

Network Design with Weighted Degree Constraints*

Takuro Fukunaga, Hiroshi Nagamochi

Department of Applied Mathematics and Physics,
Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan
{takuro,nag}@amp.i.kyoto-u.ac.jp

Abstract

In an undirected graph $G = (V, E)$ with a weight function $w : E \times V \rightarrow \mathbb{Q}_+$, the weighted degree $d_w(v; E)$ of a vertex v is defined as $\sum \{w(e, v) \mid e \in E \text{ incident to } v\}$. In this paper, we consider a network design problem which has upper-bounds on weighted degrees of vertices as its constraints while the objective is to compute a minimum cost graph with a prescribed connectivity. We propose bi-criteria approximation algorithms based on the iterative rounding, which has been successfully applied to the degree-bounded network design problem. A problem of minimizing the maximum weighted degree of vertices is also discussed.

Key words: connectivity, degree constraint, iterative rounding, network design, spanning tree

1 Introduction

Let $G = (V, E)$ be an undirected graph. A weight function $w : E \times V \rightarrow \mathbb{Q}_+$ is defined on pairs of edges and their end vertices, where \mathbb{Q}_+ is the set of non-negative rational numbers. Let $\delta(v; E)$ denote the set of edges in E incident to $v \in V$. 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 to v . Then the weighted degree of v indicates the total load of v for using the network.

In this paper, we consider a network design problem which has upper-bounds on weighted degrees of vertices as its constraints while the objective is to compute a minimum cost graph of a prescribed connectivity. In the above example of the communication network, this corresponds to the case in which each node has an upper-limit on the load that can be handled on the node.

The problem introduces two types of edges. This is useful for modeling various ways of allocating loads. For an edge $e = uv$ of the first type, weights of e on u and v are given as inputs. 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 end vertices u and v .

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 . Let \mathbb{N} be the set of natural numbers. For a given set function $f : 2^V \rightarrow \mathbb{N}$ on V , a graph $G' = (V, F)$ is called *f-connected* when $|\delta(U; F)| \geq f(U)$ holds for every non-empty set

*This work was partially supported by Grant-in-Aid for Scientific Research from the Ministry of Education, Culture, Sports, Science and Technology of Japan.

$U \subset V$. A set function f is called *skew supermodular* 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 every pair of sets $X, Y \subseteq V$. With a skew supermodular set function, f -connectivity represents a wide variety of connectivity of graphs. For example, when $f(U)$, $U \subset V$ is defined as $\max_{u \in U, v \in V-U} r(u, v)$ by a function $r : V \times V \rightarrow \mathbb{N}$, f -connectivity requires the local edge-connectivity between every two vertices u and v to be at least $r(u, v)$.

Now we formulate our problem.

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 $w_1 : E_1 \times V \rightarrow \mathbb{Q}_+$ and $\mu : E_2 \rightarrow \mathbb{Q}_+$ are respectively defined. As inputs, we are given the graph $G = (V, E = E_1 \cup E_2)$ with the weights w_1 and μ , an edge-cost $c : E \rightarrow \mathbb{Q}$ (\mathbb{Q} is the set of rational numbers), a skew supermodular set function $f : 2^V \rightarrow \mathbb{N}$, and a degree-bound $b : V \rightarrow \mathbb{Q}_+$. A solution consists of $F \subseteq E$, and weights $w_2(e, u), w_2(e, v) \in \mathbb{Q}_+$ for each edge $e = uv \in F_2$, where F_i denotes $F \cap E_i$. We call w_2 an *allocation* of μ if $w_2(e, u) + w_2(e, v) = \mu(e)$ holds for each $e = uv \in F_2$. From F , w_1 and w_2 , $w : F \times V \rightarrow \mathbb{Q}_+$ is defined as the function such that $w(e, v) = w_1(e, v)$ if $e \in F_1$, and $w(e, v) = w_2(e, v)$ otherwise. 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 $\sum_{e \in F} c(e)$.

If $f(U) = 1$ for all non-empty $U \subset V$, then the minimal solutions of this problem are spanning trees. We particularly call such instances *weighted degree bounded spanning tree problem (WDBOUNDEDTREE)*.

Feasible solutions of WDBOUNDEDTREE are Hamiltonian paths when $E_2 = \emptyset$, $w_1(e, u) = w_1(e, v) = 1$ for all $e = uv \in E_1$, and $b(v) = 2$ for all $v \in V$. This means that it is NP-hard to test whether an instance of WDBOUNDEDTREE (and hence WDBOUNDEDNETWORK) has a feasible solution or not. By this reason, it is natural to relax the degree constraints and consider bi-criteria approximation algorithms. We say that, for an instance of WDBOUNDEDNETWORK and some $\alpha, \beta \geq 1$, a solution consisting of $F \subseteq E$ and an allocation w_2 of μ is an (α, β) -*approximate solution* if it satisfies

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

Define θ as $\max\{b(u)/b(v), b(v)/b(u) \mid uv \in E_2\}$ if $E_2 \neq \emptyset$, and 0 otherwise. For WDBOUNDEDTREE and WDBOUNDEDNETWORK, we propose algorithms which achieve approximation ratios $(1, 4 + 3\theta)$ and $(2, 7 + 5\theta)$ respectively. Our algorithms take the approach successfully applied to the bounded degree spanning tree problem by Singh and Lau [1] and to the bounded-degree survivable network design problem by Lau et al. [2], which correspond to instances with uniform w_1 and $E_2 = \emptyset$ in our problems. Their approach is based on the iterative rounding originally used for the generalized Steiner network problem by Jain [3]. Roughly illustrating, they iterate rounding fractional variables in basic optimal solutions or removing constraints of a linear program relaxation. The key for guaranteeing the correctness of the algorithm is an analysis of the structure of tight constraints which determine the basic optimal solutions. In this paper, we show that this approach remains useful in our problems.

In addition, we also discuss the following variation of the above problem.

Minimum weighted degree survivable network problem (MINIMUMWDNETWORK): An undirected graph $G = (V, E)$ with a partition E_1, E_2 of E , weights $w_1 : E_1 \times V \rightarrow \mathbb{Q}_+$ and

$\mu : E_2 \rightarrow \mathbb{Q}_+$, and a skew supermodular set function $f : 2^V \rightarrow \mathbb{N}$ are given. A feasible solution consists of an f -connected subgraph $G' = (V, F)$ of G and an allocation $w_2 : F_2 \times V \rightarrow \mathbb{Q}_+$ of μ . The objective is to minimize the weighted degree $\max_{v \in V} d_w(v; F)$ of G' .

Similarly for WDBOUNDEDNETWORK, we call instances with $f(U) = 1$ for all non-empty $U \subset V$ *minimum weighted degree spanning tree problem* (MINIMUMWDTREE).

For MINIMUMWDTREE and MINIMUMWDNETWORK, our algorithms 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 next theorems summarize our contribution.

Theorem 1. WDBOUNDEDTREE is $(1, 4 + 3\theta)$ -approximable in polynomial time. MINIMUMWDTREE is 4-approximable in polynomial time if $E_2 = \emptyset$, and is $(7 + \epsilon)$ -approximable in polynomial time of $\log(1/\epsilon)$ and input size for any $\epsilon > 0$ otherwise.

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

Previous Works

The bounded degree spanning tree problem has been studied extensively in the last two decades [4, 5, 6, 7, 8, 9]. For the uniform cost (i.e., $c(e) = 1$ for $e \in E$), an optimal result was given by Fürer and Raghavachari [10]. Their algorithm computes a spanning tree which violates degree upper-bounds by at most one. For general costs, Goemans [11] 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 an optimal result for general cost was presented by Singh and Lau [1]; Their algorithm computes a spanning tree of minimum cost which violates degree upper-bounds by at most one. As mentioned above, their result is achieved by extending the iterative rounding due to Jain [3], who applied it for designing a 2-approximation algorithm to the generalized Steiner network problem.

This approach is also applied to several problems related to degree bounds. Lau et al. [2] considered the survivable network problem, and proposed an algorithm that outputs a network of cost at most twice the optimal and the degree of $v \in V$ is at most $2b(v) + 3$. This result was improved in Lau and Singh [12]. Bansal et al. [13] considered the arborescence problem and survivable network problem with intersecting supermodular connectivity. Kiraly et al. [14] generalized bounded degree spanning tree to bounded degree matroid. They also considered degree bounded submodular flow problem.

There are also several works on the network design problem with weighted degree constraints. All of these correspond to the case with $E_2 = \emptyset$ and $w_1(e, u) = w_2(e, v)$ for $e = uv \in E_1$. Ravi [15] presented an $O(\log |V|, \log |V|)$ -approximation algorithm to WDBOUNDEDTREE and an $O(\log |V|)$ -approximation algorithm to MINIMUMWDTREE. For MINIMUMWDTREE, Ghodsi et al. [16] presented a 4.5-approximation algorithm under the assumption that G is a complete graph and c is a metric cost (i.e., triangle inequality holds) while they also showed that it is NP-hard to approximate it within a factor less than 2. Notice that our algorithm described in this paper achieves $(1, 4)$ -approximation to WDBOUNDEDTREE and 4-approximation to MINIMUMWDTREE when $E_2 = \emptyset$. Hence it improves these previous works. Nutov [17] considered WDBOUNDEDNETWORK for digraphs.

Organization

The rest of this paper is organized as follows. Section 2 proves Theorem 1 The algorithms are derived from a good property of polytopes that give a linear program relaxation of the problems.

Section 2 also shows that our analysis on the property is tight. Section 3 proves Theorem 2 and shows that our analysis on the property of polytopes is tight. Section 4 concludes this paper.

2 Spanning Trees with Weighted Degree Constraints

We prove Theorem 1 in this section. For our algorithm to work recursively, we need to generalize WDBOUNDEDTREE by defining b as a function $A \rightarrow \mathbb{Q}_+$ for some $A \subseteq V$. This means that the degree upper-bound is defined only on the vertices in A .

Let \mathcal{I} stand for the instance of WDBOUNDEDTREE consisting of an undirected graph $G = (V, E)$ with $E = E_1 \cup E_2$, weights $w_1 : E_1 \times V \rightarrow \mathbb{Q}_+$ and $\mu : E_2 \rightarrow \mathbb{Q}_+$, a subset A of V , and $b : A \rightarrow \mathbb{Q}_+$. We denote by $P_T(\mathcal{I})$ the polytope that consists of vectors $x \in \mathbb{Q}^E$ and $y \in \mathbb{Q}^{E_2 \times V}$ that satisfy

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

$$0 \leq y(e, u), y(e, v) \quad \text{for all } e = uv \in E_2, \quad (2)$$

$$y(e, u) + y(e, v) = x(e) \quad \text{for all } e = uv \in E_2, \quad (3)$$

$$x(E) = |V| - 1, \quad (4)$$

$$x(E(U)) \leq |U| - 1 \quad \text{for all } U \subset V \text{ with } 2 \leq |U|, \quad (5)$$

and

$$\sum_{e \in \delta(v; E_1)} w_1(e, v)x(e) + \sum_{e \in \delta(v; E_2)} \mu(e)y(e, v) \leq b(v) \quad \text{for all } v \in A, \quad (6)$$

where $x(F)$ denotes $\sum_{e \in F} x(e)$ for $F \subseteq E$. Variable $x(e)$ decides whether edge e is chosen, and variable $y(e, v)$ decides how much of $\mu(e)$ is allocated to an end vertex v of e . Note that (5) with $U = \{u, v\}$, $uv \in E$ implies

$$x(e) \leq 1 \quad \text{for all } e \in E. \quad (7)$$

Also constraints (4) and (5) with $U = V - v$ imply

$$x(\delta(v; E)) \geq 1 \quad \text{for all } v \in V, \quad (8)$$

since $x(\delta(v; E)) = x(E) - x(E(V - v)) \geq (|V| - 1) - (|V - v| - 1) = 1$.

Observe that $\min\{c^T x \mid (x, y) \in P_T(\mathcal{I})\}$ with $A = V$ is a linear program relaxation of WDBOUNDEDTREE. Although (5) has an exponential number of constraints, the linear program is solvable in polynomial time by using the ellipsoid method [4] or by transforming it to an equivalent formulation of polynomial-size [18].

For a vector $x \in \mathbb{Q}_+^E$, let E_x denote $\{e \in E \mid x(e) > 0\}$. We say that polytope $P_T(\mathcal{I})$ is $(1, \beta)$ -bounded for some $\beta \geq 1$ if every extreme point (x^*, y^*) of the polytope satisfies at least one of the following:

- There exists a vertex $v \in V$ such that $|\delta(v; E_{x^*})| = 1$;
- There exists a vertex $v \in A$ such that $|\delta(v; E_{x^*})| \leq \beta$.

If $|\delta(v; E_{x^*})| = 1$, then $x^*(e) = 1$ holds for the edge $e \in \delta(v; E_{x^*})$ by the equalities $x(\delta(v; E_{x^*})) = x(\delta(v; E)) \geq 1$ and $x(e) \leq 1$.

In what follows, we see that the iterative rounding can be applied to WDBOUNDEDTREE when $P_T(\mathcal{I})$ is $(1, \beta)$ -bounded. By this and the fact that $P_T(\mathcal{I})$ is $(1, 3)$ -bounded (Theorem 5), we can obtain an approximation algorithm for WDBOUNDEDTREE.

Now let us describe the algorithm which works under the assumption that $P_T(\mathcal{I})$ is $(1, \beta)$ -bounded. Roughly illustrating, if the former condition of the $(1, \beta)$ -boundedness holds, then the algorithm rounds the variable corresponding to $e \in \delta(v; E_{x^*})$, and otherwise, the algorithm removes the upper-bound on the weighted degree of v satisfying $|\delta(v; E_{x^*})| \leq \beta$.

Algorithm for WDBOUNDEDTREE

Input: An undirected graph $G = (V, E)$ with a partition E_1, E_2 of E , weights $w_1 : E_1 \times V \rightarrow \mathbb{Q}_+$ and $\mu : E_2 \rightarrow \mathbb{Q}_+$, an edge-cost $c : E \rightarrow \mathbb{Q}$, and a degree-bound $b : V \rightarrow \mathbb{Q}_+$.

Output: A solution consisting of a spanning tree $T \subseteq E$ of G and an allocation $w_2 : T \times V \rightarrow \mathbb{Q}_+$ of μ , or message “INFEASIBLE.”

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

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

If $P_T(\mathcal{I}) = \emptyset$, then output “INFEASIBLE,” and terminate.

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

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

Step 4: If there exists a vertex $v \in V$ such that $|\delta(v; E_{x^*})| = 1$ (i.e., the edge $e = uv \in \delta(v; E_{x^*})$ satisfies $x^*(e) = 1$), then add the edge e to T and delete v from G . Moreover, execute one of the following operations:

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

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

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

Step 6: If $|V| = 1$, then output (T, w_2) as a solution, and terminate. Otherwise, return to Step 2.

In Step 4, we need to update the function f . In actual, we can avoid updating by solving the linear program with $f(U) - |\delta(U; F)|$ instead of $f(U)$.

Theorem 3. *If each polytope $P_T(\mathcal{I})$ constructed in Step 2 of the algorithm is $(1, \beta)$ -bounded, then WDBOUNDEDTREE is $(1, 1 + \beta(1 + \theta))$ -approximable in polynomial time.*

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

Observe that the linear program over $P_T(\mathcal{I})$ 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 $\mu(e) \leq b(u) + b(v) \leq (1 + \theta)b(v)$.

Now suppose that $P_T(\mathcal{I}) \neq \emptyset$ after Step 1. We then prove that $P_T(\mathcal{I}) \neq \emptyset$ also throughout the subsequent iterations and that the spanning tree T output by the algorithm satisfies $c(T) \leq \min\{c^T x \mid (x, y) \in P_T(\mathcal{I})\}$ and $d_w(v; T) \leq (1 + \beta(1 + \theta))b(v)$ for all $v \in V$.

Let $e_i = u_i v_i$ denote the i -th edge added to T , $\mathcal{I}_i = (G_i = (V_i, E^i), w_1, \mu, A_i, b_i)$ denote \mathcal{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 \mathcal{I}_0 stand for \mathcal{I} immediately after Step 1 of the algorithm. Assume that e_i is chosen by $|\delta(v_i; E_{x_i^*})| = 1$ in Step 4 (i.e., $V_{i+1} - V_i = \{v_i\}$).

By Steps 4 and 5, $A_{i+1} \subseteq A_i$ holds, and

$$b_{i+1}(v) = \begin{cases} b_i(v) - w_1(e_i, v) & \text{if } v = u_i \in A \text{ and } e_i \in E_1, \\ b_i(v) - \mu(e_i)y_i^*(e_i, v) & \text{if } v = u_i \in A \text{ and } e_i \in E_2, \\ b_i(v) & \text{otherwise} \end{cases} \quad (9)$$

also holds for $i \geq 1$. Moreover, each edge in $E_i - (E_{i+1} \cup \{e_i\})$ is the one such that the corresponding variable of x^* becomes 0 in some iteration before e_{i+1} is chosen in Step 4. These facts indicate that the projection of (x_i^*, y_i^*) satisfies all constraints in $P_T(\mathcal{I}_{i+1})$. Hence we have the following:

$$\text{If } P_T(\mathcal{I}_i) \neq \emptyset, \text{ then } P_T(\mathcal{I}_{i+1}) \neq \emptyset \text{ for } i \geq 0; \quad (10)$$

$$c^T x_i^* \geq c(e_i) + \min\{c^T x \mid (x, y) \in P_T(\mathcal{I}_{i+1})\} = c(e_i) + c^T x_{i+1}^* \text{ for } i \geq 1. \quad (11)$$

(i) We first see that the algorithm outputs a solution. Recall that we are assuming that $P_T(\mathcal{I}_0) \neq \emptyset$. By this and (10), $P_T(\mathcal{I}_i) \neq \emptyset$ for all $i \geq 1$. The algorithm then terminates with outputting a spanning tree $T = \{e_1, \dots, e_{|V|-1}\}$ and an allocation $w_2 : T_2 \times V \rightarrow \mathbb{Q}_+$ of μ by the way of the construction.

(ii) Next we see the optimality of $c(T)$. By applying (11) repeatedly, we obtain

$$c^T x_1^* \geq c(e_1) + c^T x_2^* \geq \dots \geq \sum_{i=1}^{|V|-2} c(e_i) + c^T x_{|V|-1}^*.$$

Since $|V_{|V|-1}| = 2$, we have $x_{|V|-1}^*(e_{|V|-1}) = 1$ and $x_{|V|-1}^*(e) = 0$ for $e \in E_{|V|-1} - \{e_{|V|-1}\}$, and hence it holds that

$$\sum_{i=1}^{|V|-2} c(e_i) + c^T x_{|V|-1}^* = \sum_{i=1}^{|V|-1} c(e_i) = c(T).$$

Notice that the algorithm constructs \mathcal{I}_1 from \mathcal{I}_0 by relaxing the degree constraints (i.e., $A_1 \subseteq A_0$). Hence $\min\{c^T x \mid (x, y) \in P_T(\mathcal{I}_0)\} \geq c^T x_1^*$ holds. We thus have $\min\{c^T x \mid (x, y) \in P_T(\mathcal{I}_0)\} \geq c(T)$, as required.

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

Consider Step 4 of the iterations during $v \in A$. Let T' be the set of edges that are added to T during those iterations. By applying (9) repeatedly, we obtain

$$b(v) \geq \sum_{e_i \in \delta(v; T'_1)} w_1(e_i, v) + \sum_{e_i \in \delta(v; T'_2)} \mu(e_i) y_i^*(e_i, v).$$

If $e_i \in \delta(v; T_2)$, then the algorithm sets $w_2(e_i, v)$ to $\mu(e_i) y_i^*(e_i, v)$.

Therefore,

$$\sum_{e_i \in \delta(v; T'_1)} w_1(e_i, v) + \sum_{e_i \in \delta(v; T'_2)} \mu(e_i) y_i^*(e_i, v) = d_{w_1}(v; T'_1) + d_{w_2}(v; T'_2).$$

This implies that $d_w(v; T') \leq b(v)$ holds.

Consider the iterations after v is removed from A . Let T'' denote the set of edges that are added to T during those iterations. When v is removed from A in Step 5, the number of remaining edges incident to v is at most β by the condition in Step 5. Hence $|\delta(v; T'')| \leq \beta$ holds. We have already seen that, after Step 1, each $e = uv \in E_1$ satisfies $w_1(e, v) \leq b(v)$ and each $e = uv \in E_2$ satisfies $w_2(e, v) \leq \mu(e) \leq (1 + \theta)b(v)$. So $d_w(v; T'') \leq \beta(1 + \theta)b(v)$. Because $d_w(v; T) = d_w(v; T') + d_w(v; T'')$, we have $d_w(v; T) \leq (1 + \beta(1 + \theta))b(v)$.

The following theorem shows that the algorithm to WDBOUNDEDTREE gives an algorithm to MINIMUMWDTREE.

Theorem 4. *Suppose that WDBOUNDEDTREE is (α', β') -approximable for some α' and β' when b is uniform. For an arbitrary $\epsilon > 0$, MINIMUMWDTREE is $(\beta' + \epsilon)$ -approximable in polynomial time of $\log(1/\epsilon)$ and input size. If $E_2 = \emptyset$, then it is β' -approximable in polynomial time of only 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 $b_r : V \rightarrow \mathbb{Q}_+$ be the function such that $b_r(v) = r$ for all $v \in V$, and $\mathcal{I}_r = (G_r, w_1, \mu, A = V, b_r)$.

We denote $\min\{r \in \mathbb{Q}_+ \mid P_T(\mathcal{I}_r) \neq \emptyset\}$ by R , and the minimum weighted degree of the given instance by OPT . Let ω (resp., W) stand for the minimum (resp., maximum) of all entries in w_1 and μ . For a given ϵ , define $\epsilon' = \epsilon\omega/(2\beta')$. Since $\omega/2 \leq \text{OPT}$, we have $\epsilon' \leq \epsilon\text{OPT}/\beta'$. Since the characteristic vector of an optimal solution to the given instance of MINIMUMWDTREE satisfies all constraints of $P_T(\mathcal{I}_{\text{OPT}})$, we have $R \leq \text{OPT}$. 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 $P_T(\mathcal{I}_r)$ $\log(W/\epsilon')$ times.

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

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 . In this case, if R' satisfies $R \leq R' \leq R + \epsilon'$, then $R' \leq \text{OPT}$. Such R' can be computed by solving the linear program $\log(W/\epsilon') \leq \log(W(\psi + 1))$ times. Then we have $d_w(v; T) \leq \beta' b_{R'}(v) \leq \beta'\text{OPT}$ for any $v \in V$, which implies that T is a β' -approximate solution. \square

Now we see that $P_T(\mathcal{I})$ is $(1, 3)$ -bounded. First let us observe that the key property of tight constraints observed in [1] holds also in our setting.

Lemma 1. *For any extreme point (x^*, y^*) of $P_T(\mathcal{I})$, there exists a laminar family $\mathcal{L} \subseteq \{U \subseteq V \mid |U| \geq 2\}$ (i.e., any $U, U' \in \mathcal{L}$ satisfy either $U \subseteq U'$, $U' \subseteq U$, or $U \cap U' = \emptyset$) and a subset X of A such that $|E_{x^*}| \leq |\mathcal{L}| + |X|$.*

Proof: By the definitions of x^* and y^* , the number of variables is equal to the dimension of the vector space spanned by the coefficient vectors of tight constraints in $P_T(\mathcal{I})$. If $x^*(e) = 0$ (resp., $y^*(e, v) = 0$), then fix the variable $x(e)$ (resp., $y(e, v)$) to 0 and remove the corresponding tight constraint of (1) (resp., (2)). We can also remove tight constraints of (3) by fixing $y(e, u)$ to $x(e) - y(e, v)$. 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 (4), (5) and (6).

Let $\mathcal{F} = \{U \subseteq V \mid |U| \geq 2, x^*(E(U)) = |U| - 1\}$ (i.e., family of vertex subsets defining tight constraints of (4) and (5)) 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 (6)).

For a subfamily \mathcal{F}' of \mathcal{F} , we denote by $\text{span}(\mathcal{F}' \cup X)$ the vector space spanned by the coefficient vectors of constraints corresponding to \mathcal{F}' and X . (Notice that coefficient vectors corresponding to X are changed from the original by the operations described at the beginning of this proof.) In [1], it is proven that a maximal laminar subfamily \mathcal{L} of \mathcal{F} satisfies $\text{span}(\mathcal{L} \cup X) = \text{span}(\mathcal{F} \cup X)$. Since the dimension of $\text{span}(\mathcal{L} \cup X)$ is at most $|\mathcal{L}| + |X|$, we have $|E_{x^*}| \leq |\mathcal{L}| + |X|$, as required. \square

Theorem 5. *Polytope $P_T(\mathcal{I})$ is $(1, 3)$ -bounded for any \mathcal{I} .*

Proof: Suppose the contrary, i.e. all vertices $v \in V$ satisfy $|\delta(v; E_{x^*})| \geq 2$ and all vertices $v \in A$ satisfy $|\delta(v; E_{x^*})| \geq 4$. Then $|E_{x^*}| \geq (2(|V| - |A|) + 4|A|)/2 = |V| + |A|$.

On the other hand, let \mathcal{L} be an arbitrary laminar family of subsets U of V with $|U| \geq 2$, and X be an arbitrary subset of A . By their definitions, $|\mathcal{L}| \leq |V| - 1$ and $|X| \leq |A|$ hold. Therefore we have $|\mathcal{L}| + |X| \leq |V| + |A| - 1 < |E_{x^*}|$, a contradiction to Lemma 1. \square

Theorem 1 is proven immediately from Theorems 3, 4 and 5.

It is a natural question to ask whether the $(1, 3)$ -boundedness of $P_T(\mathcal{I})$ can be improved to $(1, 2)$ -boundedness. Let us discuss this assuming that $E_2 = \emptyset$. Unfortunately $(1, 2)$ -boundedness does not hold even if $w_1(e, u) = w_1(e, v) = 1$ for all $e = uv \in E_1$ as mentioned in [1]. Singh and Lau [1] weakened the $(1, 2)$ -boundedness by replacing its first condition with “There exists an edge

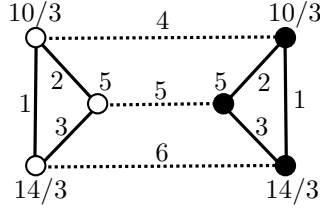


Figure 1: A counterexample for $(1, 2)$ -boundedness of $P_T(\mathcal{I})$

$e \in E$ such that $x^*(e) = 1$.” They then designed their algorithm by observing that the property holds for more general polytopes than $P_T(\mathcal{I})$. This approach does not work in our setting because there exists a counterexample for the weakened $(1, 2)$ -boundedness, which we will give in the rest of this section.

Let G be the graph in Figure 1. We let $w_1(e, u) = w_1(e, v)$ for all $e = uv \in E_1$ and the integers beside edges in the figure represent their weights. Rational numbers beside vertices represent the values of b for them. Let $A = V$, and the set of $|E| = 6$ tight constraints consist of constraints (4), (5) for the set of white vertices and for the set of black vertices, and (6) for all vertices. Then these tight constraints determine an extreme point x^* of $P_T(\mathcal{I})$ such that

$$x^*(e) = \begin{cases} 2/3 & \text{for edges represented by solid lines,} \\ 1/3 & \text{for edges represented by dotted lines.} \end{cases}$$

Clearly $x^*(e) < 1$ for any edge $e \in E$ and $\min_{v \in A=V} |\delta(v; E_{x^*})| = 3$.

3 Survivable Network with Weighted Degree Constraints

We prove Theorem 2 in this section. Similarly for the previous section, we introduce a general form of WDBOUNDEDNETWORK by defining $b : A \rightarrow \mathbb{Q}_+$ on a subset A of V . Moreover we let \mathcal{I} stand for an instance of the problem consisting of an undirected graph $G = (V, E)$ with a partition E_1, E_2 of E , weights $w_1 : E_1 \times V \rightarrow \mathbb{Q}_+$ and $\mu : E_2 \rightarrow \mathbb{Q}_+$, a skew supermodular set function $f : 2^V \rightarrow \mathbb{N}$, a subset A of V , and $b : A \rightarrow \mathbb{Q}_+$. We denote by $P_N(\mathcal{I})$ the polytope that consists of vectors $x \in \mathbb{Q}^E$ and $y \in \mathbb{Q}^{E_2 \times V}$ that satisfy

$$0 \leq x(e) \leq 1 \quad \text{for all } e \in E, \quad (12)$$

$$0 \leq y(e, u), y(e, v) \quad \text{for all } e = uv \in E_2, \quad (13)$$

$$y(e, u) + y(e, v) = x(e) \quad \text{for all } e = uv \in E_2, \quad (14)$$

$$x(\delta(U)) \geq f(U) \quad \text{for all non-empty } U \subset V, \quad (15)$$

and

$$\sum_{e \in \delta(v; E_1)} w_1(e, v)x(e) + \sum_{e \in \delta(v; E_2)} \mu(e)y(e, v) \leq b(v) \quad \text{for all } v \in A. \quad (16)$$

Observe that $\min\{c^T x \mid (x, y) \in P_N(\mathcal{I})\}$ with $A = V$ is a linear program relaxation of WDBOUNDEDNETWORK.

We say that $P_N(\mathcal{I})$ 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 $(1, \beta)$ -boundedness of $P_N(\mathcal{I})$ is weaker than that of $P_T(\mathcal{I})$.

Now we describe the algorithm which works under the assumption that $P_N(\mathcal{I})$ is (α, β) -bounded.

Algorithm for WDBOUNDEDNETWORK

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

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

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

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

If $P_N(\mathcal{I}) = \emptyset$, then output “INFEASIBLE.”

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

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 , and set $f(U) := f(U) - 1$ for all $U \subset V$ with $e \in \delta(U)$. Moreover, execute one of the following operations:

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 set $w_2(e, v) := \mu(e)y^*(e, v)/x^*(e)$. 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 all such v from A .

Step 6: If $E = \emptyset$, then output (F, w_2) as a solution, and terminate. Otherwise, return to Step 2.

Theorem 6. *If each polytope $P_N(\mathcal{I})$ constructed in Step 2 of the algorithm is (α, β) -bounded, then 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 show that the algorithm computes an $(\alpha, \alpha + \beta(1 + \theta))$ -approximate solution.

Observe that the linear program over $P_N(\mathcal{I})$ 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 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 $P_N(\mathcal{I}) \neq \emptyset$ after Step 1. We then prove that $P_N(\mathcal{I}) \neq \emptyset$ also throughout the subsequent iterations and that the edge set F output by the algorithm satisfies $c(F) \leq \alpha \min\{c^T x \mid (x, y) \in P_N(\mathcal{I})\}$ 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 . Moreover, let $\mathcal{I}_i = (G_i = (V, E^i), w_1, \mu, \nu, f_i, A_i, b_i)$ denote \mathcal{I} at the beginning of the iteration in which e_i is added to F , and (x_i^*, y_i^*) denote the basic solution computed in Step 2 of that iteration. We also let \mathcal{I}_0 stand for \mathcal{I} immediately after Step 1 of the algorithm, and assume that the algorithm outputs $F = \{e_1, \dots, e_j\}$. By Steps 4 and 5, $A_{i+1} \subseteq A_i$ holds, and

$$b_{i+1}(v') = \begin{cases} b_i(v') - w_1(e_i, v')x_i^*(e_i) & \text{if } v' \in A \text{ and } e_i \in E_1, \\ b_i(v') - \mu(e_i)y_i^*(e_i, v') & \text{if } v' \in A \text{ and } e_i \in E_2, \\ b_i(v') & \text{otherwise.} \end{cases} \quad (17)$$

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 corresponding variables of x^* took 0 in some iteration before e_{i+1} is chosen in Step 4. These facts indicate that the projection of (x_i^*, y_i^*) satisfies all constraints in $P_N(\mathcal{I}_{i+1})$. Hence we have the following:

$$\text{If } P_N(\mathcal{I}_i) \neq \emptyset, \text{ then } P_N(\mathcal{I}_{i+1}) \neq \emptyset \text{ for } i \geq 0; \quad (18)$$

$$c^T x_i^* \geq c(e_i)x_i^*(e_i) + \min\{c^T x \mid (x, y) \in P_N(\mathcal{I}_{i+1})\} \geq c(e_i)/\alpha + c^T x_{i+1}^* \text{ for } i \geq 1. \quad (19)$$

(i) We first see that the algorithm outputs a solution. Recall that we are assuming that $P_N(\mathcal{I}_0) \neq \emptyset$. By this and (18), $P_N(\mathcal{I}_i) \neq \emptyset$ holds for all $1 \leq i \leq j$. Hence the algorithm terminates with outputting a solution consisting of an f -connected subgraph $F = \{e_1, \dots, e_j\}$ and an allocation $w_2 : F_2 \times V \rightarrow \mathbb{Q}_+$ of μ by the way of construction.

(ii) Next we see the α -approximability of $c(F)$. By applying (19) repeatedly, we have

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

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 $\alpha c^T x_1^* \geq c(F)$. Notice that the algorithm constructs \mathcal{I}_1 from \mathcal{I}_0 by relaxing the degree constraints (i.e., $A_1 \subseteq A_0$). Hence $\min\{c^T x \mid (x, y) \in P_T(\mathcal{I}_0)\} \geq c^T x_1^*$. Therefore we have $\alpha \min\{c^T x \mid (x, y) \in P_N(\mathcal{I}_0)\} \geq c(F)$, 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 (17) 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,

$$\sum_{e_i \in \delta(v; F'_1)} w_1(e_i, v)x_i^*(e_i) + \sum_{e_i \in \delta(v; F'_2)} \mu(e_i)y_i^*(e_i, v) \geq d_{w_1}(v; F'_1)/\alpha + d_{w_2}(v; F'_2)/\alpha.$$

This 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 now have $d_w(v; F) \leq (\alpha + \beta(1 + \theta))b(v)$. \square

The following theorem shows that the algorithm for WDBOUNDEDNETWORK gives an algorithm for MINIMUMWDNETWORK.

Theorem 7. *Suppose that WDBOUNDEDNETWORK with uniform b is (α', β') -approximable for some α' and β' . For an arbitrary $\epsilon > 0$, MINIMUMWDNETWORK is $(\beta' + \epsilon)$ -approximable in polynomial time of $\log(1/\epsilon)$ and input size. If $E_2 = \emptyset$, then it is β' -approximable in polynomial time of only input size.*

Proof: It can be derived from Theorem 6 as Theorem 4 is derived from Theorem 3. \square

We can show that polytope $P_N(\mathcal{I})$ is $(2, 5)$ -bounded. For proving this, let us see that the key property of tight constraints observed in [3] also holds in our setting.

Lemma 2. *Let (x^*, y^*) be any extreme point of $P_N(\mathcal{I})$ 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 all $U \in \mathcal{L}$ 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 coefficient vectors of tight constraints in $P_N(\mathcal{I})$. If $x^*(e) = 0$ (resp., $y^*(e, v) = 0$), then fix the variable $x(e)$ (resp., $y(e, v)$) to 0 and remove the corresponding tight constraint of (12) (resp., (13)). We can also remove tight constraints of (14) by fixing $y(e, u)$ to $x(e) - y(e, v)$. 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 (15) and (16).

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 (15)) and $X = \{v \in A \mid \sum_{e \in \delta(v; E_1)} w_1(e, v)x^*(e) + \sum_{e \in \delta(v; E_2)} \mu(e)y^*(e, v) + \sum_{e \in \delta(v; E_3)} \nu(e)y^*(e, v) = b(v)\}$ (i.e., set of vertices defining tight constraints of (16)). For a subfamily \mathcal{F}' of \mathcal{F} , we denote by $\text{span}(\mathcal{F}')$ the vector space spanned by the characteristic vectors of $\delta(U; E_{x^*})$, $U \in \mathcal{F}'$. In [3], it is proven that a maximal laminar subfamily \mathcal{F}' of \mathcal{F} satisfies $\text{span}(\mathcal{F}') = \text{span}(\mathcal{F})$. From \mathcal{F}' , choose the maximal number of members whose coefficient vectors are linearly independent, and define \mathcal{L} as the family of them. Then \mathcal{L} and X satisfy the required properties. \square

Theorem 8. *Polytope $P_N(\mathcal{I})$ is $(2, 5)$ -bounded for any \mathcal{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 2. 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 [3], 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 [3]. \square

Theorem 2 can be proven immediately from Theorems 6, 7 and 8.

Lau et. al. [2] 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 $P_N(\mathcal{I})$ 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 2, $f(U) = 1$ for all non-empty $U \subset V$, and $A = V$. We suppose that $|E| = |E_1| = 42$ tight constraints consists of (15) 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 (16) for

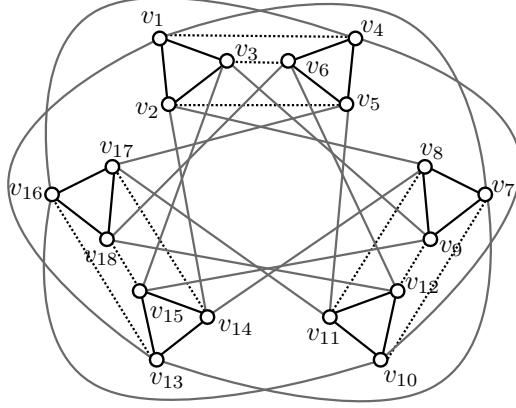


Figure 2: A counterexample for $(2, 4)$ -boundedness of $P_N(I)$

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 Conclusions

In this paper, we designed algorithms to network design problems related to the weighted degree constraints. Those algorithms are based on the iterative rounding, which has been applied for several network design problems with degree constraints. Our main contribution is to show that the iterative rounding is still applicable even after extending the degree constraints to the weighted degree constraints. For general formulation, we also introduced two definitions of edges weights, and showed that the iterative rounding works for them. The examples we obtained imply that the approximation guarantees of our algorithms are hard to improve.

References

- [1] M. Singh, L. C. Lau, Approximating minimum bounded degree spanning trees to within one of optimal, in: Proceedings of the 39th ACM Symposium on Theory of Computing (2007) 661–670.
- [2] L. C. Lau, J. S. Naor, M. Singh, M. R. Salavatipour, Survivable network design with degree or order constraints, SIAM Journal on Computing 39 (2009) 1062–1087.
- [3] K. Jain, A factor 2 approximation algorithm for the generalized Steiner network problem, Combinatorica 21 (2001) 39–60.
- [4] K. Chaudhuri, S. Rao, S. Riesenfeld, K. Talwar, What would Edmonds do? augmenting paths and witnesses for degree-bounded MSTs, Algorithmica 55 (2009) 157–189.
- [5] K. Chaudhuri, S. Rao, S. Riesenfeld, K. Talwar, A push-relabel algorithm for approximating degree bounded MSTs, Theoretical Computer Science 410 (2009) 4489–4503.

- [6] J. Könemann, R. Ravi, A matter of degree: Improved approximation algorithms for degree-bounded minimum spanning trees, *SIAM Journal on Computing* 31 (2002) 1783–1793.
- [7] J. Könemann, R. Ravi, Primal-dual meets local search: approximating MST’s with nonuniform degree bounds, *SIAM Journal on Computing* 34 (2005) 763–773.
- [8] R. Ravi, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz, H. B. Hunt III, Many birds with one stone: Multi-objective approximation algorithms, in: *Proceedings of the 25th ACM Symposium on Theory of Computing* (1993) 438–447.
- [9] R. Ravi, M. Singh, Delegate and conquer: An LP-based approximation algorithm for minimum degree MSTs, in: *Proceedings of 33rd International Colloquium on Automata, Languages and Programming, Lecture Notes in Computer Science* 4051 (2006) 169–180.
- [10] M. Fürer, B. Raghavachari, Approximating the minimum-degree Steiner tree to within one of optimal, *Journal of Algorithms* 17 (1994) 409–423.
- [11] M. X. Goemans, Minimum bounded-degree spanning trees, in: *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science* (2006) 273–282.
- [12] L. C. Lau, M. Singh, Additive approximation for bounded degree survivable network design, in: *Proceedings of the 40th ACM Symposium on Theory of Computing* (2008) 759–768.
- [13] N. Bansal, R. Khandekar, V. Nagarajan, Additive guarantees for degree-bounded directed network design, *SIAM Journal on Computing* 39 (2009) 1413–1431.
- [14] T. Kiraly, L. C. Lau, M. Singh, Degree bounded matroids and submodular flows, in: *Proceedings of 13th Conference on Integer Programming and Combinatorial Optimization, Lecture Notes in Computer Science* 5035 (2008) 259–272.
- [15] R. Ravi, Steiner trees and beyond: Approximation algorithms for network design, Ph.D. thesis, Department of Computer Science, Brown University (1993).
- [16] M. Ghodsi, H. Mahini, K. Mirjalali, S. O. Gharan, A. S. Sayedi R., M. Zadimoghaddam, Spanning trees with minimum weighted degrees, *Information Processing Letters* 104 (2007) 113–116.
- [17] Z. Nutov, Approximating directed weighted-degree constrained networks, in: *Proceedings of 11th International Workshop, APPROX 2008, and 12th International Workshop, RANDOM 2008, Lecture Notes in Computer Science* 5171 (2008) 219–232.
- [18] M. Grötschel, L. Lovász, A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag (1988).