RIMS Kôkyûroku Bessatsu
B23 (2010), 49–62

Survivable Network Design Problems with Weighted Degree Constraints

By

Takuro FUKUNAGA* and Hiroshi NAGAMOCHI**

Abstract

In this paper, we discuss a network design problem with degree constraints, which has been extensively studied so far. A successful approach for this problem is the iterative rounding. In this paper, we see that the iterative rounding can be applied to more general problem obtained by replacing degree constraints with weighted degree constraints. We also briefly review several previous works to the network design problem with degree constraints.

§ 1. Introduction

Let $G = (V, E)$ be an undirected graph with an edge cost $c : E \to \mathbb{Q}_+$, where $\mathbb{Q}_+$ is the set of non-negative rational numbers. It is a fundamental problem to find a minimum cost subset F of E that satisfies some connectivity requirement. In fact, there are many studies on this topic from the view point of algorithms so far [11, 13]. In this paper, we consider problems that demand a solution F to satisfy both connectivity and degree constraints.

For stating our problems formally, let us define several notations related to connectivity of graphs. For a subset U of V and a subset F of E, $\delta(U; F)$ denotes the set of edges in F which join vertices in U with those in $V - U$, and $F(U)$ denotes the set of edges in F whose both end vertices are in U. We sometimes represent a singleton $\{v\}$ by v. The degree $|\delta(v, F)|$ of a vertex v in the graph (V, F) is denoted by $d(v; F)$. Let

Received October 17, 2008. Revised July 3, 2009.
2000 Mathematics Subject Classification(s): 05C85, 90C27
Key Words: degree constraint, edge connectivity, iterative rounding, network design
This work was partially supported by Grant-in-Aid for Scientific Research from the Ministry of Education, Culture, Sports, Science and Technology of Japan.
*Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan.
E-mail: takuro@amp.i.kyoto-u.ac.jp
**Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan.
E-mail: nag@amp.i.kyoto-u.ac.jp

$\mathbb{N}$ be the set of natural numbers. For a given set function $f : 2^V \to \mathbb{N}$ on V, a graph $G' = (V, F)$ is called *f-connected* when $|\delta(U; F)| \geq f(U)$ holds for every non-empty $U \subset V$. If $f(X)+f(Y) \leq f(X \cap Y)+f(X \cup Y)$ or $f(X)+f(Y) \leq f(X-Y)+f(Y-X)$ holds for any $X, Y \subseteq V$, then f is called *skew supermodular*. With a skew supermodular set function, f-connectivity represents a wide variety of connectivity of graphs. Let $\lambda(u, v; G)$ denote the edge-connectivity between vertices u and v in a graph G, and define $f(U)$ as $\max\{r(u, v) \mid u \in U, v \in V - U\}$ for non-empty $U \subset V$ from some $r : V \times V \to \mathbb{N}$. Then $\lambda(u, v; G) \geq r(u, v)$ holds for each $u, v \in V$ if and only if G is f-connected.

Now we formulate a problem.

Degree Bounded Survivable Network Problem (DBoundedNetwork): An input consists of an undirected graph $G = (V, E)$, an edge-cost $c : E \to \mathbb{Q}_+$, a skew supermodular set function $f : 2^V \to \mathbb{N}$, and a degree-bound $b : V \to \mathbb{Q}_+$. A solution $F \subseteq E$ is feasible if $G' = (V, F)$ is f-connected and degree constraint $d(v; F) \leq b(v)$ is satisfied for each $v \in V$. The objective is to find a feasible solution that minimizes its cost $\sum_{e \in F} c(e)$.

If $f(X) = 1$ for all non-empty $X \subset V$, then minimal feasible solutions of the problem are trees. We call instances with such f *degree bounded spanning tree problem* (DBoundedTree).

Feasible solutions of DBoundedTree are Hamiltonian paths when $b(v) = 2$ for all $v \in V$. This means that it is NP-hard to test whether an instance of DBoundedTree (and hence DBoundedNetwork) is feasible or not. One way to avoid this difficulty is to restrict instances. Fukunaga and Nagamochi [7] considered the metric case of problem DBoundedNetwork. That is to say, they assume that the graph G is complete, c satisfies triangle inequalities, and each edge can be chosen more than once as multiple edges. Another way is to consider bi-criteria approximation algorithms by relaxing the degree constraints. Lau et al. [18] proposed an algorithm that outputs a network of cost at most twice the optimal while the degree of $v \in V$ is at most $2b(v) + 3$. This result was improved in Lau and Singh [19]. These results are based on the iterative rounding of an LP relaxation.

This paper has two aims. One is to briefly review previous approaches for problem DBoundedNetwork. The other is to show that the approach of Lau and Singh [19] can be applied to more general setting.

Define a weight function $w : E \times V \to \mathbb{Q}_+$ on pairs of edges and their end vertices. We define the *weighted degree* of a vertex $v \in V$ in G as $\sum_{e \in \delta(v;E)} w(e, v)$, and denote it by $d_w(v; E)$. The weighted degree of G is defined as $\max_{v \in V} d_w(v; E)$. The weighted degree of a vertex measures load on the vertex in applications. For constructing a network with balanced load, it is important to consider weighted degree of networks. Take a communication network for example, and suppose that $w(e, v)$ represents the load (e.g., communication traffic, communication charge) for the communication device on a node v to use a link e incident with v. Then the weighted degree of v indicates the total load of v for using the network. In this paper, we consider constraints on weighted degrees of vertices instead of degree constraints.

We also introduces two types of edges. This is useful for modeling various ways to allocating loads. For an edge $e = uv$ of the first type, weights of e on u and v are given as an input. For an edge $e = uv$ of the second type, the sum of weights of e on u and v is given, and we can decide how much is allocated to the end vertices u and v. In the example of a communication network, the first type models the situation where the administrator decides the charge for each user to use the link, and the second type models the situation where the users can decide by theirselves how much charge they pay.

Weighted Degree Bounded Survivable Network Problem (WDBoundedNetwork): Let $G = (V, E)$ be an undirected graph where E is the union of disjoint sets E_1 and E_2. For those edge sets, weights $w_1 : E_1 \times V \to \mathbb{Q}_+$ and $\mu : E_2 \to \mathbb{Q}_+$ are respectively defined. An input consists of the graph $G = (V, E = E_1 \cup E_2)$ with the weights w_1 and μ, an edge-cost $c : E \to \mathbb{Q}$ ($\mathbb{Q}$ is the set of rational numbers), a skew supermodular set function $f : 2^V \to \mathbb{N}$, and a degree-bound $b : V \to \mathbb{Q}_+$. A solution consists of $F \subseteq E$, weights $w_2(e, u) \in \mathbb{Q}_+$ and $w_2(e, v) \in \mathbb{Q}_+$ for each $e = uv \in F_2$, where F_i denotes $F \cap E_i$. We call w_2 *allocation* of μ when $w_2(e, u) + w_2(e, v) = \mu(e)$ for $e = uv \in F_2$. Throughout this paper, we let $w : F \times V \to \mathbb{Q}_+$ refer to the function that returns $w_i(e, v)$ for $e \in F_i$ and $v \in V$. The solution is defined to be feasible if $G' = (V, F)$ is f-connected, w_2 is an allocation of μ, and degree constraint $d_w(v; F) \leq b(v)$ is satisfied for each $v \in V$. The goal of this problem is to find a feasible solution that minimizes its cost $\sum_{e \in F} c(e)$.

Similarly for the previous problems, we call the instances where $f(X) = 1$ for non-empty $X \subset V$ *weighted degree bounded spanning tree problem* (WDBoundedTree).

It is sometimes more useful to minimize the maximum weighted degree of networks. Hence we also consider a variation of the problem.

Minimum Weighted Degree Survivable Network Problem (MinimumWDNetwork): An input consists of an undirected graph $G = (V, E = E_1 \cup E_2)$ with weights $w_1 : E_1 \times V \to \mathbb{Q}_+$ and $\mu : E_2 \to \mathbb{Q}_+$, and a skew supermodular set function $f : 2^V \to \mathbb{N}$ are given. A feasible solution consists of an f-connected subgraph (V, F) of G and an allocation $w_2 : E_2 \times V \to \mathbb{Q}_+$ of μ. The objective is to minimize the maximum weighted degree $\max_{v \in V} d_w(v; F)$.

Again, we call the instances where $f(X) = 1$ for non-empty $X \subset V$ *minimum weighted degree spanning tree problem* (MinimumWDTree).

For an instance of WDBoundedNetwork and some $\alpha, \beta \geq 1$, we call a solution consisting of $F \subseteq E$ and an allocation w_2 of μ by (α, β)-*approximate solution* if it satisfies

- $\sum_{e \in F} c(e) \leq \alpha \min\{\sum_{e \in F'} c(e) \mid F' \subseteq E$ is in a feasible solution$\}$,
- $d_w(v; F) \leq \beta b(v)$ for all $v \in V$.

So far, there are a few works on the network design problem with weighted degree constraints. All of these correspond to the case with $E_2 = \emptyset$ and $w_1(e, u) = w_1(e, v)$ for $e = uv \in E_1$. Ravi [22] presented an $(O(\log |V|), O(\log |V|))$-approximation algorithm to problem WDBoundedTree. Ghodsi et al. [9] presented a 4.5-approximation algorithm to the metric case of MinimumWDTree while they also showed that it is NP-hard to approximate it within a factor less than 2. Nutov [21] considered a similar problem for digraphs. For problems WDBoundedTree and WDBoundedNetwork, we have proposed algorithms which achieve approximation ratios $(1, 4 + 3\theta)$ and $(2, 7 + 5\theta)$ respectively where θ is defined as $\{b(u)/b(v), b(v)/b(u) \mid uv \in E_2\}$ if $E_2 \neq \emptyset$ and 0 otherwise. For problems MinimumWDTree and MinimumWDNetwork, we have proposed algorithms which achieve approximation ratios $7 + \epsilon$ and $12 + \epsilon$ in polynomial time of $\log(1/\epsilon)$ and input size for an arbitrary $\epsilon > 0$. If $E_2 = \emptyset$, we can remove ϵ from the ratios while the algorithms run in polynomial time of only input size.

The rest of this paper is organized as follows. In Section 2, we briefly review the previous approaches to problem DBoundedNetwork. Section 3 gives our algorithms to problems WDBoundedNetwork and MinimumWDNetwork.

§ 2. Previous Works for DBoundedNetwork

In this section, we briefly review how to solve problem DBoundedNetwork.

As stated in Section 1, it is hard to test the feasibility of instances in this problem. Fukunaga and Nagamochi [7] considered the metric case with the following assumptions: The given graph G is complete and each edge can be picked more than once as multiple edges; The cost function c satisfies the triangle inequality; $f(X)$ is defined as $\max\{r(u, v) \mid u \in X, v \in V - X\}$ from some $r : V \times V \to \mathbb{N}$ such that $r(u, v) \geq 2$ for all $u, v \in V$. With this assumption, it is easy to characterize instances having feasible solutions. Note that even with this assumption, the problem generalizes metric TSP.

Their algorithm is strongly based on a graph operation called *splitting*. Let e denote an edge joining vertices u and s, and e' denote an edge joining vertices v and

s. Splitting the pair $\{e, e'\}$ of the edges incident to s denotes the operation replacing those edges by a new edge joining u and v. This operation decreases the degree of s by 2 while it preserves the degrees of the other vertices. Although it may decrease the edge-connectivity of the graph, several results [1, 6, 14, 20] present conditions for the existence of pairs of edges such that the graph obtained by splitting them satisfies some edge-connectivity requirements. One of the results is the essential part of the algorithm due to Fukunaga and Nagamochi [7].

Theorem 2.1 ([7]). *Let $G = (V, E)$ be an undirected connected graph, and $s \in V$ be a vertex such that $d(s; G) \neq 3$ and no cut-edge is incident to s. Then there exists at least one pair $\{e = us, e' = vs\}$ of edges such that the graph G' obtained from G by splitting $\{e, e'\}$ satisfies $\lambda(x, y; G') = \lambda(x, y; G)$ for all $x, y \in V - \{s\}$ and $\lambda(x, s; G') = \min\{\lambda(x, s; G), d(s; G')\}$ for all $x \in V - \{s\}$.*

Let us describe outline of the algorithm. At first, the algorithm constructs an f-connected graph whose cost is at most $2 + 1/\min\{r(u, v) \mid u, v \in V\}$ times the optimal. Then it transforms the graph into a feasible solution by using Theorem 2.1. Since the splitting does not increase the cost by the triangle inequality, this finally gives a $(2 + 1/\min\{r(u, v) \mid u, v \in V\})$-approximate solution. We note that this is the same approach with the classic approximation algorithms to metric TSP [5].

Another approach for problem DBoundedNetwork is to relax the degree constraint and consider bi-criteria algorithms. By this approach, problem DBoundedTree has been studied extensively in the last two decades [3, 4, 16, 17, 23, 24]. In particular, Goemans [10] gave an algorithm to compute a spanning tree of the minimum cost although it violates degree upper-bounds by at most two. The algorithm obtains such a spanning tree by rounding a basic optimal solution of an LP relaxation with the matroid intersection algorithm. Afterwards this was improved by Singh and Lau [25] using more sophisticated rounding algorithm. It computes a spanning tree of minimum cost which violates degree upper-bounds by at most one. We remark that the algorithm extends the *iterative rounding* due to Jain [13], who applied it for designing a 2-approximation algorithm to the generalized Steiner network problem.

For problem DBoundedNetwork, a bi-criteria algorithm is given by Lau et al. [18]. Their algorithm outputs a network of cost at most twice the optimal and the degree of $v \in V$ is at most $2b(v) + 3$. This is achieved by rounding an optimal basic solution $x^* \in \mathbb{Q}_+^E$ for a linear program:

$$\begin{aligned} \text{minimize} \ & c^T x \\ \text{subject to} \ & x(\delta(U; E)) \geq f(U) \quad \text{for each non-empty } U \subset V, \\ & x(\delta(v; E)) \leq b(v) \quad \text{for each } v \subset A, \\ & x \in \mathbb{Q}_+^E, \end{aligned}$$

where $x(F)$ denotes $\sum_{e\in F} x(e)$ for $F \subseteq E$. Here $A \subseteq V$ is defined as a set of vertices whose degrees are bounded. This is introduced for the rounding algorithm to work in an inductive manner.

Let x^* be an optimal basic solution for the linear program, and E_{x^*} be $\{e \in E \mid x^*(e) > 0\}$. Lau et al. [18] observed that x^* satisfies at least one of the following conditions for any c:

- There exists an edge $e \in E_{x^*}$ such that $x^*(e) \geq 1/2$;
- There exists a vertex $v \in A$ such that $|\delta(v; E_{x^*})| \leq 4$.

This property enables the algorithm to round x^* into a good solution although we omit the detail.

The result due to Lau et al. [18] was improved by Lau and Singh [19] for some special cases. For example, if $f(U)$ is defined from a demand $r : V \times V \to \mathbb{N}$ as $\max\{r(u, v) \mid u \in U, v \in V - U\}$ for $U \subseteq V$ (i.e., feasible solutions are Steiner networks), then the degree of a vertex $v \in V$ in a solution is guaranteed to be at most $b(v)+3+6\max_{u,v\in V} r(u, v)$. Furthermore, if $r(u, v) \in \{0, 1\}$ for every $u, v \in V$ (i.e., feasible solutions are Steiner forests), then the degree is at most $b(v) + 3$.

Inspired by these results, the iterative rounding has been applied to many optimization problems with degree bounds. Bansal et al. [2] applied the iterative rounding for the degree bounded arborescence problem and degree bounded survivable network problem with intersecting supermodular connectivity. Kiraly et al. [15] generalized degree bounded spanning trees to degree bounded matroid bases. Kiraly et al. [15] also considered degree bounded submodular flow problem.

§ 3. Survivable Network with Weighted Degree Constraints

The section describes a bi-criteria approximation algorithm for problem WD-BoundedNetwork. We let I stand for the set of an undirected graph $G = (V, E)$ with $E = E_1 \cup E_2$, weights $w_1 : E_1 \times V \to \mathbb{Q}_+$ and $\mu : E_2 \to \mathbb{Q}_+$, a skew supermodular set function $f : 2^V \to \mathbb{N}$, a subset A of V, and $b : A \to \mathbb{Q}_+$. We denote by $\mathrm{P_N}(I)$ the polytope that consists of vectors $x \in \mathbb{Q}^E$ and $y \in \mathbb{Q}^{E_2\times V}$ that satisfy

$$0 \leq x(e) \leq 1 \text{ for all } e \in E, \tag{3.1}$$

$$0 \leq y(e, u),\ y(e, v) \text{ for all } e = uv \in E_2, \tag{3.2}$$

$$y(e, u) + y(e, v) = x(e) \text{ for all } e = uv \in E_2, \tag{3.3}$$

$$x(\delta(U)) \geq f(U) \text{ for all non-empty } U \subset V, \tag{3.4}$$

and

$$\sum_{e\in\delta(v;E_1)} w_1(e, v)x(e) + \sum_{e\in\delta(v;E_2)} \mu(e)y(e, v) \leq b(v) \quad \text{for all } v \in A. \tag{3.5}$$

Observe that $\min\{c^T x \mid (x,y) \in \mathrm{P_N}(I)\}$ with $A = V$ is an LP relaxation of problem WDBOUNDEDNETWORK.

We say that $\mathrm{P_N}(I)$ is (α, β)-*bounded* for some $\alpha, \beta \geq 1$ if every extreme point (x^*, y^*) of the polytope satisfies at least one of the following:

- There exists an edge $e \in E_{x^*}$ such that $x^*(e) \geq 1/\alpha$;
- There exists a vertex $v \in A$ such that $|\delta(v; E_{x^*})| \leq \beta$.

Notice that this notion has appeared in Lau et al. [18]. Below we describe an approximation algorithm that works when $\mathrm{P_N}(I)$ has this property. Afterwards, we show that polytope $\mathrm{P_N}(I)$ is $(2,5)$-bounded. Combining these tells that problem WDBOUNDEDNETWORK can be solved approximately.

Algorithm for problem WDBOUNDEDNETWORK

Input: An undirected graph $G = (V, E)$ with $E = E_1 \cup E_2$, weights $w_1 : E_1 \times V \to \mathbb{Q}_+$ and $\mu : E_2 \to \mathbb{Q}_+$, an edge-cost $c : E \to Q$, a skew supermodular set function $f : 2^V \to \mathbb{N}$, and a degree-bound $b : V \to \mathbb{Q}_+$.

Output: A solution consisting of an f-connected subgraph (V, F) of G and an allocation $w_2 : F_2 \times V \to \mathbb{Q}_+$ of μ, or message "INFEASIBLE".

Step 1: Set $A := V$ and $F := \emptyset$.

- Delete $e = uv \in E_1$ from G if $w_1(e, u) > b(u)$ or if $w_1(e, v) > b(v)$.
- Delete $e = uv \in E_2$ from G if $\mu(e) > b(u) + b(v)$.

If $\mathrm{P_N}(I) = \emptyset$, then output "INFEASIBLE";

Step 2: Compute a basic solution (x^*, y^*) that minimizes $\sum_{e \in E} c(e)x^*(e)$ over $(x^*, y^*) \in \mathrm{P_N}(I)$;

Step 3: Remove edges in $E - E_{x^*}$ from E;

Step 4: If there exists an edge $e = uv \in E$ such that $x^*(e) \geq 1/\alpha$, then add e to F, delete e from E, set $f(U) := f(U) - 1$ for all $U \subset V$ with $e \in \delta(U)$. Moreover, execute one of the following operations according to the class of e:

Case of $e \in E_1$**:** If $u \in A$, then set $b(u) := b(u) - w_1(e, u)x^*(e)$. If $v \in A$, then set $b(v) := b(v) - w_1(e, v)x^*(e)$.

Case of $e \in E_2$**:** Set $w_2(e, u) := \mu(e)y^*(e, u)/x^*(e)$ and $w_2(e, v) := \mu(e)y^*(e, v)/x^*(e)$. If $u \in A$, then set $b(u) := b(u) - \mu(e)y^*(e, u)$. If $v \in A$, then set $b(v) := b(v) - \mu(e)y^*(e, v)$;

Step 5: If there exists a vertex $v \in A$ such that $|\delta(v; E_{x^*})| \leq \beta$, then remove v from A;

Step 6: If $E = \emptyset$, then output F as a solution, and terminate. Otherwise, return to Step 2.

Now we define $\theta = \max\{b(u)/b(v), b(v)/b(u) \mid uv \in E_2\}$ if $E_2 \neq \emptyset$, and $\theta = 0$ otherwise.

Theorem 3.1. *If each* $\mathrm{P_N}(I)$ *constructed in Step 2 of the algorithm is* (α, β)*-bounded, then problem* WDBOUNDEDNETWORK *is* $(\alpha, \alpha + \beta(1 + \theta))$*-approximable in polynomial time.*

Proof. It is clear that the algorithm described above runs in polynomial time. In what follows, we see that the algorithm computes an $(\alpha, \alpha + \beta(1 + \theta))$-approximate solution.

Observe that the linear program over $\mathrm{P_N}(I)$ is still a relaxation of the given instance after Step 1. Hence the original instance has no feasible solutions when the algorithm outputs "INFEASIBLE". Each edge $e = uv \in E$ satisfies the following properties after Step 1:

- If $e = uv \in E_1$, then $w_1(e, u) \leq b(u)$ and $w_1(e, v) \leq b(v)$;
- If $e = uv \in E_2$, then $\mu(e) \leq b(u) + b(v) \leq (1 + \theta)b(u)$ and $\mu(e) \leq b(u) + b(v) \leq (1 + \theta)b(v)$.

In what follows, suppose that $\mathrm{P_N}(I) \neq \emptyset$ after Step 1. We then prove that $\mathrm{P_N}(I) \neq \emptyset$ also throughout the subsequent iterations and that the edge set F outputted by the algorithm satisfies $c(F) \leq \alpha \min\{c^T x \mid (x, y) \in \mathrm{P_N}(I)\}$, and $d_w(v; F) \leq (\alpha+\beta(1+\theta))b(v)$ for all $v \in V$.

Let $e_i = u_i v_i$ denote the i-th edge added to F, $I_i = (G_i = (V, E^i), w_1, \mu, \nu, f_i, A_i, b_i)$ denote I at the beginning of the iteration in which e_i is added to T, and (x_i^*, y_i^*) denote the basic solution computed in Step 2 of that iteration. We also let I_0 stand for I immediately after Step 1 of the algorithm, and assume that the algorithm outputs $F = \{e_1, \ldots, e_j\}$. By Steps 4 and 5, $A_{i+1} \subseteq A_i$ holds, and

$$b_{i+1}(v') = \begin{cases} b_i(v') - w_1(e_i, v')x_i^*(e_i) & \text{if } v' \in A \text{ and } e_i \in E_1, \\ b_i(v') - \mu(e_i)y_i^*(e_i, v') & \text{if } v' \in A \text{ and } e_i \in E_2, \\ b_i(v') & \text{otherwise.} \end{cases} \tag{3.6}$$

also holds for $v' \in \{u_i, v_i\}$, $i \geq 1$. Moreover, all edges in $E_{i+1} - E_i$ except e_i are those such that the corresponding variable of x_i^* took 0. These facts indicate that the projection of (x_i^*, y_i^*) satisfies all constraints in $\mathrm{P_N}(I_{i+1})$. Hence we have the following:

$$\text{If } \mathrm{P_N}(I_i) \neq \emptyset, \text{ then } \mathrm{P_N}(I_{i+1}) \neq \emptyset \text{ for } i \geq 0; \tag{3.7}$$

(3.8)

$$c^T x_i^* \geq c(e_i)x_i^*(e_i) + \min\{c^T x \mid (x,y) \in \mathrm{P_N}(I_{i+1})\} \geq c(e_i)x_i^*(e_i) + c^T x_{i+1}^* \quad \text{for } i \geq 1.$$

(i) We first see that the algorithm outputs a solution. Recall that we are assuming that $\mathrm{P_N}(I_0) \neq \emptyset$. By this and (3.7), $\mathrm{P_N}(I_i) \neq \emptyset$ holds for all $1 \leq i \leq j$. Hence the algorithm terminates with outputting an f-connected subgraph $F = \{e_1, \ldots, e_j\}$ and an allocation $w_2 : F_2 \times V \to \mathbb{Q}_+$ of μ by the way of construction.

(ii) Next we see the α-approximability of $c(F)$. By applying (3.8) repeatedly, we have

$$c^T x_1^* \geq c(e_1)x_1^*(e_1) + c^T x_2^* \geq \cdots \geq \sum_{i=1}^{j-1} c(e_i)x_i^*(e_i) + c^T x_j^* \geq \sum_{i=1}^{j} c(e_i)x_i^*(e_i).$$

Notice that $x_i^*(e_i) \geq 1/\alpha$ holds for all $1 \leq i \leq j$ by the condition of Step 4. Hence,

$$\sum_{i=1}^{j} c(e_i)x_i^*(e_i) \geq c(F)/\alpha,$$

implying that $\alpha c^T x_1^* \geq c(F)$. Recall that the algorithm constructs I_1 from I_0 by relaxing the degree constraints (i.e., $A_1 \subseteq A_0$). Hence $\min\{c^T x \mid (x,y) \in \mathrm{P_T}(I_0)\} \geq c^T x_1^*$. Therefore we have $\alpha \min\{c^T x \mid (x,y) \in \mathrm{P_N}(I_0)\} \geq c(F)$, as required.

(iii) Fix v as an arbitrary vertex. Now we prove that $d_w(v;F) \leq (\alpha + \beta(1+\theta))b(v)$ holds.

Consider Step 4 of the iterations during $v \in A$. Let F' be the set of edges that are added to F during those iterations. By applying (3.6) repeatedly, we obtain

$$b(v) \geq \sum_{e_i \in \delta(v;F_1')} w_1(e_i,v)x_i^*(e_i) + \sum_{e_i \in \delta(v;F_2')} \mu(e_i)y_i^*(e_i,v).$$

If $e_i \in \delta(v;E_2)$, then $w_2(e_i,v) = \mu(e_i)y_i^*(e_i,v)/x_i^*(e_i)$. Recall that $x_i^*(e_i) \geq 1/\alpha$. Therefore,

$$\sum_{e_i \in \delta(v;F_1')} w_1(e_i,v)x_i^*(e_i) + \sum_{e_i \in \delta(v;F_2')} \mu(e_i)y_i^*(e_i,v) \geq d_{w_1}(v;F_1')/\alpha + d_{w_2}(v;F_2')/\alpha.$$

It implies that $d_w(v;F') \leq \alpha b(v)$ holds.

Consider the iterations after v is removed from A. Let F'' denote the set of edges that are added to F during those iterations. When v is removed from A in Step 5, the number of remaining edges incident with v is at most β by the condition in Step 5. Hence $|\delta(v;F'')| \leq \beta$ holds. We have already seen that, after Step 1, $e = uv \in E_1$ satisfies $w_1(e,v) \leq b(v)$ and $e = uv \in E_2$ satisfies $w_2(e,v) \leq \mu(e) \leq (1+\theta)b(v)$. So $d_w(v;F'') \leq \beta(1+\theta)b(v)$. Because $d_w(v;F) = d_w(v;F') + d_w(v;F'')$, we have $d_w(v;F) \leq (\alpha + \beta(1+\theta))b(v)$, which completes the claim. □

Theorem 3.2. *Suppose that problem* WDBOUNDEDNETWORK *with uniform b is (α', β')-approximable for some α' and β'. For an arbitrary $\epsilon > 0$, problem* MINIMUMWDNETWORK *is $(\beta' + \epsilon)$-approximable in polynomial time of $\log(1/\epsilon)$ and the input size. If $E_2 = \emptyset$, then it is β'-approximable in polynomial time of only the input size.*

Proof. For an $r \in \mathbb{Q}$, define G_r as the subgraph obtained from G by deleting each edge $e = uv \in E_1$ such that $\max\{w_1(e, u), w_1(e, v)\} > r$ and each edge $e \in E_2$ such that $\mu(e) > 2r$. Let $b_r : V \to \mathbb{Q}_+$ be the function such that $b_r(v) = r$ for all $v \in V$, and $I_r = (G_r, w_1, \mu, f, A = V, b_r)$ be the instance for problem WDBOUNDEDNETWORK.

We denote $\min\{r \in \mathbb{Q}_+ \mid \mathrm{P_N}(I_r) \neq \emptyset\}$ by R, and the minimum weighted degree of the given instance by OPT. Let ω and W stand for the minimum and maximum entry of w_1 and μ, respectively. For given ϵ, define $\epsilon' = \epsilon\omega/(2\beta')$. Since $\omega/2 \leq \mathrm{OPT}$, we have $\epsilon' \leq \epsilon\mathrm{OPT}/\beta'$. Since the characteristic vector of an optimal solution to the given instance of problem MINIMUMWDNETWORK satisfies all constraints of $\mathrm{P_N}(I_{OPT})$, we have $R \leq \mathrm{OPT}$. It is possible to compute a value R' such that $R \leq R' \leq R + \epsilon'$ by the binary search on interval $[0, W]$, which needs to solve the linear program over $\mathrm{P_N}(I_r)$ $\log(W/\epsilon')$ times.

Let T be an (α', β')-approximate solution to the instance of problem WDBOUNDEDNETWORK consisting of $I_{R'}$ and an arbitrary edge-cost c. We then have $d_w(v; T) \leq \beta' b_{R'}(v) \leq \beta'(R + \epsilon') \leq (\beta' + \epsilon)\mathrm{OPT}$ for any $v \in V$. This implies that T is a $(\beta' + \epsilon)$-approximate solution to problem MINIMUMWDNETWORK.

When $E_2 = \emptyset$, set ϵ so that $1/(\psi + 1) \leq \epsilon' < 1/\psi$ holds, where ψ is the maximum denominator of all entries in w_1 and μ. In this case, if R' satisfies $R \leq R' \leq R + \epsilon'$, then $R' \leq \mathrm{OPT}$. Such R' can be computed by solving the linear program $\log(W/\epsilon') \leq \log(W(\psi + 1))$ times. Then we have $d_w(v; T) \leq \beta' b_{R'}(v) \leq \beta'\mathrm{OPT}$ for any $v \in V$, which implies that T is a β'-approximate solution. □

For proving the $(2, 5)$-boundedness of $\mathrm{P_N}(I)$, let us see that the key property of tight constraints observed in [13] also holds in our setting.

Lemma 3.3. *Let (x^*, y^*) be any extreme point of $\mathrm{P_N}(I)$ and suppose that $x^*(e) < 1$ for all $e \in E$. There exists a laminar family $\mathcal{L} \subseteq 2^V$ and a subset X of A such that characteristic vectors of $\delta(U; E_{x^*})$ for $U \in \mathcal{L}$ are linearly independent and $|E_{x^*}| \leq |\mathcal{L}| + |X|$.*

Proof. By the definitions of x^* and y^*, the number of variables is equal to the dimension of the vector space spanned by the coefficients vectors of tight constraints in $\mathrm{P_N}(I)$. If $x^*(e) = 0$ (resp., $y^*(e, v)$), then fix the variable $x(e)$ (resp., $y(e, v)$) to 0 and remove the corresponding tight constraint of (3.1) (resp., (3.2)). We can also remove the constraints (3.3) by fixing $y(e,u)$ to $x(e) - y(e,v)$ for all $e = uv \in E_2$. Then the number of remaining variables, which is at least $|E_{x^*}|$, is equal to the dimension of the vector space spanned by the tight constraints of (3.4) and (3.5).

Let $\mathcal{F} = \{U \subset V \mid U \neq \emptyset, x^*(\delta(U)) = f(U)\}$ (i.e., family of vertex subsets defining tight constraints of (3.4)) and $X = \{v \in A \mid \sum_{e\in\delta(v;E_1)} w_1(e,v)x^*(e) + \sum_{e\in\delta(v;E_2)} \mu(e)y^*(e,v) = b(v)\}$ (i.e., set of vertices defining tight constraints of (3.5)). In [13], it is proven that a maximal laminar subfamily $\mathcal{F}'$ of $\mathcal{F}$ spans the same vector space with $\mathcal{F}$. Therefore we can obtain the required $\mathcal{L}$ and X by removing linearly dependent sets from $\mathcal{F}'$. □

Theorem 3.4. *Polytope* $\mathrm{P_N}(I)$ *is* $(2,5)$*-bounded for any* I*.*

Proof. Suppose the contrary, i.e., all edges $e \in E_{x^*}$ satisfy $x^*(e) < 1/2$, and all vertices $v \in A$ satisfy $|\delta(v;E_{x^*})| \geq 6$.

Let $\mathcal{L}$ and X be those in Lemma 3.3. We define a child-parent relationship between all elements in $\mathcal{L}$ and X as follows: For $U \in \mathcal{L}$ or $v \in X$, define its parent as the inclusion-wise minimal element in $\mathcal{L}$ that contains it if any. Note that when $v \in X$ and $\{v\} \in \mathcal{L}$, $\{v\}$ is the parent of v.

We assign one token to each end vertex of edges in E_{x^*}. Define the *co-requirement* of $U \in \mathcal{L}$ as $|\delta(U;E_{x^*})|/2 - f(U)$. Following the approach in [13], we observe that it is possible to distribute these tokens to all elements in $\mathcal{L}$ and in X so that

- each element having the parent owns two tokens,
- each element having no parent owns at least three tokens,
- and it owns exactly three only if its co-requirement equals to 1/2.

First two of these mean that the number of all tokens is more than $2(|\mathcal{L}| + |X|)$. Since the number of tokens is exactly $2|E_{x^*}|$, this indicates that $|E_{x^*}| > |\mathcal{L}| + |X|$, which contradicts $|E_{x^*}| \leq |\mathcal{L}| + |X|$.

We prove the claim inductively. The base case is when the elements have no child. An element $v \in X$ owns at least six tokens by $|\delta(v;E_{x^*})| \geq 6$. An element $U \in \mathcal{L}$ with no child owns at least three tokens because $|\delta(U;E_{x^*})| \geq 3$ by $x^*(e) < 1/2$ for each $e \in \delta(U;E_{x^*})$ and $f(U) \geq 1$. It owns exactly three tokens if and only if $|\delta(U;E_{x^*})| = 3$. Since $|\delta(U;E_{x^*})| = 3$ indicates that $f(U) = 1$, it means the co-requirement $|\delta(U;E_{x^*})|/2 - f(U)$ equals to 1/2 .

Let us consider the case in which an element $U \in \{\mathcal{L}\}$ has some children. If U has children from X, then it is possible to redistribute tokens so that U owns at least four tokens, and each child owns two tokens. If the children of U are all from $\mathcal{L}$, then the argument is proven in [13]. □

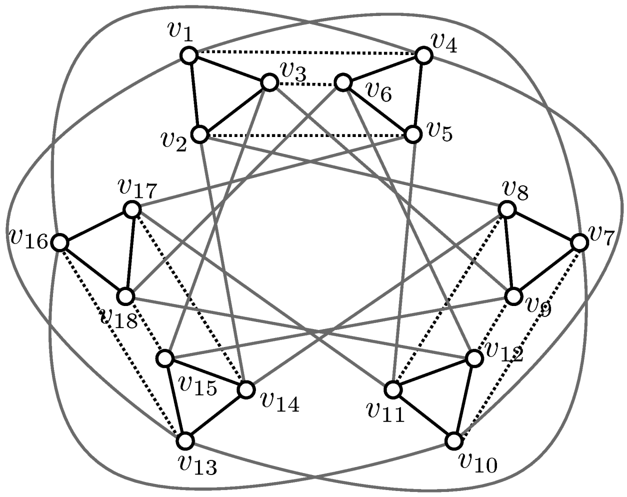

Figure 1. A counterexample for $(2, 4)$-boundedness of $\mathrm{P_N}(I)$

Corollary 3.5. *Problem* WDBOUNDEDNETWORK *is* $(2, 7 + 5\theta)$*-approximable in polynomial time. Problem* MINIMUMWDNETWORK *is* 7*-approximable in polynomial time if* $E_2 = \emptyset$*, and is* $(12 + \epsilon)$*-approximable in polynomial time of* $\log(1/\epsilon)$ *and the input size for any* $\epsilon > 0$ *otherwise.*

Proof. Immediate from Theorems 3.1, 3.2 and 3.4. □

As explained in Section 1, Lau et. al. [18] designed their algorithm for $w_1(e, u) = w_1(e, v) = 1$, $e = uv \in E_1$ and $E_2 = \emptyset$ by observing that $\mathrm{P_N}(I)$ is $(2, 4)$-bounded with such instances. However, an example indicates that this does not hold in our problem even if $w_1(e, u) = w_1(e, v)$ for all $e = uv \in E_1$ and $E_2 = \emptyset$.

Let G be the graph in Figure 1, $f(U) = 1$ for all non-empty $U \subset V$, and $A = V$. We suppose that $|E| = |E_1| = 42$ tight constraints consists of (3.4) for all singletons, for $\{v_i, v_{i+1}, v_{i+2}\}$ with $i = 1, 4, 7, 10, 13, 16$, and for $\{v_i, v_{i+1}, v_{i+2}, v_{i+3}, v_{i+4}, v_{i+5}\}$ with $i = 1, 7, 13$, and (3.5) for all vertices. We set w_1 so that the above tight constraints are linearly independent. Setting b appropriately, we then have a basic optimal solution x^* such that

$$x^*(e) = \begin{cases} 1/3 & \text{for edges represented by black solid lines,} \\ 1/6 & \text{for edges represented by dotted lines,} \\ 1/12 & \text{for edges represented by gray solid lines.} \end{cases}$$

Notice that $x^*(e) < 1/2$ for all $e \in E$ and $|\delta(v; E_{x^*})| \geq 5$ for all $v \in V$.

§ 4. Concluding Remarks

In this paper, we have presented approximation algorithms to problems WD-BOUNDEDNETWORK and MINIMUMWDNETWORK. We also have seen that it is hard to improve the approximation ratios by our approach based on the iterative rounding method.

For obtaining better approximation ratios to problems WDBOUNDEDTREE and MINIMUMWDTREE, we need stronger LP relaxations. Explaining more concretely, the relaxation obtained by replacing (3.4) with $\sum_{e \subseteq U} x(e) \leq |U| - 1$, $\emptyset \neq U \subset V$ and $x(E) = |V| - 1$ gives our results for the problems.

References

[1] Bang-Jensen, J. and Jordán, T., Splitting off edges within a specified subset preserving the edge-connectivity of the graph, *Journal of Algorithms*, **37** (2000), 326–343.

[2] Bansal, N., Khandekar, R., and Nagarajan, V., Additive guarantees for degree bounded directed network design, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, 2008, 769–778.

[3] Chaudhuri, K., Rao, S., Riesenfeld, S., and Talwar, K., What would Edmonds do? augmenting paths and witnesses for degree-bounded MSTs, *Proceedings of 8th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, Lecture Notes in Computer Science*, **3624** (2005), 26–39.

[4] Chaudhuri, K., Rao, S., Riesenfeld, S., and Talwar, K., A push-relabel algorithm for approximating degree bounded MSTs, *Proceedings of 33rd International Colloquium on Automata, Languages and Programming, Lecture Notes in Computer Science*, **4051** (2006), 191–201.

[5] Christofides, N., Worst-case analysis of a new heuristics for the traveling salesman problem, Technical Report, Graduate School of Industrial Administration, Carnegie-Mellon University, 1976.

[6] Frank, A., On a theorem of Mader, *Discrete Mathematics*, **191** (1992), 49–57.

[7] Fukunaga, T. and Nagamochi, H., Approximating minimum cost multigraphs of specified edge-connectivity under degree bounds, *Journal of the Operations Research Society of Japan,* **50** (2007), 339–350.

[8] Fürer, M. and Raghavachari, B., Approximating the minimum-degree Steiner tree to within one of optimal, *Journal of Algorithms*, **17** (1994), 409–423.

[9] Ghodsi, M., Mahini, H., Mirjalali, K., Gharan, S.O., Sayedi R., A.S., and Zadimoghaddam, M., Spanning trees with minimum weighted degrees, *Information Processing Letters*, **104** (2007), 113–116.

[10] Goemans, M.X., Minimum bounded-degree spanning trees, *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, 2006, 273–282.

[11] Goemans, M.X. and Williamson, D.P., The primal-dual method for approximation algorithms and its application to network design problems, *Approximation Algorithms for NP-Hard Problems*, 1997, 144–191.

[12] Grötschel, M., Lovász, L., and Schrijver, A., *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, 1988.

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

[14] Jordán, T., Constrained edge-splitting problems, *SIAM Journal of Discrete Mathematics*, **17** (2003), 88–102.

[15] Király, T., Lau, L.C., and Singh, M., Degree bounded matroids and submodular flows, *Proceedings of 13th Conference on Integer Programming and Combinatorial Optimization, Lecture Notes in Computer Science*, **5035** (2008), 259–272.

[16] Könemann, J. and Ravi, R., A matter of degree: Improved approximation algorithms for degree-bounded minimum spanning trees, *SIAM Journal on Computing*, **31** (2002), 1783–1793.

[17] Könemann, J. and Ravi, R., Primal-dual meets local search: approximating MST's with nonuniform degree bounds, *SIAM Journal on Computing*, **34** (2005), 763–773.

[18] Lau, L.C., Naor, J.S., Singh, M., and Salavatipour, M.R., Survivable network design with degree or order constraints, *Proceedings of the 39th ACM Symposium on Theory of Computing*, 2007, 651–660.

[19] Lau, L.C. and Singh, M., Additive approximation for bounded degree survivable network design, *Proceedings of the 40th ACM Symposium on Theory of Computing*, 2008, 759–768.

[20] Mader, W., A reduction method for edge-connectivity in graphs, *Annals of Discrete Mathematics*, **3** (1978), 145–164.

[21] Nutov, Z., Approximating directed weighted-degree constrained networks, *Proceedings of 11th International Workshop, APPROX 2008, and 12th International Workshop, RANDOM 2008, Lecture Notes in Computer Science*, **5171** (2008), 219–232.

[22] Ravi, R., Steiner Trees and Beyond: Approximation Algorithms for Network Design, PhD thesis, Department of Computer Science, Brown University, 1993.

[23] Ravi, R., Marathe, M.V., Ravi, S.S., Rosenkrantz, D.J. and Hunt III, H.B., Many birds with one stone: Multi-objective approximation algorithms, *Proceedings of the 25th ACM Symposium on Theory of Computing*, 1993, 438–447.

[24] Ravi, R. and Singh, M., Delegate and conquer: An LP-based approximation algorithm for minimum degree MSTs, *Proceedings of 33rd International Colloquium on Automata, Languages and Programming, Lecture Notes in Computer Science*, **4051** (2006), 169–180.

[25] Singh, M. and Lau, L.C., Approximating minimum bounded degree spanning trees to within one of optimal, *Proceedings of the 39th ACM Symposium on Theory of Computing*, 2007, 661–670.