leaves. This way we obtain a solution of instance I' of the simplex b-edge-cover problem with γ-cost not higher than the c-cost of the solution instance I of Problem 21 that we started with. This relation finishes the proof of this lemma. □

We note that the corresponding result is given in [26] for the special case $r \equiv 1$.

In the rest of the paper we give a polynomial time algorithm for the simplex b-edge-cover problem. This will be done in two steps: in Section 4.2 we obtain a pseudo-polynomial time algorithm, and based on this algorithm we show in Section 4.3 how to obtain a strongly polynomial time algorithm.

4.2 Pseudo-polynomial time algorithm

Suppose that we are given an instance of the simplex b-edge-cover problem (Problem 23) consisting of a simple 2-3 hypergraph $H = (T, \mathcal{E})$, requirement $b : T \to \mathbb{Z}_+$ and cost $\gamma : \mathcal{E} \to \mathbb{R}_+$. Let us introduce the notation $B = \sum_{t \in T} b(t)$. Our pseudo-polynomial time algorithm for this problem is as follows.

Pseudo-polynomial time algorithm for Problem 23

Step 1 Construct a Simplex Matching Problem instance consisting of the simple 2-3 hypergraph $(T^+, \mathcal{E}_+ \cup \mathcal{E}_0)$ and costs as follows.

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

Step 1-2 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 $i, j \in \{1, 2, \ldots, B+2\}$. 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 $i, j, k \in \{1, 2, \ldots, B + 2\}$.

Step 1-3 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 $b(t) + 1 \le i < j \le B + 2$, and add edges $\{t^{(i)}, t^{(j)}, t^{(k)}\}$ with cost 0 for $b(t) + 1 \le i < j < k \le B + 2$.

Step 2 Solve the obtained Simplex Matching Problem instance using Theorem 22. Then, from the optimal solution of the Simplex Matching Problem, we can construct a solution of Problem 23 by ignoring the hyperedges in $\mathcal{E}_0$ and contracting $t^{(1)}, t^{(2)}, \ldots, t^{(B+2)}$ to a single vertex for each $t \in T$.

Before proving the correctness of this algorithm, we give a small claim on the optimal solutions of the simplex b-edge-cover problem.

Claim 26. *Problem 23 always has an optimal solution $H' = (T, \mathcal{F})$ such that $d_{H'}(t) \in \{b(t), b(t)+1, \ldots, B\}$ for any $t \in T$, where $B = \sum_{t \in T} b(t)$.*

Proof. It suffices to show that any minimal solution $H' = (T, \mathcal{F})$ of Problem 23 satisfies that $|\mathcal{F}| \leq B$. Define $\mathcal{F}_{\text{tight}} := \{e \in \mathcal{F}$: there exists $t \in T$ such that $d_{H'}(t) = b(t)$ and e enters $t\}$. By the minimality of $\mathcal{F}$, we have $\mathcal{F} = \mathcal{F}_{\text{tight}}$. Therefore, we have

$$|\mathcal{F}| = |\mathcal{F}_{\text{tight}}| \leq \sum_{t \in T:\, d_{H'}(t) = b(t)} d_{H'}(t) \leq \sum_{t \in T} b(t) = B.$$

□

Now we are ready to prove the following theorem, which will be improved in Section 4.3.

Theorem 27. *Our algorithm solves the simplex b-edge-cover problem (Problem 23) in polynomial time in $|T|$ and $B = \sum_{t \in T} b(t)$. Furthermore, we can solve Problem 21 in polynomial time in $|V|$ and $R = \sum_{t \in T} r(t)$.*

Proof. We show the optimality of the output of our algorithm. Without the set of hyperedges $\mathcal{E}_0$ added in Step 1-3 in our algorithm, Step 2 would find a minimum cost multihypergraph $H' = (T, \mathcal{F})$ such that $\mathcal{F}$ is a multiset of $\mathcal{E}$ and $d_{H'}(t) = B + 2$ for any $t \in T$. By using edges in $\mathcal{E}_0$, we can cover k vertices in $t^{(b(t)+1)}, t^{(b(t)+2)}, \ldots, t^{(B+2)}$, where k can be $0, 2, 3, 4, \ldots, B + 2 - b(t)$ (note that we cannot cover exactly one vertex with a zero cost hyperedge). Therefore, in Step 2 of our algorithm, we obtain a minimum cost multihypergraph $H' = (T, \mathcal{F})$ such that $\mathcal{F}$ is a multiset of $\mathcal{E}$ and $d_{H'}(t) \in \{b(t), b(t) + 1, \ldots, B\}$ for any $t \in T$, which is an optimal solution of Problem 23 by the above argument and Claim 26. We note that since we introduced $B+2$ vertices for each vertex $u \in T$ in Step 1-1, the running time of our algorithm is polynomial in $|T|$ and $B = \sum_{t \in T} b(t)$.

Finally, by Lemma 25, the above algorithm solves Problem 21 in polynomial time in $|V|$ and $R = \sum_{t \in T} r(t)$. □

Note that in the SODA version [6] of this paper we have shown the following strengthening of Claim 26: if an instance of the simplex b-edge-cover problem is obtained from an instance of Problem 21 using Lemma 25, then it has an optimal solution $H' = (T, \mathcal{F})$ such that $d_{H'}(t) \leq \max\{r(t) : t \in T\}$. This observation reduces the running time of the algorithm when used for solving Problem 21: just use $\max\{r(t) : t \in T\}$ in the algorithm instead of $R = \sum_{t \in T} r(t)$.

4.3 Strongly polynomial time algorithm

In this subsection, we improve the running time of Theorem 27 to strongly polynomial time.

Let $H = (T, \mathcal{E})$ be a simple 2-3 hypergraph and let $\mathcal{E}_2$ and $\mathcal{E}_3$ be the sets of hyperedges in $\mathcal{E}$ of sizes 2 and 3, respectively. A multihypergraph $H' = (T, \mathcal{F})$, where $\mathcal{F}$ is a multiset of $\mathcal{E}$, is represented by a pair (x, y) with $x \in \mathbb{Z}_+^{\mathcal{E}_2}$ and $y \in \mathbb{Z}_+^{\mathcal{E}_3}$, where $x(e)$ is the multiplicity of $e \in \mathcal{E}_2$ contained in $\mathcal{F}$ and $y(e)$ is the multiplicity of $e \in \mathcal{E}_3$ contained in $\mathcal{F}$. The cost of (x, y) is denoted by

$$\gamma(x, y) := \sum_{e \in \mathcal{E}_2} x(e)\gamma(e) + \sum_{e \in \mathcal{E}_3} y(e)\gamma(e).$$

For $t \in T$, define $d_x(t) := \sum\{x(e) \mid e \in \mathcal{E}_2,\ t \in e\}$ and $d_y(t) := \sum\{y(e) \mid e \in \mathcal{E}_3,\ t \in e\}$.

We first show that there exists an optimal solution of the simplex b-edge-cover problem that contains not so many hyperedges of size 3.

Lemma 28. *There exists an optimal solution (x^*, y^*) of the simplex b-edge-cover problem (Problem 23) such that $d_{y^*}(t) \leq 1$ for any $t \in T$.*

Proof. Let (x^*, y^*) be an optimal solution of the simplex b-edge-cover problem that minimizes $\|y^*\|_1$, i.e., it contains a minimum number of hyperedges of size 3. In what follows, we show that $d_{y^*}(t) \leq 1$ for any $t \in T$.

Assume that $y^*(e) \geq 2$ for some $e = \{u_1, u_2, u_3\}$. By decreasing $y^*(e)$ by two and increasing $x^*(\{u_1, u_2\})$, $x^*(\{u_2, u_3\})$, and $x^*(\{u_3, u_1\})$ by one, we obtain a feasible solution of the problem. Furthermore, since the total cost is not increased by the simplex condition, the obtained solution is also an optimal solution, which contradicts that (x^*, y^*) is an optimal solution minimizing $\|y^*\|_1$.

Assume that $y^*(\{u_1, u_2, u_3\}) \geq 1$ and $y^*(\{u_1, u_2, u_4\}) \geq 1$ for distinct u_1, u_2, u_3 and u_4. Let (x_1, y_1) and (x_2, y_2) be feasible solutions of the problem such that

- (x_1, y_1) is obtained from (x^*, y^*) by replacing $\{u_1, u_2, u_3\}$ and $\{u_1, u_2, u_4\}$ with $\{u_1, u_2\}, \{u_1, u_3\}$, and $\{u_2, u_4\}$, and
- (x_2, y_2) is obtained from (x^*, y^*) by replacing $\{u_1, u_2, u_3\}$ and $\{u_1, u_2, u_4\}$ with $\{u_1, u_2\}, \{u_1, u_4\}$, and $\{u_2, u_3\}$.

Since

$$\begin{aligned} &2\gamma(\{u_1, u_2, u_3\}) + 2\gamma(\{u_1, u_2, u_4\}) \\ &\qquad \geq 2\gamma(\{u_1, u_2\}) + \gamma(\{u_1, u_3\}) + \gamma(\{u_1, u_4\}) + \gamma(\{u_2, u_3\}) + \gamma(\{u_2, u_4\}) \end{aligned}$$

by the simplex condition, we have

$$2\gamma(x^*, y^*) \geq \gamma(x_1, y_1) + \gamma(x_2, y_2),$$

which implies that $\gamma(x^*, y^*) = \gamma(x_1, y_1) = \gamma(x_2, y_2)$ by the optimality of (x^*, y^*). This contradicts that (x^*, y^*) is an optimal solution minimizing $\|y^*\|_1$.

Assume that $y^*(\{u_1, u_2, u_3\}) \geq 1$ and $y^*(\{u_1, u_4, u_5\}) \geq 1$ for distinct u_1, u_2, u_3, u_4 and u_5. Let (x_1, y_1) and (x_2, y_2) be feasible solutions of the problem such that

- (x_1, y_1) is obtained from (x^*, y^*) by replacing $\{u_1, u_2, u_3\}$ and $\{u_1, u_4, u_5\}$ with $\{u_1, u_2\}, \{u_1, u_3\}$, and $\{u_4, u_5\}$, and
- (x_2, y_2) is obtained from (x^*, y^*) by replacing $\{u_1, u_2, u_3\}$ and $\{u_1, u_4, u_5\}$ with $\{u_1, u_4\}, \{u_1, u_5\}$, and $\{u_2, u_3\}$.

Since

$$2\gamma(\{u_1, u_2, u_3\}) + 2\gamma(\{u_1, u_4, u_5\}) \\ \geq \gamma(\{u_1, u_2\}) + \gamma(\{u_1, u_3\}) + \gamma(\{u_4, u_5\}) + \gamma(\{u_1, u_4\}) + \gamma(\{u_1, u_5\}) + \gamma(\{u_2, u_3\})$$

by the simplex condition, we have

$$2\gamma(x^*, y^*) \geq \gamma(x_1, y_1) + \gamma(x_2, y_2),$$

which implies that $\gamma(x^*, y^*) = \gamma(x_1, y_1) = \gamma(x_2, y_2)$ by the optimality of (x^*, y^*). This contradicts that (x^*, y^*) is an optimal solution minimizing $\|y^*\|_1$.

Therefore, any two hyperedges of size 3 do not contain a common vertex, that is, $d_{y^*}(t) \leq 1$ for any $t \in T$. □

Let (x^*, y^*) be an optimal solution of the simplex b-edge-cover problem such that $d_{y^*}(t) \leq 1$ for any $t \in T$ as in Lemma 28. Let $b^* \in \mathbb{Z}^T_+$ be the vector defined by $b^*(t) = \min\{d_{x^*}(t), b(t)\}$ for $t \in T$. Then, we can see that $b \geq b^* \geq b - \mathbf{1}$, where $\mathbf{1} \in \mathbb{Z}^T$ is the all 1's vector, and x^* is a minimum cost b^*-edge-cover (in the graph $(T, \mathcal{E}_2)$) by the optimality of (x^*, y^*). Here, for a graph $G = (V, E)$ and a vector $b \in \mathbb{Z}^V_+$, we say that $z \in \mathbb{Z}^E_+$ is a b-edge-cover if $\sum_{e\colon v \in e} z(e) \geq b(v)$ for any $v \in V$.

Since there exists a strongly polynomial time algorithm that computes a b-edge-cover with minimum total cost (see e.g. [22, Chapter 34]), we would like to utilize it to compute a minimum cost b^*-edge-cover x^*. However, since b^* is not known in advance, we cannot compute x^* directly. Our idea is to use a minimum cost b-edge-cover, which can be computed in strongly polynomial time, instead of the minimum cost b^*-edge-cover x^*. The following lemma guarantees that the minimum cost b-edge-cover is close to x^* to some extent (for a vector $p \in \mathbb{R}^A$ let $\|p\|_\infty = \max_{a \in A} |p(a)|$ and $\|p\|_1 = \sum_{a \in A} |p(a)|$).

Lemma 29 (see [22, Lemma 31.4β])**.** *Let $G = (V, E)$, let $b, b' \in \mathbb{Z}^V_+$ and let $c : E \to \mathbb{R}_+$ be a cost function. Then, for any minimum cost b-edge-cover $z \in \mathbb{Z}^E$, there exists a minimum cost b'-edge-cover $z' \in \mathbb{Z}^E$ satisfying that $\|z - z'\|_\infty \leq 2\|b - b'\|_1$.*

Note that this lemma is stated in terms of a *maximum weight b-matching* in [22, Lemma 31.4β]. Since the minimum cost b-edge-cover problem and the maximum weight b-matching problem are equivalent by considering the complement, we can see that Lemma 29 is equivalent to [22, Lemma 31.4β]. To make the paper self-contained, we give a sketch of the proof of Lemma 29.

Proof Sketch of Lemma 29. It suffices to consider the case when $\|b - b'\|_1 = 1$. By symmetry, we may assume that there exists a vertex $u \in V$ such that $b'(u) = b(u) + 1$ and $b'(v) = b(v)$ for $v \in V \setminus \{u\}$. Let $z \in \mathbb{Z}^E$ be a minimum cost b-edge-cover, let

$z_1 \in \mathbb{Z}^E$ be a minimum cost b'-edge-cover, and suppose that z is not a b'-edge-cover. By a standard alternating path argument, we can find a walk $P = (v_0, e_1, v_1, \ldots, e_l, v_l)$ such that

1. $v_0 = u$, $z_1(e_i) > z(e_i)$ if i is odd, and $z_1(e_i) < z(e_i)$ if i is even,
2. each edge e is traversed at most $\min\{|z_1(e) - z(e)|, 2\}$ times, and
3. $\sum_{e:\, v_l \in e} z_1(e) < \sum_{e:\, v_l \in e} z(e)$ if l is even, and $\sum_{e:\, v_l \in e} z_1(e) > \sum_{e:\, v_l \in e} z(e)$ if l is odd (if $v_l = v_0$ and l is odd, then $\sum_{e:\, v_l \in e} z_1(e) \geq \sum_{e:\, v_l \in e} z(e) + 2$).

Let $z_P \in \mathbb{Z}^E$ be the vector defined by

$$z_P(e) := \begin{cases} |\{i : e_i = e\}| & \text{if } z_1(e) > z(e), \\ -|\{i : e_i = e\}| & \text{if } z_1(e) < z(e), \\ 0 & \text{otherwise.} \end{cases}$$

Then, $z + z_P$ is a b'-edge-cover, $z_1 - z_P$ is a b-edge-cover, and $\|z_P\|_\infty \leq 2$. Since z is a minimum cost b-edge-cover, we have $c \cdot z \leq c \cdot (z_1 - z_P)$, and hence $c \cdot (z + z_P) \leq c \cdot z_1$. This shows that $z' := z + z_P$ is a minimum cost b'-edge-cover satisfying that $\|z' - z\|_\infty = \|z_P\|_\infty \leq 2$. □

We now describe our strongly-polynomial time algorithm for the simplex b-edge-cover problem (Problem 23).

Strongly-polynomial time algorithm for Problem 23

Step 1 Let $x_0 \in \mathbb{Z}_+^{\mathcal{E}_2}$ be a minimum γ-cost b-edge-cover in the graph $(T, \mathcal{E}_2)$, which can be computed in strongly polynomial time.

Step 2 Define $x_1 \in \mathbb{Z}_+^{\mathcal{E}_2}$ by $x_1(e) = \max\{x_0(e) - 2|T|, 0\}$ for $e \in \mathcal{E}_2$.

Step 3 Let $(x_2, y_2) \in \mathbb{Z}_+^{\mathcal{E}_2} \times \mathbb{Z}_+^{\mathcal{E}_3}$ be a minimum cost pair such that $d_{x_2}(t) + d_{y_2}(t) \geq \max\{b(t) - d_{x_1}(t), 0\}$ for any $t \in T$.

Step 4 Output $(x_1 + x_2, y_2)$.

Theorem 30. *Our algorithm solves Problem 23 in strongly polynomial time.*

Proof. Let (x^*, y^*) be an optimal solution of the simplex b-edge-cover problem such that $d_{y^*}(t) \leq 1$ for any $t \in T$ as in Lemma 28. Let $b^* \in \mathbb{Z}_+^T$ be the vector defined by $b^*(t) = \min\{d_{x^*}(t), b(t)\}$ for $t \in T$. As noted earlier, $b \geq b^* \geq b - \mathbf{1}$, where $\mathbf{1} \in \mathbb{Z}^T$ is the all 1's vector, and x^* is a minimum cost b^*-edge-cover (in the graph $(T, \mathcal{E}_2)$) by the optimality of (x^*, y^*). By Lemma 29, there exists a minimum cost b^*-edge-cover x^{**} (which might coincide with x^*) such that

$$\|x_0 - x^{**}\|_\infty \leq 2\|b - b^*\|_1 \leq 2|T|.$$

By the above inequality, it holds that $x^{**} \geq x_1 \geq \mathbf{0}$.

Obviously, $(x_1 + x_2, y_2)$ is a feasible solution of the simplex b-edge-cover problem. Since $(x, y) = (x^{**} - x_1, y^*)$ satisfies that $d_x(t) + d_y(t) \geq \max\{b(t) - d_{x_1}(t), 0\}$ for any $t \in T$, it holds that $\gamma(x_2, y_2) \leq \gamma(x^{**} - x_1, y^*)$ by the choice of (x_2, y_2). Hence, we have $\gamma(x_1 + x_2, y_2) \leq \gamma(x^{**}, y^*) = \gamma(x^*, y^*)$, which means that $(x_1 + x_2, y_2)$ is an optimal solution of the problem.

The only thing left is to show is that Step 3 can be implemented in strongly polynomial time. We use our pseudo-polynomial time algorithm for the problem and note that it runs in polynomial time in $|T|$ and $\sum_{t \in T}(b(t) - d_{x_1}(t))$ by Theorem 27. Since

$$b(t) - d_{x_1}(t) \leq d_{x_0}(t) - d_{x_1}(t) \leq (|T| - 1)\|x_0 - x_1\|_\infty \leq 2|T|^2$$

for any $t \in T$, the running time of this part is indeed polynomial in $|T|$. □

By Lemma 25, we have the following as a corollary.

Corollary 31. *Problem 21 can be solved in strongly polynomial time.*

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 32. *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 $cap(uv) = \sum_{t \in T} r(t)$ for every pair $u, v \in V$. We could not extend our results to Problem 32. 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. Many thanks to Tamás Király for fruitful discussions about the topic and for checking draft versions.

References

[1] Elliot Anshelevich and Adriana Karagiozova, *Terminal backup, 3D matching, and covering cubic graphs*, SIAM J. Comput. **40** (2011), 678–708.

[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] Attila Bernáth and Yusuke Kobayashi, *The generalized terminal backup problem*, Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, 2014, pp. 1678–1686.

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

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

[9] ______, *Connections in combinatorial optimization*, Oxford University Press, 2011.

[10] 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.

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

[12] 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.

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

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

[15] 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.

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

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

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

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

[20] 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.

[21] Gyula Pap, *A TDI description of restricted 2-matching polytopes*, Integer Programming and Combinatorial Optimization, LNCS 3064, 2004, pp. 139–151.

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

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

[24] ———, *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.

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

[26] 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.