Network Design with Weighted Degree Constraints*

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

Abstract

In an undirected graph $G = (V, E)$ with a weight function $w : E \times V \to \mathbb{Q}_+$, the weighted degree $d_w(v; E)$ of a vertex v is defined as $\sum\{w(e, v) \mid e \in E \text{ incident with } v\}$. In this paper, we consider a network design problem with upper-bound on weighted degree of each vertex. Inputs of the problem are an undirected graph $G = (V, E)$ with $E = E_1 \dot\cup E_2 \dot\cup E_3$, weights $w_1 : E_1 \times V \to \mathbb{Q}_+$, $\mu : E_2 \to \mathbb{Q}_+$, $\nu : E_3 \to \mathbb{Q}_+$, an edge-cost $c : E \to \mathbb{Q}$, a skew supermodular set function $f : 2^V \to \mathbb{N}$, and a degree-bound $b : A \to \mathbb{Q}_+$. A solution consists of $F \subseteq E$, and weights $w_i : F_i \times V \to \mathbb{Q}_+$ for $i \in \{2, 3\}$, where F_i stands for $F \cap E_i$. It is defined to be feasible if the cut-size of U in (V, F) is at least $f(U)$ for $U \subset V$, $w_2(e, u) + w_2(e, v) = \mu(e)$ for $e = uv \in F_2$, $\{w_3(e, u), w_3(e, v)\} = \{0, \nu(e)\}$ for $e = uv \in F_3$, and $d_{w_1}(v; F_1) + d_{w_2}(v; F_2) + d_{w_3}(v; F_3) \le b(v)$ for each $v \in V$. The goal of this problem is to find a feasible solution that minimizes its cost $\sum_{e \in F} c(e)$.

Relaxing the constraints on weighted degree, we propose a bi-criteria approximation algorithm based on the iterative rounding, which has been successfully applied to the degree-bounded spanning tree problem. Our algorithm computes a $(2, 9 + 5\theta)$-approximate solution, where $\theta = \max\{b(u)/b(v), b(v)/b(u) \mid uv \in E_2\}$ if $E_2 \ne \emptyset$ and $\theta = 0$ if $E_2 = \emptyset$, where (α, β)-approximate solution has the cost at most α times the optimal and the weighted degree of v at most $\beta b(v)$. We also give a $(1, 5 + 3\theta)$-approximation algorithm to the case of $f(U) = 1$ for $U \subset V$. Moreover, a problem minimizing the maximum weighted degree of vertices is also discussed.

1 Introduction

Let $G = (V, E)$ be an undirected graph. A weight function $w : E \times V \to \mathbb{Q}_+$ 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 with $v \in V$. 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 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 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

*Technical report #2008-005, April 24, 2008.

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 three types of edges. For an edge $e = uv$ of the first type, weights $w(e, u)$ and $w(e, v)$ are given as inputs. For an edge $e = uv$ of the second type, weight $\mu(e)$ is given and we can allocate it to u and v. In other words, we decide $w(e, u)$ and $w(e, v)$ so that $w(e, u) + w(e, v) = \mu(e)$. For an edge of the third type, weight $\nu(e)$ is given and we can decide $w(e, u)$ and $w(e, v)$ so that $w(e, u) + w(e, v) = \nu(e)$ similarly for the second type while $\{w(e, u), w(e, v)\} = \{0, \nu(e)\}$ must hold for the third type.

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 $f : 2^V \to \mathbb{N}$ 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 such as the local edge-connectivity.

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, E_2, and E_3, and possibly contains parallel edges. For those sets, weights $w_1 : E_1 \times V \to \mathbb{Q}_+$, $\mu : E_2 \to \mathbb{Q}_+$ and $\nu : E_3 \to \mathbb{Q}_+$ are respectively defined. As inputs, we are given the graph $G = (V, E = E_1 \cup E_2 \cup E_3)$ with the weights w_1, μ and ν, 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 $w_i(e, u) \in \mathbb{Q}_+$ and $w_i(e, v) \in \mathbb{Q}_+$ for each $e = uv \in F_i$, $i \in \{2, 3\}$, where F_i denotes $F \cap E_i$. We call w_2 (resp., w_3) *allocation* of μ (resp., ν) when $w_2(e, u) + w_2(e, v) = \mu(e)$ for $e = uv \in F_2$ (resp., $\{w_3(e, u), w_3(e, v)\} = \{0, \nu(e)\}$ for $e = uv \in F_3$). 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 and w_3 are allocations of μ and ν respectively, and degree constraint $d_w(v; F) \leq b(v)$ for each $v \in V$ is satisfied. 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 are spanning trees. We particularly call such instances *weighted degree bounded spanning tree problem* (WDBoundedTree).

Feasible solutions of WDBoundedTree are Hamiltonian paths when $E_2 = E_3 = \emptyset$, $w_1(e, u) = w_1(e, v) = 1$ 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) is feasible 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 $F \subseteq E$, an allocation w_2 of μ, and an allocation w_3 of ν 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. Let κ be 1 if $E_3 \neq \emptyset$, and 0 otherwise. For problems WDBoundedTree and WDBoundedNetwork, we propose algorithms which achieve approximation ratios $(1, 4+3\theta+\kappa)$ and $(2, 7+5\theta+2\kappa)$ respectively in $O(L(|V|+|E|))$ time, where L is the time for solving a linear programming. Our algorithms take the approach successfully applied to the bounded degree spanning tree problem by Singh and Lau [17] and to the bounded-degree survivable network design problem by Lau et al. [12], which correspond to instances with uniform w_1 and $E_2 = E_3 = \emptyset$ in our problems. Their approach is based on the iterative rounding originally used for the generalized Steiner network problem by Jain [8]. Roughly illustrating, they iterate rounding fractional variables in basic optimal solutions or removing constraints of a linear programming 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 even if the weighted degree is introduced.

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 $E = E_1 \cup E_2 \cup E_3$, weights $w_1 : E_1 \times V \to \mathbb{Q}_+$, $\mu : E_2 \to \mathbb{Q}_+$, $\nu : E_3 \to \mathbb{Q}_+$, and a skew supermodular set function $f : 2^V \to \mathbb{N}$ are given. A feasible solution consists of a f-connected subgraph $G' = (V, F)$ of G, an allocation $w_2 : F_2 \times V \to \mathbb{Q}_+$ of μ, and an allocation $w_3 : F_3 \times V \to \mathbb{Q}_+$ of ν. The objective is to minimize the weighted degree $\max_{v \in V} d_w(v; F)$ of G'.

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

For problems MinimumWDTree and MinimumWDNetwork, our algorithms achieve approximation ratios $4+\kappa$ and $7+2\kappa$ in $O(L(|E|+|V|+\log(W/\psi))$ time if $E_2 = \emptyset$, where $W = \sum_{e=uv \in E_1}(w_1(e,u) + w_1(e,v)) + \sum_{e \in E_2} \mu(e) + \sum_{e \in E_3} \nu(e)$, and ψ denotes the maximum denominator of all given weights w_1, μ and ν. If $E_2 \neq \emptyset$, our algorithms achieve approximation ratios $7+\kappa+\epsilon$ and $12+2\kappa+\epsilon$ in $O(L(|E|+|V|+\log(W/(\omega\epsilon)))$ time for an arbitrary $\epsilon > 0$, where ω denotes the minimum of all given weights w_1, μ and ν.

Previous Works

The bounded degree spanning tree problem has been studied extensively in the last two decades [2, 3, 10, 11, 15, 16]. For the uniform cost (i.e., $c(e) = 1$ for $e \in E$), an optimal result was given by Fürer and Raghavachari [4]. Their algorithm computes a spanning tree which violates degree upper-bounds by at most one. For general costs, Goemans [6] 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 [17]; 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 [8], who applied it for designing a 2-approximation algorithm to the generalized Steiner network problem.

After their algorithm, this approach is applied to several problems with degree bounds. Lau et al. [12] 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 [13]. Bansal et al. [1] considered the arborescence problem and survivable network problem with intersecting supermodular connectivity. Kiraly et al. [9] generalized bounded degree spanning tree to bounded degree matroid. They also considered degree bounded submodular flow problem.

There also are several works on the network design problem with weighted degree constraints. All of these correspond to the case with $E_2 = E_3 = \emptyset$ and $w_1(e,u) = w_2 = (e,v)$ for $e = uv \in E_1$. Ravi [14] presented an $O(\log|V|, \log|V|)$-approximation algorithm to problem WDBoundedTree and an $O(\log|V|)$-approximation algorithm to problem MinimumWDTree. For problem MinimumWDTree, Ghodsi et al. [5] 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 problem WDBoundedTree and 4-approximation to problem MinimumWDTree when $E_2 = E_3 = \emptyset$. Hence it improves these previous works.

Organization

The rest of this paper is organized as follows. Section 2 presents our algorithms to problems WDBoundedTree and MinimumWDTree. The algorithms are derived from a good property of polytopes that give a linear programming relaxation of the problems. Section 2 also shows that our analysis on the property is tight. Section 3 gives our algorithms to problems WDBoundedNetwork and MinimumWDNetwork, and shows that our analysis on the property of polytopes is tight. Section 4 concludes this paper with some remarks.

2 Spanning Trees with Weighted Degree Constraints

In this section, we let I stand for the set of an undirected graph $G = (V,E)$ with $E = E_1 \cup E_2 \cup E_3$, weights $w_1 : E_1 \times V \to \mathbb{Q}_+$, $\mu : E_2 \to \mathbb{Q}_+$ $\nu : E_3 \to \mathbb{Q}_+$, a subset A of V, and $b : A \to \mathbb{Q}_+$. Note that A is a set of vertices whose weighted degrees are bounded by b. We denote by $\mathrm{P_T}(I)$ the polytope that consists of vectors $x \in \mathbb{Q}^E$ and $y \in \mathbb{Q}^{(E_2 \cup E_3)\times V}$ that satisfy

$$\begin{aligned}
0 &\le x(e) && \text{for all } e \in E, && (1)\\
0 &\le y(e,u),\ y(e,v) && \text{for all } e = uv \in E_2 \cup E_3, && (2)\\
y(e,u) + y(e,v) &= x(e) && \text{for all } e = uv \in E_2 \cup E_3, && (3)\\
x(E) &= |V| - 1, && && (4)\\
x(E(U)) &\le |U| - 1 && \text{for all } U \subset V \text{ with } 2 \le |U|, && (5)
\end{aligned}$$

and

$$\sum_{e \in \delta(v;E_1)} w_1(e,v)x(e) + \sum_{e \in \delta(v;E_2)} \mu(e)y(e,v) + \sum_{e \in \delta(v;E_3)} \nu(e)y(e,v) \le b(v) \quad \text{for all } v \in A, \qquad (6)$$

where $x(F)$ denotes $\sum_{e \in F} x(e)$ for $F \subseteq E$. Remark that (5) with $U = \{u,v\}$, $uv \in E$ implies

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

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

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

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

Observe that $\mathrm{P_T}(I)$ with $A = V$ is the polytope of a linear programming relaxation of problem WDBoundedTree. Although (5) has an exponentially many number of constraints, linear programming over the polytope is solvable in polynomial time by using the ellipsoid method [2] or by transforming it to a polynomial-size formulation [7].

For a vector $x \in \mathbb{Q}_+^E$, let E_x denote $\{e \in E \mid x(e) > 0\}$. We say that polytope $\mathrm{P_T}(I)$ 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 problem WDBoundedTree when $\mathrm{P_T}(I)$ is $(1,\beta)$-bounded. By this and the fact that $\mathrm{P_T}(I)$ is $(1,3)$-bounded (Theorem 3), we can obtain an approximation algorithm for problem WDBoundedTree.

Now let us describe the algorithm which works under the assumption that $\mathrm{P_T}(I)$ is $(1,\beta)$-bounded.

Algorithm for problem WDBoundedTree

Input: An undirected graph $G = (V,E)$ with $E = E_1 \cup E_2 \cup E_3$, weights $w_1 : E_1 \times V \to \mathbb{Q}_+$, $\mu : E_2 \to \mathbb{Q}_+$, $\nu : E_3 \to \mathbb{Q}_+$, an edge-cost $c : E \to Q$, and a degree-bound $b : V \to \mathbb{Q}_+$.

Output: A solution consisting of a spanning tree $T \subseteq E$ of G, an allocation $w_2 : T_2 \times V \to \mathbb{Q}_+$ of μ and an allocation $w_3 : T_3 \times V \to \mathbb{Q}_+$ of ν, or message "INFEASIBLE".

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

- 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 $\mu(e) > b(u) + b(v)$.
- Delete $e = uv \in E_3$ from G if $\nu(e) > \max\{b(u), b(v)\}$.
- If $e = uv \in E_3$ and $b(u) \geq \nu(e) > b(v)$, then move e from E_3 to E_1 with setting $w_1(e,u) := \nu(e)$ and $w_1(e,v) := 0$. If $e \in E_3$ and $b(v) \geq \nu(e) > b(u)$, then move e from E_3 to E_1 with setting $w_1(e,u) := 0$ and $w_1(e,v) := \nu(e)$.

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

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

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 e to T and delete v from G. 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 $b(u) := b(u) - w_1(e,u)$;

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 $b(u) := b(u) - w_2(e,u)$.

Case of $e \in E_3$**:** If $y^*(e,u) \geq y^*(e,v)$, then set $w_3(e,u) := \nu(e)$ and $w_3(e,v) := 0$. If $y^*(e,u) < y^*(e,v)$, then set $w_3(e,u) := 0$ and $w_3(e,v) := \nu(e)$. If $u \in A$, then set $b(u) := b(u) - \nu(e)y^*(e,u)$.

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 $|V| = 1$, then output (T, w_2, w_3) as a solution, and terminate. Otherwise, return to Step 2.

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. Moreover, define $\kappa = 1$ if $E_3 \neq \emptyset$, and $\kappa = 0$ otherwise. We let L denote the time for solving the linear programming over $\mathrm{P_T}(I)$.

Theorem 1. *If each* $\mathrm{P_T}(I)$ *constructed in Step 2 is* $(1,\beta)$*-bounded, then problem* WDBoundedTree *is* $(1, 1+\beta(1+\theta)+\kappa)$*-approximable in* $O(L(|V|+|E|))$ *time.*

Proof. It is clear that the algorithm described above runs in $O(L(|V|+|E|))$ time. In what follows, we see that the algorithm computes a $(1, 1+\beta(1+\theta)+\kappa)$-approximate solution.

Observe that the linear programming over $\mathrm{P_T}(I)$ 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 $\mu(e) \leq b(u)+b(v) \leq (1+\theta)b(v)$;
- If $e = uv \in E_3$, then $\nu(e) \leq b(u)$ and $\nu(e) \leq b(v)$.

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

Let $e_i = u_i v_i$ denote the i-th edge added to T, $I_i = (G_i = (V_i, E^i), w_1, \mu, \nu, A_i, b_i)$ 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. 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

$$b_{i+1}(v) = \begin{cases} b_i(v) - w_1(e_i, v) & \text{if } v = u_i \in A \text{ and } e_i \in E_1, \\ b_i(v) - \mu(e_i)y_i^*(e_i, v) & \text{if } v = u_i \in A \text{ and } e_i \in E_2, \\ b_i(v) - \nu(e_i)y_i^*(e_i,) & \text{if } v = u_i \in A \text{ and } e_i \in E_3, \\ b_i(v) & \text{otherwise.} \end{cases} \quad (9)$$

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 $\mathrm{P_T}(I_{i+1})$. Hence we have the following:

$$\text{If } \mathrm{P_T}(I_i) \neq \emptyset, \text{ then } \mathrm{P_T}(I_{i+1}) \neq \emptyset \text{ for } i \geq 0; \quad (10)$$

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

(i) We first see that the algorithm outputs a solution. Recall that we are assuming that $\mathrm{P_T}(I_0) \neq \emptyset$. By this and (10), $\mathrm{P_T}(I_i) \neq \emptyset$ for all $i \geq 1$. The algorithm then terminates with outputting a spanning tree $T = \{e_1, \ldots, e_{|V|-1}\}$, an allocation $w_2 : T_2 \times V \to \mathbb{Q}_+$ of μ, and an allocation $w_3 : T_3 \times V \to \mathbb{Q}_+$ of ν, by the way of the construction.

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

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

Since $|V_{|V|-1}| = 2$, $x_{|V|-1}^*(e_{|V|-1}) = 1$ and $x_{|V|-1}^*(e) = 0$ for $e \in E_{|V|-1} - \{e_{|V|-1}\}$ obviously hold, and hence $\sum_{i=1}^{|V|-2} c(e_i) + c^T x_{|V|-1}^* = \sum_{i=1}^{|V|-1} c(e_i) = c(T)$, where T denotes it outputted by the algorithm. Notice that the algorithm constructs I_1 from I_0 by relaxing the degree constraints (i.e., $A_1 \subseteq A_0$). Hence $\min\{c^T x \mid (x, y) \in \mathrm{P_T}(I_0)\} \geq c^T x_1^*$ holds. Hence we have $\min\{c^T x \mid (x, y) \in \mathrm{P_T}(I_0)\} \geq c(T)$, as required.

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

$$b(v) \geq \sum_{e_i \in \delta(v; T_1')} w_1(e_i, v) + \sum_{e_i \in \delta(v; T_2')} \mu(e_i) y_i^*(e_i, v) + \sum_{e_i \in \delta(v; T_3')} \nu(e_i) y_i^*(e_i, v).$$

If $e_i \in \delta(v; E_2)$, then $w_2(e_i, v) = \mu(e_i) y_i^*(e_i, v)$. If $e_i \in \delta(v; E_3)$, then $\nu(e_i) y_i^*(e_i, v) \geq w_3(e_i, v)/2$ holds because even in the case of $w_3(e_i, v) = \nu(e_i)$, $y_i^*(e_i, v) \geq y_i^*(e_i, u)$ holds, and hence $y_i^*(e_i, v) \geq (y_i^*(e_i, u) + y_i^*(e_i, v))/2 = x_i^*(e_i)/2 = 1/2$. Therefore,

$$\sum_{e_i \in \delta(v; T_1')} w_1(e_i, v) + \sum_{e_i \in \delta(v; T_2')} \mu(e_i) y_i^*(e_i, v) + \sum_{e_i \in \delta(v; T_3')} \nu(e_i) y_i^*(e_i, v)$$
$$\geq d_{w_1}(v; T_1') + d_{w_2}(v; T_2') + d_{w_3}(v; T_3')/2.$$

It implies that $d_w(v; T') \leq b(v)$ holds if $E_3 = \emptyset$, and $d_w(v; T') \leq 2b(v)$ otherwise.

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 with v is at most β by the condition in Step 5. Hence $|\delta(v; T_a)| \leq \beta$ holds. We have already seen that, after Step 1, $e = uv \in E_1$ satisfies $w_1(e, v) \leq b(v)$, $e = uv \in E_2$ satisfies $w_2(e, v) \leq \mu(e) \leq (1 + \theta)b(v)$, and $e = uv \in E_3$ satisfies $w_3(e, v) \leq \nu(e) \leq b(v)$. 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 $d_w(v; T) \leq (1 + \beta(1 + \theta))b(v)$ if $E_3 = \emptyset$, and $d_w(v; T) \leq (2 + \beta(1 + \theta))b(v)$ otherwise. This completes the claim. □

Now we let W be $\sum_{e=uv \in E_1}(w(e, u) + w(e, v)) + \sum_{e \in E_2} \mu(e) + \sum_{e \in E_3} \nu(e)$, ψ be the maximum denominator of weights w, μ and ν, and ω be the minimum of weights w, μ and ν. The following theorem shows that the algorithm to problem WDBoundedTree gives an algorithm to problem MinimumWDTree.

Theorem 2. *Suppose that problem* WDBoundedTree *is (α', β')-approximable for some α' and β'. For an arbitrary $\epsilon > 0$, problem* MinimumWDTree *is $(\beta' + \epsilon)$-approximable in $O(L(|E| + |V| + \log(W/(\omega\epsilon)))$ time. If $E_2 = \emptyset$, then it is β'-approximable in $O(L(|E| + |V| + \log(W/\psi))$ time.*

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 $\max\{w_1(e, u), w_1(e, v)\} > r$, each edge $e \in E_2$ such that $\mu(e) > 2r$, and each edge

$e \in E_3$ such that $\nu(e) > r$. Let $b_r : V \to \mathbb{Q}_+$ be the function such that $b_r(v) = r$ for all $v \in V$, and $I_r = (G_r, w_1, \mu, \nu, A = V, b_r)$.

We denote $\min\{r \in \mathbb{Q}_+ \mid \mathrm{P_T}(I_r) \neq \emptyset\}$ by R, and the minimum weighted degree of the given instance by OPT. For given ϵ, define $\epsilon' = \omega\epsilon$. Since $\omega \leq \mathrm{OPT}$, we have $\epsilon' \leq \epsilon\mathrm{OPT}$. Since the characteristic vector of an optimal solution to the given instance of problem MinimumWDTree satisfies all constraints of $\mathrm{P_T}(I_{OPT})$, 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 programming over $\mathrm{P_T}(I_r)$ $\log(W/\epsilon')$ times.

Let T be an (α, β)-approximate solution to the instance of problem WDBoundedTree consisting of $I_{R'}$ and an arbitrary edge-cost c. By the β-approximability of T, we have $d_w(v;T) \leq \beta b_{R'}(v) \leq \beta(R+\epsilon') \leq \beta(1+\epsilon)\mathrm{OPT}$ for any $v \in V$. This implies that T is a $\beta(1+\epsilon)$-approximate solution to problem MinimumWDTree.

When $E_2 = \emptyset$, set ϵ so that $\epsilon' < \psi$ holds. In this case, if R' satisfies $R \leq R' \leq R + \epsilon'$, then $R' = R$. Such R' can be computed by solving the linear programming over $\log(W/\epsilon') = \log(W/\psi)$ times. Hence 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. □

Now we see that $\mathrm{P_T}(I)$ is $(1, 3)$-bounded. First let us observe that the key property of tight constraints observed in [17] also holds in our setting.

Lemma 1. *For any extreme point (x^*, y^*) of $\mathrm{P_T}(I)$, there exists a laminar family $\mathcal{L} = \{U \subseteq V \mid |U| \geq 2\}$ (i.e., any $U, U' \in \mathcal{L}$ satisfy either $U \subseteq U'$, $U' \subseteq U$, or $U \cap U' \neq \emptyset$) and $X \subseteq A$ 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 coefficients vectors of tight constraints in $\mathrm{P_T}(I)$. 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 (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 $X = \{v \in A \mid \sum_{e\in\delta(v;E_1)} w_1(e, v)x^*(e) + \sum_{e\in\delta(v;E_2)} \mu(e)y^*(e, v) + \sum_{e\in\delta(v;E_3)} \nu(e)y^*(e, v) = b(v)\}$ (i.e., set of vertices defining tight constraints of (6)).

For a subfamily $\mathcal{F}'$ of $\mathcal{F}$, we denote by *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 above operations.) In [17], it is proven that a maximal laminar subfamily $\mathcal{L}$ of $\mathcal{F}$ satisfies *span*$(\mathcal{L} \cup X) =$ *span*$(\mathcal{F} \cup X)$. Since the dimension of *span*$(\mathcal{L} \cup X)$ is at most $|\mathcal{L}| + |X|$, we have $|E_{x^*}| \leq |\mathcal{L}| + |X|$, as required. □

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

Corollary 1. *Problem* WDBoundedTree *is $(1, 4 + 3\theta + \kappa)$-approximable in $O(L(|V| + |E|))$ time. Problem* MinimumWDTree *is $(4 + \kappa)$-approximable in $O(L(|E| + |V| + \log(W/\psi))$ time*

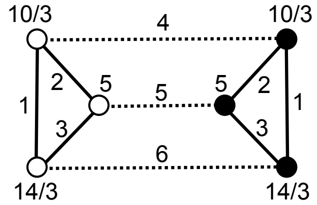

Figure 1: A counterexample for $(1, 2)$-boundedness of $\mathrm{P_T}(I)$

if $E_2 = \emptyset$*, and is* $(7 + \kappa + \epsilon)$*-approximable in* $O(L(|E| + |V| + \log(W/(\omega\epsilon)))$ *time for any* $\epsilon > 0$ *otherwise.*

Proof. Immediate from Theorems 1, 2 and 3. □

It is a natural question to ask whether the $(1, 3)$-boundedness of $\mathrm{P_T}(I)$ can be improved to $(1, 2)$-boundedness. Let us discuss this assuming that $E_2 = E_3 = \emptyset$. 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 [17]. Singh and Lau [17] weakened the $(1, 2)$-boundedness by replacing its first condition with the following:

- There exists an edge $e \in E$ such that $x^*(e) = 1$.

They then designed their algorithm by observing that the property holds for more general polytopes than $\mathrm{P_T}(I)$. This approach is also not useful for our setting because there exists a counterexample, 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 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 $\mathrm{P_T}(I)$ 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

In this section, we let I stand for the set of an undirected graph $G = (V, E)$ with $E = E_1 \cup E_2 \cup E_3$, weights $w_1 : E_1 \times V \to \mathbb{Q}_+$, $\mu : E_2 \to \mathbb{Q}_+$, $\nu : E_3 \to \mathbb{Q}_+$, 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 $y \in \mathbb{Q}^{(E_2 \cup E_3) \times V}$ that satisfy

$$0 \le x(e) \le 1 \quad \text{for all } e \in E, \tag{12}$$
$$0 \le y(e, u),\ y(e, v) \quad \text{for all } e = uv \in E_2 \cup E_3, \tag{13}$$
$$y(e, u) + y(e, v) = x(e) \quad \text{for all } e = uv \in E_2 \cup E_3, \tag{14}$$
$$x(\delta(U)) \ge f(U) \quad \text{for all non-empty } U \subset V, \tag{15}$$

and

$$\sum_{e\in\delta(v;E_1)} w_1(e,v)x(e) + \sum_{e\in\delta(v;E_2)} \mu(e)y(e,v) + \sum_{e\in\delta(v;E_3)} \nu(e)y(e,v) \le b(v) \quad \text{for all } v \in A. \qquad (16)$$

Observe that $\mathrm{P_N}(I)$ with $A = V$ is the polytope of a linear programming relaxation of problem WDBoundedNetwork.

We say that $\mathrm{P_N}(I)$ is (α,β)-*bounded* for some $\alpha,\beta \ge 1$ 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 $x^*(e) \ge 1/\alpha$;
- There exists a vertex $v \in A$ such that $|\delta(v;E_{x^*})| \le \beta$.

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

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

Algorithm for problem WDBoundedNetwork

Input: An undirected graph $G = (V,E)$ with $E = E_1 \cup E_2 \cup E_3$, weights $w_1 : E_1 \times V \to \mathbb{Q}_+$, $\mu : E_2 \to \mathbb{Q}_+$, $\nu : E_3 \to \mathbb{Q}_+$, 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, an allocation $w_2 : F_2 \times V \to \mathbb{Q}_+$ of μ and an allocation $w_2 : F_2 \times V \to \mathbb{Q}_+$ of ν, 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 $\mu(e) > b(u) + b(v)$.
- Delete $e = uv \in E_3$ from G if $\nu(e) > \max\{b(u), b(v)\}$.
- If $e = uv \in E_3$ and $b(u) \ge \nu(e) > b(v)$, then move e from E_3 to E_1 with setting $w_1(e,u) := \nu(e)$ and $w_1(e,v) := 0$. If $e \in E_3$ and $b(v) \ge \nu(e) > b(u)$, then move e from E_3 to E_1 with setting $w_1(e,u) := 0$ and $w_1(e,v) := \nu(e)$.

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 $x^*(e) \ge 1/\alpha$, 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 $b(v) := b(v) - \mu(e)y^*(e,v)$.

Case of $e \in E_3$**:** If $y^*(e,u) \geq y^*(e,v)$, then set $w_3(e,u) := \nu(e)$ and $w_3(e,v) := 0$. If $y^*(e,u) < y^*(e,v)$, then set $w_3(e,u) := 0$ and $w_3(e,v) := \nu(e)$. If $u \in A$, then set $b(u) := b(u) - \nu(e)y^*(e,u)$. If $v \in A$, then set $b(v) := b(v) - \nu(e)y^*(e,v)$;

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 as in Section 2, 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. Moreover define $\kappa = 1$ if $E_3 \neq \emptyset$, and $\kappa = 0$ otherwise. We let L denote the time for solving the linear programming over $\mathrm{P_N}(I)$.

Theorem 4. *If each* $\mathrm{P_N}(I)$ *constructed in Step 2 is* (α, β)*-bounded, then problem* WDBoundedNetwork *is* $(\alpha, \alpha(1+\kappa) + \beta(1+\theta))$*-approximable in* $O(L(|V|+|E|))$ *time.*

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

Observe that the linear programming 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 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 $\mu(e) \leq b(u) + b(v) \leq (1+\theta)b(v)$;
- If $e = uv \in E_3$, then $\nu(e) \leq b(u)$ and $\nu(e) \leq b(v)$.

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 $d_w(v;F) \leq (\alpha(1+\kappa) + \beta(1+\theta))b(v)$ 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

$$b_{i+1}(v') = \begin{cases} b_i(v') - w_1(e_i, v')x_i^*(e_i) & \text{if } v' \in A \text{ and } e_i \in E_1, \\ b_i(v') - \mu(e_i)y_i^*(e_i, v') & \text{if } v' \in A \text{ and } e_i \in E_2, \\ b_i(v') - \nu(e_i)y_i^*(e_i, v') & \text{if } v' \in A \text{ and } e_i \in E_3, \\ b_i(v') & \text{otherwise.} \end{cases} \tag{17}$$

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 variable 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 $\mathrm{P_N}(I_{i+1})$. Hence we have the following:

$$\text{If } \mathrm{P_N}(I_i) \neq \emptyset, \text{ then } \mathrm{P_N}(I_{i+1}) \neq \emptyset \text{ for } i \geq 0; \tag{18}$$

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

(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 (18), $\mathrm{P_N}(I_i) \neq \emptyset$ holds for all $1 \leq i \leq j$. Hence the algorithm terminates with outputting an f-connected subgraph $F = \{e_1, \ldots, e_j\}$, an allocation $w_2 : F_2 \times V \to \mathbb{Q}_+$ of μ, and an allocation $w_3 : F_3 \times V \to \mathbb{Q}_+$ of ν, by the way of construction.

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

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

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 $\alpha c^T x_1^* \geq c(F)$. Notice 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 $d_w(v; F) \leq (\alpha(1 + \kappa) + \beta(1 + \theta))b(v)$ 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

$$b(v) \geq \sum_{e_i \in \delta(v; F_1')} w_1(e_i, v)x_i^*(e_i) + \sum_{e_i \in \delta(v; F_2')} \mu(e_i)y_i^*(e_i, v) + \sum_{e_i \in \delta(v; F_3')} \nu(e_i)y_i^*(e_i, v).$$

If $e_i \in \delta(v; E_2)$, then $w_2(e_i, v) = \mu(e_i)y_i^*(e_i, v)/x_i^*(e_i)$. If $e_i \in \delta(v; E_3)$, then $\nu(e_i)y_i^*(e_i, v) \geq w_3(e_i, v)x_i^*(e_i)/2$ holds because even in the case of $w_3(e_i, v) = \nu(e_i)$, $y_i^*(e_i, v) \geq y_i^*(e_i, u)$ holds, and hence $y_i^*(e_i, v) \geq (y_i^*(e_i, u) + y_i^*(e_i, v))/2 = x_i^*(e_i)/2$. Recall that $x_i^*(e_i) \geq 1/\alpha$. Therefore,

$$\begin{aligned} &\sum_{e_i \in \delta(v; F_1')} w_1(e_i, v)x_i^*(e_i) + \sum_{e_i \in \delta(v; F_2')} \mu(e_i)y_i^*(e_i, v) + \sum_{e_i \in \delta(v; F_3')} \nu(e_i)y_i^*(e_i, v) \\ &\qquad \geq d_{w_1}(v; F_1')/\alpha + d_{w_2}(v; F_2')/\alpha + d_{w_3}(v; F_3')/(2\alpha). \end{aligned}$$

It implies that $d_w(v; F') \leq \alpha b(v)$ holds if $E_3 = \emptyset$, and $d_w(v; F') \leq 2\alpha b(v)$ otherwise.

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 $|\delta(v; F_a)| \leq \beta$ holds. We have already seen that, after Step 1, $e = uv \in E_1$ satisfies $w_1(e, v) \leq b(v)$, $e = uv \in E_2$ satisfies $w_2(e, v) \leq \mu(e) \leq (1 + \theta)b(v)$, and $e = uv \in E_3$ satisfies $w_3(e, v) \leq \nu(e) \leq b(v)$. 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 $d_w(v; F) \leq (\alpha + \beta(1 + \theta))b(v)$ if $E_3 = \emptyset$, and $d_w(v; F) \leq (2\alpha + \beta(1 + \theta))b(v)$ otherwise. This completes the claim. □

The following theorem shows that the algorithm for problem WDBoundedNetwork gives an algorithm for problem MinimumWDNetwork. Now we define W, ψ, and ω as in Section 2.

Theorem 5. *Suppose that problem* WDBoundedNetwork *is (α', β')-approximable for some α' and β'. For an arbitrary $\epsilon > 0$, problem* MinimumWDNetwork *is $(\beta' + \epsilon)$-approximable in $O(L(|E|+|V|+\log(W/(\omega\epsilon)))$. If $E_2 = \emptyset$, then it is β'-approximable in $O(L(|E|+|V|+\log(W/\psi))$.*

Proof. It can be derived from Theorem 4 as Theorem 2 is derived from Theorem 1. □

Now we see that polytope $\mathrm{P_N}(I)$ is $(2,5)$-bounded. First let us see that the key property of tight constraints observed in [8] also holds in our setting.

Lemma 2. *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* $\mathcal{L} = \{U \subset V \mid U \neq \emptyset, x^*(\delta(U)) = f(U)\}$ *and* $X = \{v \in A \mid \sum_{e\in\delta(v;E_1)} w_1(e,v)x^*(e) + \sum_{e\in\delta(v;E_2)} \mu(e)y^*(e,v) + \sum_{e\in\delta(v;E_3)} \nu(e)y^*(e,v) = b(v)\}$ *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 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 (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 *span*$(\mathcal{F}')$ the vector space spanned by the characteristic vectors of $\delta(U; E_{x^*})$, $U \in \mathcal{F}'$. In [8], it is proven that a maximal laminar subfamily $\mathcal{F}'$ of $\mathcal{F}$ satisfies *span*$(\mathcal{F}')$ = *span*$(\mathcal{F})$. So bases of *span*$(\mathcal{F}')$ and X gives the required $\mathcal{L}$ and X. $\square$

Theorem 6. *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 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 [8], 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 [8]. $\square$

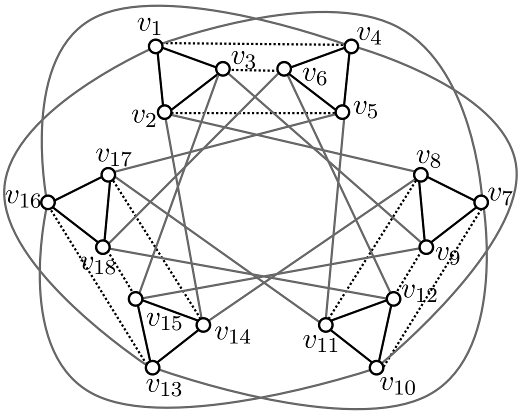

Figure 2: A counterexample for $(2,4)$-sparseness of $\mathrm{P_N}(I)$

Corollary 2. *Problem* WDBoundedNetwork *is* $(2, 7 + 5\theta + 2\kappa)$*-approximable in* $O(L(|V| + |E|))$. *Problem* MinimumWDNetwork *is* $(12+2\kappa+)$*-approximable in* $O(L(|E|+|V|+\log(W/\psi))$ *if* $E_2 = \emptyset$*, and is* $(12 + 2\kappa + \epsilon)$*-approximable in* $O(L(|E| + |V| + \log(W/(\omega\epsilon)))$ *for any* $\epsilon > 0$ *otherwise.*

Proof. Immediate from Theorems 4, 5 and 6. □

Lau et. al. [12] designed their algorithm for the case with $w_1(e, u) = w_1(e, v) = 1$, $e = uv \in E_1$ and $E_2 = E_3 = \emptyset$ by observing that $\mathrm{P_N}(I)$ is $(2, 4)$-bounded with such instances. However, 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 $E_2 = E_3 = \emptyset$ as indicated by the following counterexample; 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 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 for the network design problems which has upper-bound on weighted degree of each vertex. We also have seen that it is hard to to improve the approximation ratios by our approach based on the iterative rounding method. For further investigation, it may be interesting to extend problems which have constraints on degrees of vertices (e.g. matching problem, edge cover problem) by introducing the weighted degree.

Acknowledgements

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

References

[1] N. Bansal, R. Khandekar, and V. Nagarajan, *Additive guarantees for degree bounded directed network design*, tech. rep., IBM Research, 2007.

[2] K. Chaudhuri, S. Rao, S. Riesenfeld, and K. Talwar, *What would Edmonds do? augmenting paths and witnesses for degree-bounded MSTs*, in Proceedings of 8th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, Lecutre notes in computer science 3624, 2005, pp. 26–39.

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

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

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

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

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

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

[9] T. Kiraly, L. C. Lau, and M. Singh, *Degree bounded matroids and submodular flows*, in Proceedings of 13th Conference on Integer Programming and Combinatorial Optimization, 2008, to appear.

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

[11] J. Könemann and R. Ravi, *Primal-dual meets local search: approximating MST's with nonuniform degree bounds*, in Proceedings of the 35th annual ACM symposium on Theory of computing, 2003, pp. 389–395.

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

[13] L. C. Lau and M. Singh, *Additive approximation for bounded degree survivable network design*, in Proceedings of the 40th ACM Symposium on Theory of Computing, 2008, to appear.

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

[15] R. Ravi, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz, and 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, pp. 438–447.

[16] R. Ravi and 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, pp. 169–180.

[17] M. Singh and 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, pp. 661–670.