On Half-integrality of Network Synthesis Problem

Than Nguyen HAU, Hiroshi HIRAI*, and Nobuyuki TSUCHIMURA†

September, 2012
July, 2013 (version 2)
April, 2014 (version 3)

Abstract

Network synthesis problem is the problem of constructing a minimum cost network satisfying a given flow-requirement. A classical result of Gomory and Hu is that if the cost is uniform and the flow requirement is integer-valued, then there exists a half-integral optimal solution. They also gave a simple algorithm to find a half-integral optimal solution.

In this paper, we show that this half-integrality and the Gomory-Hu algorithm can be extended to a class of fractional cut-covering problems defined by skew-supermodular functions. Application to approximation algorithm is also given.

1 Introduction

Let K_V be a complete undirected graph on node set V. We are given a nonnegative integer-valued flow-requirement $r_{ij} \in \mathbf{Z}_+$ for each unordered pair ij of nodes. A nonnegative edge-capacity $x : E(K_V) \to \mathbf{R}_+$ is said to be *feasible* if, for every node-pair ij, the maximum value of an (i, j)-flow under the capacity x is at least r_{ij}. We are also given a nonnegative edge-cost $a : E(K_V) \to \mathbf{R}_+$. The *network synthesis problem* (NSP) is the problem of finding a feasible edge-capacity of the minimum cost, where the cost of edge-capacity x is defined as $\sum_{e \in E(K_V)} a(e)x(e)$.

A classical result by Gomory and Hu [10] is that NSP admits a half-integral optimal solution provided the edge-cost is uniform.

Theorem 1.1 ([10]). *Suppose $a(e) = 1$ for $e \in E(K_V)$. Then we have the following:*

(1) *The optimal value of* NSP *is equal to* $\displaystyle\frac{1}{2}\sum_{i \in V} \max\{r_{ij} \mid j \in V \setminus \{i\}\}$.

(2) *There exists a half-integral optimal solution in* NSP.

See [5, Chapter 4], [7, Section 7.2.3], and [21, Section 62.3]. Gomory and Hu [10] presented the following simple algorithm to find a half-integral optimal solution, where 1_Y denotes the incidence vector of a set Y:

1. Define an edge-weight r on K_V by $r(ij) := r_{ij}$.

*Department of Mathematical Informatics, Graduate School of Information Science and Technology, The University of Tokyo, Tokyo, 113-8656, Japan. E-mail: hirai@mist.i.u-tokyo.ac.jp

†Department of Informatics, School of Science and Technology, Kwansei Gakuin University, Hyogo, 669-1337, Japan. E-mail: tutimura@kwansei.ac.jp

2. Compute a maximum weight spanning tree T of K_V with respect to r. This tree is called a *dominant requirement tree.*

3. Restrict r to $E(T)$. Decompose r into $r = \sum_{F \in \mathcal{G}} \sigma(F) 1_{E(F)}$ for a family $\mathcal{G}$ of subtrees in T and a positive integral weight σ on $\mathcal{G}$ such that

   ($*$) for $F, F' \in \mathcal{G}$, one of F, F' is a subgraph of the other, or F and F' are vertex-disjoint.

4. For $F \in \mathcal{G}$, take a cycle C_F (in K_V) of vertices $V(F)$.

5. Define $x : E(K_V) \to \mathbf{R}_+$ by

$$x := \sum_{F \in \mathcal{G}: |V(F)|=2} \sigma(F) 1_{E(C_F)} + \frac{1}{2} \sum_{F \in \mathcal{G}: |V(F)|>2} \sigma(F) 1_{E(C_F)}.$$

   Then x is an optimal solution of NSP with unit edge-cost.

The running time of this algorithm is $O(n^2)$; see [16, Chapter 12]. For general edge-costs, this half-integrality fails, and, in subsequent paper [11], Gomory and Hu presented a practically efficient algorithm for NSP by the column generation method applied to an LP-formulation of an exponential size (though NSP has an LP-formulation of a polynomial size; see [21, p. 1054]).

Let us introduce a well-studied class of exponential-size linear problems capturing NSP. Let $f : 2^V \to \mathbf{Z}_+$ be a symmetric nonnegative integer-valued set function on V satisfying $f(\emptyset) = f(V) = 0$, where a set function f is called *symmetric* if it satisfies

$$f(X) = f(V \setminus X) \quad (X \subseteq V). \tag{1.1}$$

For $X \subseteq V$, let δX denote the set of edges in K_V connecting X and $V \setminus X$. Let $\mathrm{Cover}(f)$ denote the set of nonnegative edge-capacities $x : E(K_V) \to \mathbf{R}_+$ satisfying the cut-covering constraint $\sum_{e \in \delta X} x(e) \geq f(X)$ for each $X \subseteq V$. Namely,

$$\mathrm{Cover}(f) := \left\{ x \in \mathbf{R}_+^{E(K_V)} \;\middle|\; \sum_{e \in \delta X} x(e) \geq f(X) \quad (X \subseteq V) \right\}. \tag{1.2}$$

As above, we are given an edge-cost $a : E(K_V) \to \mathbf{R}_+$. Consider the following minimum-cost fractional cut-covering problem:

$$\mathrm{NSP}[f]: \qquad \mathrm{Min.} \sum_{e \in E(K_V)} a(e)x(e) \quad \text{s.t.} \quad x \in \mathrm{Cover}(f).$$

A number of combinatorial optimization problem can be formulated in this way (see the next section). In particular, NSP is a special case of NSP$[f]$. Indeed, for flow-requirement r_{ij}, define R by

$$R(X) := \max\{r_{ij} \mid i \in X \not\ni j\} \quad (\emptyset \neq X \subset V), \tag{1.3}$$

and $R(\emptyset) = R(V) = 0$. By the max-flow min-cut theorem, NSP$[R]$ coincides with NSP.

Our main result is about a half-integrality property of NSP$[f]$ for a special skew-supermodular function f and a special edge-cost a, extending Theorem 1.1. Recall that a symmetric set function f is said to be *skew-supermodular* if it satisfies

$$f(X) + f(Y) \leq \max\{f(X \cap Y) + f(X \cup Y), f(X \setminus Y) + f(Y \setminus X)\} \quad (X, Y \subseteq V). \tag{1.4}$$

The skew-supermodularity has played important roles in optimizations over Cover(f); see the next section. Observe that the inequality (1.4) for a disjoint pair is trivial. We introduce a new property imposed on disjoint pairs. A skew-supermodular function f is said to be *normal* if it satisfies

$$f(X) + f(Y) - f(X \cup Y) \geq 0 \quad (X, Y \subseteq V : X \cap Y = \emptyset), \tag{1.5}$$

and is said to be *evenly-normal* if it satisfies

$$f(X) + f(Y) - f(X \cup Y) \in 2\mathbf{Z}_+ \quad (X, Y \subseteq V : X \cap Y = \emptyset). \tag{1.6}$$

Next we consider special edge-costs. An edge-cost a is called a *tree metric* if a is represented by the distances between a subset of vertices in a weighted tree. It is well-known that a is a tree metric if and only if there exists a pair $(\mathcal{F}, l)$ of a cross-free family $\mathcal{F} \subseteq 2^V$ and a nonnegative weight l on $\mathcal{F}$ such that $a = \sum_{X \in \mathcal{F}} l(X) 1_{\delta X}$; see [3]. Recall that a family $\mathcal{F} \subseteq 2^V$ is said to be *cross-free* if for every $X, Y \in \mathcal{F}$ one of $X \cap Y$, $V \setminus (X \cup Y)$, $X \setminus Y$, and $Y \setminus X$ is emtpy. The main result of this paper is the following.

Theorem 1.2. *Suppose that f is evenly-normal skew-supermodular and a is a tree metric represented as $a = \sum_{X \in \mathcal{F}} l(X) 1_{\delta X}$ for a cross-free family $\mathcal{F}$ and a nonnegative weight $l : \mathcal{F} \to \mathbf{R}_+$. Then we have the following:*

(1) *The optimal value of* NSP$[f]$ *is equal to* $\sum_{X \in \mathcal{F}} l(X) f(X)$.

(2) *There exists an integral optimal solution in* NSP$[f]$.

Furthermore there exists an $O(n\theta + n^2)$ algorithm to find an integral optimal solution in NSP$[f]$, *where $n := |V|$ and θ is the running time of evaluating f.*

This theorem includes the half-integrality for NSP$[f]$ for a normal skew-supermodular function f. One can see this fact from: (1) if f is normal skew-supermodular, then $2f$ is evenly-normal skew-supermodular, and (2) if x is optimal to NSP$[2f]$, then $x/2$ is optimal to NSP$[f]$. Also Theorem 1.2 includes Theorem 1.1. Indeed, it is easy to see that R is normal skew-supermodular (the skew-supermodularity of R is well-known [7, Lemma 8.1.9]). Since the unit cost is represented as $\sum_{i \in V} (1/2) 1_{\delta\{i\}}$, we can take $\{\{i\} \mid i \in V\}$ as $\mathcal{F}$, with $l(\{i\}) := 1/2$ $(i \in V)$. Applying Theorem 1.2 to NSP$[2R]$, we obtain Theorem 1.1. Note that R is evaluated in $O(n)$ time; $R(X)$ is equal to $\max\{r_{ij} \mid ij \in E(T), i \in X \not\ni j\}$ for a dominant requirement tree T. Therefore the running time of our algorithm is $O(n^2)$; our algorithm in fact generalizes the Gomory-Hu algorithm. Also there are many $O(n^2)$ algorithms to determine whether a is a tree metric and to obtain an expression $a = \sum_{X \in \mathcal{F}} l(X) 1_{\delta X}$; *Neighbor-Joining* [20] is a popular method.

The rest of this paper is organized as follows. In the next section (Section 2), we discuss the relevance to previous works on skew-supermodular survivable network design. We also present applications of Theorem 1.2 to approximation algorithms, though our original motivation was to understand the half-integrality property and the Gomory-Hu algorithm of NSP from a set-function property of f. In Section 3, we give a proof of Theorem 1.2.

2 Related work and application

Related work. Integer linear optimization over Cover(f) with capacity bound constraint $l \leq x \leq u$, denoted by SND$[f; l, u]$, is a general form of the *survivable network design problem*, and can formulate various combinatorial optimization problems; see [17, Chapter 20] and references therein. The natural LP-relaxation of SND$[f; l, u]$ is denoted by SND$^*[f; l, u]$. In particular SND$^*[f; 0, +\infty]$ is equal to NSP$[f]$. The integer network synthesis SND$[f; 0, +\infty]$ is denoted by INSP$[f]$.

Let us mention examples as well as relevances to our result. For $T \subseteq V$ with $|T|$ even, define a set function f_T by $f_T(X) := 1$ for $X \subseteq V$ with $|X \cap T|$ odd and $f_T(X) := 0$ for others. Then INSP$[f_T]$ is the minimum-cost T-join problem (with nonnegative costs). The Edmonds-Johnson theorem [12] says that the LP-relaxation NSP$[f_T]$ is exact. Namely the integrality holds for NSP$[f_T]$ with every cost function a. This set function f_T is evenly-normal skew-supermodular. Our theorem asserts the integrality only for tree-metric edge-costs, and that an optimal T-join can be greedily found in this case.

For a positive integer $k > 0$, define a normal skew-supermodular function f_k by $f_k(X) := k$ $(\emptyset \neq X \neq V)$. If k is even, then f_k is evenly-normal. Then INSP$[f_k]$ is the minimum k-edge-connected subgraph problem. In particular, INSP$[f_2]$ with the degree constraint is nothing but the traveling salesman problem. Suppose that a is a metric. Then NSP$[f_2]$ is equivalent to the subtour elimination LP-relaxation of TSP; see [25, 23.12]. Suppose further that a is a tree metric. TSP on a tree is quite easy. An optimal tour is a tour which traces each edge in the tree (at most) twice. This tour in fact coincides with our integral optimal solution in Theorem 1.2.

Consider the case $f = R$ for connectivity requirement $\{r_{ij}\}$ (see (1.3)). Then SND$[R; l, +\infty]$ is the *connectivity augmentation problem.* Frank [6] gave a polynomial time algorithm to SND$[R; l, +\infty]$ for node-induced edge-costs. An edge-cost a is called *node-induced* if there is $b : V \to \mathbf{R}_+$with

$$a(ij) = b(i) + b(j) \quad (i, j \in V).$$

As a corollary, he proved the half-integrality of SND$^*[R; l, +\infty]$ for node-induced edge-costs. Actually Frank's argument works for a *proper function* [17, Definition 20.17], which is a symmetric set function f satisfying

$$\text{(2.1)} \qquad \max\{f(X), f(Y)\} \geq f(X \cup Y) \quad (X, Y \subseteq V : X \cap Y = \emptyset).$$

See [2] for details. Notice that R is proper. The condition (2.1) is stronger than the normality condition (1.5), and is stronger than the skew-supermodularity (1.4); see [17, Proposition 20.18]. Observe that a node-induced cost function is a tree metric corresponding to a star. So our result extends Frank's half-integrality result in the case of $l = 0$. Note that Frank's argument is based on the edge-splitting technique, and does not explain the simplicity of the Gomory-Hu algorithm. Note also that our theorem is not applicable to SND$^*[R; l, +\infty]$ (since the negative of cut function $(X \mapsto \sum_{e \in \delta X} l(e))$ is not normal in general).

In the study of hypergraph connectivity augmentation, Szigeti [23] showed that for an arbitrary skew-supermodular function f there is a half-integral optimal solution in NSP$[f]$ with uniform-cost. His proof is also based on the edge-splitting. We do not know how to find this half-integral solution in polynomial time, since the edge-splitting approach needs to check whether a given $x \in \mathbf{R}^{E(K_V)}$ belongs to Cover(f); see the argument below.

Approximation algorithm of SND$[f; l, u]$ for proper/skew-supermodular functions f has also been extensively studied; see [25, Chapters 22, 23] and [17, Section 20.3]. The integer network synthesis INSP$[R]$ is NP-hard for general edge-cost. The skew-supermodular INSP$[f]$ is NP-hard even if the edge-cost is uniform, since it includes an NP-hard subclass of the NA-connectivity augmentation problem [18]; see [14, Lemma

1.1]. There are two major approximation algorithms for SND$[f; l, u]$: Jain's 2-approximation algorithm [15] and the primal-dual $2H(f_{\max})$-approximation algorithm [9], where $f_{\max} := \max_{X \subseteq V} f(X)$ and $H(k) := 1 + 1/2 + \cdots + 1/k$. The half-integrality of SND*$[f; l, u]$ would yield a 2-approximation algorithm for SND$[f; l, u]$. However SND*$[f; l, u]$ does not have the half-integrality in general; see [25, Lemma 23.2] and [17, p. 544–545]. In [15], Jain discovered a weaker property that every basic solution x of SND*$[f; l, u]$ has an edge e with $x(e) \geq 1/2$. Based on this property, he devised a 2-approximation algorithm for SND$[f; l, u]$, provided a separation oracle of Cover(f) or a polynomial time algorithm solving LP-relaxation SND*$[f; l, u]$ is available. The primal-dual approximation algorithm also needs a feasibility-checking oracle of Cover(f), an oracle of checking whether a given x belongs to Cover(f). Another notable result is a 7/4-approximation algorithm by Nutov [19] for SND$[f; l, +\infty]$ with uniform edge-cost. His algorithm also needs a feasibility-checking oracle of Cover(f). For a proper function f (given by an oracle), there is an efficient separation algorithm for Cover(f) [17, Theorem 20.20], and SND*$[f; l, u]$ can be solved in polynomial time by the ellipsoid method. In addition, if $f = R$, then the feasibility-check of Cover(R) can be done by any max-flow min-cut algorithm, and SND*$[R; l, u]$ has a polynomial-size LP formulation, which can be solved in polynomial time by the interior point method.

For a general skew-supermodular function f (given by an oracle), however, no efficient feasibility-checking/separation algorithm for Cover(f) is known; see [17, p. 534]. This problem is reduced to the problem of maximizing a skew-supermodular function, which is also not known to be (oracle-)tractable; see EGRES Open [13]. Even if the normality condition (1.5) is imposed, we still do not know whether Cover(f) has an efficient separation algorithm, and we do not know whether NSP$[f]$ is solvable in polynomial time. From this point of view, our result might be interesting since it gives a new class of oracle-tractable NSP$[f]$.

Application to approximation algorithm. As is well-known, the half-integrality leads to a 2-approximation algorithm; see [25]. For a half-integral optimal solution x of NSP$[f]$, by rounding up $x(e)$ to $\lceil x(e) \rceil$, we obtain a feasible solution $\lceil x \rceil$ of INSP$[f]$, which is a 2-approximate solution of INSP$[f]$.

Theorem 2.1. *Suppose that f is a normal skew-supermodular function given by an evaluation oracle. There is a 2-approximation algorithm for* INSP$[f]$ *with tree-metric costs.*

An interesting point is that this algorithm does not require any feasibility-checking oracle of Cover(f). Furthermore, by combining Theorem 1.2 with a standard argument of Bartal's probabilistic embedding [1] (see [24, Section 8.5, 8.6]), we obtain a randomized $O(\log n)$-approximation algorithm for INSP$[f]$ with general cost as follows. We can assume that edge-cost a is a metric, i.e., it satisfies the triangle inequalities $a(ij)+a(jk) \geq a(ik)$ $(i, j, k \in V)$ (see the proof of [25, Theorem 3.2]), and there is no edge e with $a(e) = 0$ (otherwise, contract all edges e with $a(e) = 0$). It is shown by [4] that there exists a randomized $O(n^2)$ algorithm to find a tree metric τ with $a(e) \leq \tau(e)$ and $\mathrm{E}[\tau(e)] \leq O(\log n)a(e)$ $(e \in E(K_V))$, where $\mathrm{E}[X]$ is the expected value of a random variable X. More precisely, there is an $O(n^2)$ algorithm to sample a tree metric from the space $\mathcal{T}$ of tree metrics τ dominating a with respect to a probability measure μ on $\mathcal{T}$ satisfying $\mathrm{E}[\tau(e)] = \int_{\tau \in \mathcal{T}} \tau(e) \mathrm{d}\mu \leq O(\log n)a(e)$ $(e \in E(K_V))$. Let x^τ be a half-integral optimal solution x of NSP$[f]$ for tree-metric cost τ (obtained by the algorithm in Theorem 1.2). The rounding solution $\lceil x^\tau \rceil$ is a 2-approximate solution of INSP$[f]$ with cost τ (by Theorem 2.1), and has the expected objective value at most $O(\log n)$

times the optimal value of INSP$[f]$ with cost a, since

$$\begin{aligned}
&\mathrm{E}\left[\sum_e a(e)\lceil x^\tau(e)\rceil\right] = \int_{\tau\in\mathcal{T}} \sum_e a(e)\lceil x^\tau(e)\rceil \mathrm{d}\mu \le \int_{\tau\in\mathcal{T}} \sum_e \tau(e)\lceil x^\tau(e)\rceil \mathrm{d}\mu \\
&\le \int_{\tau\in\mathcal{T}} 2\sum_e \tau(e) y^\tau(e)\mathrm{d}\mu \le \int_{\tau\in\mathcal{T}} 2\sum_e \tau(e)y(e)\mathrm{d}\mu = 2\sum_e y(e)\int_{\tau\in\mathcal{T}} \tau(e)\mathrm{d}\mu \\
&\le O(\log n)\sum_e a(e)y(e),
\end{aligned}$$

where y^τ and y denote optimal solutions of INSP$[f]$ with cost τ and of INSP$[f]$ with cost a, respectively. The same argument implies that $\mathrm{E}[\sum a(e)x^\tau(e)]$ is at most $O(\log n)$ times the optimal value of NSP$[f]$ with cost a.

Theorem 2.2. *Suppose that f is a normal skew-supermodular function given by an evaluation oracle. There exists a randomized $O(\log n)$-approximation algorithm for* NSP$[f]$ *and for* INSP$[f]$.

Our algorithm for INSP$[f]$ is comparable to the primal-dual $2H(f_{\max})$-approximation algorithm in the case where $f_{\max}$ is a polynomial of the number n of nodes, and is of course much inferior than Jain's algorithm in approximation factor. Also our algorithm is not extendable to SND$[f; l, u]$. However our algorithm works only with an evaluation oracle of f, and is considerably fast. For the special case of $f = R$, Jain's algorithm needs to solve the LP-relaxation SND* in each step. This is quite costly, and almost impossible for a large instance; the running time of Jain's algorithm is beyond $O(n^6)$, as estimated in [15, Section 8]. Note also that the running time of the primal-dual approximation algorithm is beyond $O((f_{\max})^2 n^2)$; see [17, p. 539]. On the other hand, the running time of our algorithm is $O(n^2)$ par one trial. So our algorithm may also be useful to obtain a good initial feasible solution for local search heuristics, e.g., [22]. An experimental study will be given in a future work.

3 Proof

We need two lemmas. The first lemma is a general property of a symmetric skew-supermodular function. We denote $\sum_{e\in F} x(e)$ by $x(F)$ for $F \subseteq E(K_V)$.

Lemma 3.1. *Let $f : 2^V \to \mathbf{Z}_+$ be a symmetric skew-supermodular function and $\mathcal{F}$ a cross-free family on V. If $x : E(K_V) \to \mathbf{R}_+$ satisfies $x(\delta X) = f(X)$ for all $X \in \mathcal{F}$, then one of the following holds:*

(1) *x satisfies $x(\delta X) \ge f(X)$ for all $X \subseteq V$.*

(2) *There exists $W \subseteq V$ such that $x(\delta W) < f(W)$ and $\mathcal{F} \cup \{W\}$ is cross-free.*

In particular, if $\mathcal{F}$ is a maximal cross-free family, then (1) *holds.*

Proof. By symmetry, we may assume $Y \in \mathcal{F} \Leftrightarrow V \setminus Y \in \mathcal{F}$. Suppose that (1) does not hold. Then there is $Z \subseteq V$ with $x(\delta Z) < f(Z)$. Take such a $Z \subseteq V$ such that the crossing number $N_Z := |\{X \in \mathcal{F} \mid Z \text{ and } X \text{ are crossing}\}|$ is minimum, where X and Y are said to be *crossing* if all $X \cap Y$, $V \setminus (X \cup Y)$, $X \setminus Y$, and $Y \setminus X$ are nonempty. If $N_Z = 0$, we are done. Suppose not. Take $Y \in \mathcal{F}$ such that Z and Y are crossing. By the skew-supermodularity of f, we have

$$f(Y) + f(Z) \le f(Y \cap Z) + f(Y \cup Z) \text{ or } f(Y) + f(Z) \le f(Y \setminus Z) + f(Z \setminus Y).$$

By symmetry, we may assume the first case; otherwise replace Y by $V \setminus Y$. By $x(\delta Y) = f(Y)$ and $x(\delta Z) < f(Z)$, we have

$$x(\delta Y) + x(\delta Z) < f(Y) + f(Z) \leq f(Y \cap Z) + f(Y \cup Z).$$

By $x \geq 0$, we have $x(\delta(Y \cap Z)) + x(\delta(Y \cup Z)) \leq x(\delta Y) + x(\delta Z)$. Thus $x(\delta(Y \cap Z)) < f(Y \cap Z)$ or $x(\delta(Y \cup Z)) < f(Y \cup Z)$. Again, by symmetry, we may assume $x(\delta(Y \cap Z)) < f(Y \cap Z)$; otherwise replace Y by $V \setminus Y$ and replace Z by $V \setminus Z$.

Then $N_{Y \cap Z} < N_Z$ (see [25, Lemma 23.15]), and this contradicts the minimality assumption. □

The second lemma is about the path decomposition of a capacitated trivalent tree. A tree is said to be *trivalent* if each node that is not a leaf has degree three, where a *leaf* of a tree is a node of degree one.

Lemma 3.2. *Let T be a trivalent tree, and $c : E(T) \to \mathbf{Z}_+$ an integer-valued edge-capacity. If $c(e) + c(e') - c(e'') \in 2\mathbf{Z}_+$ holds for every pairwise-incident triple (e, e', e'') of edges, then there exists a pair $(\mathcal{P}, \lambda)$ of a set $\mathcal{P}$ of simple paths connecting leaves and an integral weight $\lambda : \mathcal{P} \to \mathbf{Z}_+$ such that $\sum_{P \in \mathcal{P}} \lambda(P) 1_{E(P)} = c$.*

Proof. For every incident pair e, e' of edges, define $l(e, e')$ by

$$l(e, e') := (c(e) + c(e') - c(e''))/2,$$

where e'' is the third edge incident to e and to e'. Then $l(e, e')$ is a nonnegative integer, and $c(e) = l(e, e') + l(e, e'')$. $(\mathcal{P}, \lambda)$ is constructed as follows.

Let $\mathcal{P} := \emptyset$ initially. Take edge $e = uv$ with $c(e) > 0$. Suppose that u is not a leaf. Then there is an edge e' incident to u with $l(e, e') > 0$. Necessarily $c(e') > 0$ (otherwise $c(e') = 0$ and $l(e, e') = 0$). Hence we can extend e to a simple path $P = (e_0, e_1, \ldots, e_k)$ connecting leaves. Add P to $\mathcal{P}$. Define $\lambda(P) := \min_{i=1,\ldots,k} l(e_{i-1}, e_i)$ (> 0). Let $\tilde{c} := c - \lambda(P) 1_{E(P)}$. Then $\tilde{c}$ satisfies the condition of this lemma. To see this, take an arbitrary pairwise-incident triple (e, e', e'') of edges. We show $\tilde{c}(e) + \tilde{c}(e') - \tilde{c}(e'') \in 2\mathbf{Z}_+$. Here $E(P) \cap \{e, e', e''\}$ is $\emptyset$, $\{e', e''\}$, $\{e, e''\}$, or $\{e, e'\}$. For the first three cases, we have $\tilde{c}(e) + \tilde{c}(e') - \tilde{c}(e'') = c(e) + c(e') - c(e'') \in 2\mathbf{Z}_+$. For the last case, we have $\tilde{c}(e) + \tilde{c}(e') - \tilde{c}(e'') = c(e) + c(e') - c(e'') - 2\lambda(P)$, which must be a nonnegative even integer by definition of $\lambda(P)$.

Let $c \leftarrow \tilde{c}$, and repeat this process. In each step, at least one of $l(e, e')$ is zero. After $O(|V(T)|)$ step, we have $c = 0$ and obtain a desired $(\mathcal{P}, \lambda)$. □

Proof of Theorem 1.2. Consider the LP-dual of NSP[f], which is given by

$$\begin{array}{lll} \text{DualNSP}[f]: & \text{Max.} & \displaystyle\sum_{X \subseteq V} \pi(X) f(X) \\ & \text{s.t.} & \displaystyle\sum_{X \subseteq V} \pi(X) 1_{\delta X} \leq a \\ & & \pi : 2^V \to \mathbf{R}_+. \end{array}$$

Suppose that a is represented by $a = \sum_{X \in \mathcal{F}} l(X) 1_{\delta X}$ for some cross-free family $\mathcal{F}$ and some nonnegative weight l on $\mathcal{F}$. Define $\pi : 2^V \to \mathbf{R}_+$ by

$$\pi(X) = \begin{cases} l(X) & \text{if } X \in \mathcal{F}, \\ 0 & \text{otherwise}, \end{cases} \qquad (X \subseteq V).$$

Then π is feasible to DualNSP[f] with the objective value $\sum_{X\in\mathcal{F}} l(X)f(X)$. We are going to construct a feasible integral solution x in NSP[f] satisfying

(3.1) $$x(\delta X) = f(X) \quad (X \in \mathcal{F}).$$

If this is possible, then, by the complementary slackness, x is optimal to NSP[f] and π is optimal to DualNSP[f]; hence Theorem 1.2 is proved.

Take a maximal cross-free family $\mathcal{F}^*$ including $\mathcal{F}$. Here recall the tree-representation of a cross-free family; see [7, Section 1.4] and [21, Section 13.4]. By the maximality of $\mathcal{F}^*$, there exists a trivalent tree T on vertex set $V \cup I$ with the following properties:

(3.2) (1) V is the set of leaves of T, and I is the set of non-leaf nodes.

(2) $\mathcal{F}^* \setminus \{\emptyset, V\} = \bigcup_{e\in E(T)}\{A_e, B_e\}$, where $\{A_e, B_e\}$ denotes the bipartition of V such that A_e (or B_e) is the set of leaves of one of components of $T - e$.

Define edge-weight $c : E(T) \to \mathbf{Z}_+$ by

(3.3) $$c(e) := f(A_e)(= f(B_e)) \quad (e \in E(T)).$$

By symmetry (1.1) and the evenly-normal property (1.6) of f, for each pairwise-incident triple (e, e', e'') of edges in T, we have

$$c(e) + c(e') - c(e'') = f(A_e) + f(A_{e'}) - f(A_{e''}) \in 2\mathbf{Z}_+,$$

where we can assume $A_e \cap A_{e'} = \emptyset$ and $A_{e''} = A_e \cup A_{e'}$. By Lemma 3.2, there exists a pair $(\mathcal{P}, \lambda)$ of a set $\mathcal{P}$ of simple paths connecting V and a positive integral weight λ on $\mathcal{P}$ with $\sum_{P\in\mathcal{P}} \lambda(P)1_{E(P)} = c$. Define $x : E(K_V) \to \mathbf{Z}_+$ by

(3.4) $$x(ij) := \begin{cases} \lambda(P) & \text{if } \exists P \in \mathcal{P} : P \text{ connects } i \text{ and } j, \\ 0 & \text{otherwise,} \end{cases} \qquad (ij \in E(K_V)).$$

Since each P is simple, we have

$$x(\delta A_e) = c(e) = f(A_e) \quad (e \in E(T)).$$

By (3.2) (2), this implies

$$x(\delta X) = f(X) \quad (X \in \mathcal{F}^*).$$

By Lemma 3.1, x is feasible to NSP[f]. By $\mathcal{F} \subseteq \mathcal{F}^*$, x satisfies (3.1). Therefore, x is an integral optimal solution in NSP[f], π is an optimal solution in DualNSP[f], and the optimal value is equal to $\sum_{X\in\mathcal{F}} l(X)1_{\delta X}$. □

Algorithm to find an integral optimal solution in Theorem 1.2. Our proof gives the following $O(n\theta + n^2)$ algorithm to find an integral optimal solution, where $n := |V|$, and θ denotes the running time of an oracle of f.

step 1: Take a maximal cross-free family $\mathcal{F}^*$ including $\mathcal{F}$.

step 2: Construct a trivalent tree T with (3.2).

step 3: Define edge-weight c by (3.3).

step 4: Decompose c as $c = \sum_{P\in\mathcal{P}} \lambda(P)1_{E(P)}$ according to the proof of Lemma 3.2.

step 5: Define x by (3.4), and then x is an integral optimal solution in NSP[f].

Steps 1,2 can be done in $O(n)$ time, step 3 can be done by $O(n)$ calls of f, and steps 4,5 can be done in $O(n^2)$ time.

Gomory-Hu algorithm reconsidered. The Gomory-Hu algorithm can be viewed as a special case of our algorithm. First note that, in the case of unit cost, we can take an arbitrary maximal cross-free family in step 1. Consider a dominant requirement tree T with respect to r. For $e \in E(T)$, let $\{A_e, B_e\}$ denote the bipartition of V determined by $T - e$. Then $\mathcal{F} := \bigcup_{e \in E(T)} \{A_e, B_e\}$ is cross-free. Extend $\mathcal{F}$ to a maximal cross-free family $\mathcal{F}^*$. Take a trivalent tree $\bar{T}$ corresponding to $\mathcal{F}^*$. Define $c : E(\bar{T}) \to \mathbf{Z}_+$ by (3.3) with $f := R$. Recall that R is proper, i.e., it satisfies (2.1). By symmetry, the maximum of $R(A)$, $R(B)$, and $R(A \cup B)$ is attained at least twice. This implies the following property of c:

(3.5) For each pairwise-incident triple (e, e', e'') of edges, the maximum of $c(e)$, $c(e')$, and $c(e'')$ is attained at least twice.

Decompose c as $c = \sum_{F \in \bar{\mathcal{G}}} \sigma(F) 1_{E(F)}$ for a family of subtrees $\bar{\mathcal{G}}$ and a positive integral weight σ on $\bar{\mathcal{G}}$ with the property $(*)$ in the step 3 of the Gomory-Hu algorithm. By (3.5), the set of leaves of each subtree $F \in \bar{\mathcal{G}}$ belongs to V. Therefore we may apply the path decomposition in Lemma 3.2 to each $\sigma(F)1_{E(F)}$ independently. From the path decomposition of $\sigma(F)1_{E(F)}$, define x_F by $x_F := (\sigma(F)/2)1_{E(C_F)}$ if $|V(F)| \geq 3$ and $x_F := \sigma(F)1_{E(C_F)}$ if $|V(F)| = 2$, where a cycle C_F of vertices $V(F)$ in K_V. Then $x := \sum_{F \in \bar{\mathcal{G}}} x_F$ is optimal.

By construction, T can be regarded as a tree obtained by contracting some of edges of $\bar{T}$. So we can regard $E(T)$ as $E(T) \subseteq E(\bar{T})$. Since T is a maximum spanning tree, we have

$$r(e) = R(A_e)(= R(B_e)) \quad (e \in E(T)).$$

This means that r coincides with the restriction of c to $E(T)$. Also one can see from the definition of R that the family obtained from $\bar{\mathcal{G}}$ by contracting the edges coincides with the family $\mathcal{G}$ in the Gomory-Hu algorithm (see Introduction). Therefore, the above-mentioned process coincides with the Gomory-Hu algorithm.

Remark 3.3. Lemma 3.1 is viewed as a symmetric analogue of the following well-property of submodular functions: If f is a submodular function on V and $x : V \to \mathbf{R}$ satisfies $x(Y) = f(Y)$ $(Y \in \mathcal{F})$ for some maximal chain $\mathcal{F}$ in 2^V, then $x(X) \leq f(X)$ for all $X \subseteq V$. See [7, 8, 21]. This property guarantees the correctness of the greedy algorithm for the base polytope. Also in our algorithm, Lemma 3.1 is used for a similar purpose. So our algorithm may be a symmetric analogue of the greedy algorithm.

Acknowledgments

We thank referees for helpful comments; the proof of Lemma 3.1 was simplified by a suggestion of a referee. We learned some of bibliographical information from the lectures given by Toshimasa Ishii at July 2013. The second author is partially supported by a Grant-in-Aid for Scientific Research from the Ministry of Education, Culture, Sports, Science and Technology of Japan, and is partially supported by Aihara Project, the FIRST program from JSPS.

References

[1] Y. Bartal, Probabilistic approximations of metric spaces and its algorithmic applications, in: *Proceedings of the 37th Annual Symposium on Foundations of Computer Science (FOCS 96)*, 184–193, IEEE Computer Society Press, Los Alamitos, California, 1996.

[2] A. Bernáth, T. Király, and L. Végh, Special skew-supermodular functions and a generalization of Mader's splitting-off theorem, EGRES Technical Report no. 2011-10, (2011).

[3] A. Dress, K.T. Huber, J. Koolen, V. Moulton, and A. Spillner, *Basic Phylogenetic Combinatorics*, Cambridge University Press, Cambridge, 2012.

[4] J. Fakcharoenphol, S. Rao, and K. Talwar, A tight bound on approximating arbitrary metrics by tree metrics, *Journal of Computer and System Sciences* **69** (2004) 485–497.

[5] L. R. Ford, Jr. and D. R. Fulkerson, *Flows in Networks*, Princeton University Press, Princeton, 1962.

[6] A. Frank, Augmenting graphs to meet edge-connectivity requirements. *SIAM Journal on Discrete Mathematics* **5** (1992), 25–53.

[7] A. Frank, *Connections in Combinatorial Optimization*, Oxford University Press, Oxford, 2011.

[8] S. Fujishige, *Submodular Functions and Optimization, 2nd Edition*, Elsevier, Amsterdam, 2005.

[9] M.X. Goemans, D.B. Shmoys, A.V. Goldberg, É. Tardos, S. Plotkin, and D. P. Williamson, Improved approximation algorithms for network design problems, In *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '94)*, ACM, New York, 1994.

[10] R. E. Gomory and T. C. Hu, Multi-terminal network flows, *Journal of the Society for Industrial and Applied Mathematics* **9** (1961), 551–570.

[11] R. E. Gomory and T. C. Hu, An application of generalized linear programming to network flows, *Journal of the Society for Industrial and Applied Mathematics* **10** (1962) 260–283.

[12] J. Edmonds and E. L. Johnson, Matching, Euler tours and the Chinese postman, *Mathematical Programming* **5** (1973), 88–124.

[13] EGRES Open, `http://lemon.cs.elte.hu/egres/open/Main_Page`.

[14] T. Ishii, Y. Akiyama, and H. Nagamochi, Minimum augmentation of edge-connectivity between vertices and sets of vertices in undirected graphs, *Algorithmica* **56** (2010), 413–436.

[15] K. Jain, A factor 2 approximation algorithm for the generalized Steiner network problem, *Combinatorica* **21** (2001), 39–60.

[16] D. Jungnickel, *Graphs, Networks and Algorithms, Fourth Edition*, Springer-Verlag, Berlin, 2013.

[17] B. Korte and J. Vygen, *Combinatorial Optimization, Fifth Edition,* Springer-Verlag, Berlin, 2012.

[18] H. Miwa and H. Ito, NA-edge-connectivity augmentation problems by adding edges, *Journal of the Operations Research Society of Japan* **47** (2004), 224–243.

[19] Z. Nutov, Approximating connectivity augmentation problems, *ACM Transactions on Algorithms* **6**, (2009), Article No. 5.

[20] N. Saitou and M. Nei, The neighbor-joining method: A new method for reconstructing phylogenetic trees, *Molecular Biology and Evolution* **4** (1987), 406–425.

[21] A. Schrijver, *Combinatorial Optimization*, Springer-Verlag, Berlin, 2003.

[22] K. Steiglitz, P. Weiner, and D. J. Kleitman, The design of minimum-cost survivable networks, *IEEE Transactions on Circuit Theory*, CT-16, 1969, 455–460.

[23] Z. Szigeti, Hypergraph connectivity augmentation. *Mathematical Programing, Series B* **84** (1999), 519–527.

[24] D. P. Williamson and D. B. Shmoys, *The Design of Approximation Algorithms*, Cambridge University Press, Cambridge, 2011.

[25] V. V. Vazirani, *Approximation Algorithms*, Springer-Verlag, Berlin, 2001.