重み付き次数制約を持つネットワーク設計問題

福永拓郎　　永持仁

京都大学情報学研究科数理工学専攻

概要

重み $w : E \times V \to \mathbb{Q}_+$ が与えられている無向グラフ $G = (V, E)$ を考える．節点 $v \in V$ の重み付き次数 $d_w(v; E)$ を $\sum \{w(e, v) \mid v$ に接続する辺 $e \in E\}$ と定義する．本研究では，各節点の重み付き次数に対する上限が制約として与えられているネットワーク設計問題を考える．問題の入力は，辺集合 $E = E_1 \dot\cup E_2 \dot\cup E_3$ を持つ無向グラフ $G = (V, E)$，辺コスト $c : E \to \mathbb{Q}$，次数上限 $b : V \to \mathbb{Q}_+$ である．解は全域木 $T \subseteq E$ と重み $w_i : T_i \times V \to \mathbb{Q}_+$ $(i \in \{2, 3\})$ から成る（ただし，$T_i = T \cap E_i$）．$w_2(e, u) + w_2(e, v) = \mu(e)$ $(e = uv \in T_2)$，$\{w_3(e, u), w_3(e, v)\} = \{0, \nu(e)\}$ $(e = uv \in T_3)$，$d_{w_1}(v; T_1) + d_{w_2}(v; T_2) + d_{w_3}(v; T_3) \leq b(v)$ $(v \in V)$ を満たすとき，解が実行可能であると定義する．問題の目的は，$\sum_{e \in T} c(e)$ が最小となる実効可能解を求めることである．

我々は，重み付き次数に関する制約を違反することを許した上で，コストと重み付き次数を同時に抑えるような解を計算するアルゴリズムを提案する．また，解の最大重み付き次数を最小化する問題についても考える．

Network Design with Weighted Degree

Takuro Fukunaga　　Hiroshi Nagamochi

Department of Applied Mathematics and Physics,
Graduate School of Informatics, Kyoto University

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 $v \in V$ is defined as $\sum \{w(e, v) \mid e \in E$ incident with $v\}$. In this paper, we consider a network design problem with the 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 $w_1 : E_1 \times V \to \mathbb{Q}_+$, $\mu : E_2 \to \mathbb{Q}_+$ and $\nu : E_3 \to \mathbb{Q}_+$, an edge-cost $c : E \to \mathbb{Q}$, and a degree-bound $b : V \to \mathbb{Q}_+$. A solution consists of a spanning tree $T \subseteq E$ and weights $w_i : T_i \times V \to \mathbb{Q}_+$ for $i \in \{2, 3\}$, where T_i stands for $T \cap E_i$. It is defined to be feasible if it satisfies $w_2(e, u) + w_2(e, v) = \mu(e)$ for $e = uv \in T_2$, $\{w_3(e, u), w_3(e, v)\} = \{0, \nu(e)\}$ for $e = uv \in T_3$, and $d_{w_1}(v; T_1) + d_{w_2}(v; T_2) + d_{w_3}(v; T_3) \leq b(v)$ for each $v \in V$. The goal of this problem is to find a feasible solution that minimizes its cost $\sum_{e \in T} c(e)$. Relaxing the constraints on weighted degree, we propose bi-criteria approximation algorithms based on the iterative rounding. We also consider another problem that asks to minimize the maximum weighted degree of vertices.

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 report, 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. 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 report, 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

- $\sum_{e \in F} c(e) \leq \alpha \min\{\sum_{e \in F'} c(e) \mid F' \subseteq E$ 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 [14] and to the bounded-degree

survivable network design problem by Lau et al. [10], 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 [7]. 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 report, 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 [12, 8, 9, 1, 2, 13]. For the uniform cost (i.e., $c(e) = 1$ for $e \in E$), an optimal result was given by Fürer and Raghavachari [3]. Their algorithm computes a spanning tree which violates degree upper-bounds by at most one. For general costs, Goemans [5] 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 [14]; 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 [7], 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. [10] 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 [?]. Bansal et al. [?] considered the arborescence problem and survivable network problem with intersecting supermodular connectivity. Kiraly et al. [?] 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 [11] 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. [4] 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 report 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 report is organized as follows. Section 2 presents our algorithms to problems WDBound-edTree 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. Due to the space limitation, discussion about problems WDBoundedNetwork and MinimumWDNetwork is omitted.

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 $P_T(I)$ 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

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

$$0 \leq y(e, u), \ y(e, v) \qquad \text{for all } e = uv \in E_2 \cup E_3, \tag{2}$$

$$y(e, u) + y(e, v) = x(e) \qquad \text{for all } e = uv \in E_2 \cup E_3, \tag{3}$$

$$x(E) = |V| - 1, \tag{4}$$

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

and

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

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

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

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

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

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

Observe that $P_T(I)$ with $A = V$ is the polytope of a linear programming relaxation of problem WD-BoundedTree. Although (5) has an exponentially many number of constraints, linear programming over the polytope is solvable in polynomial time by using the ellipsoid method [1] or by transforming it to a polynomial-size formulation [6].

For a vector $x \in \mathbb{Q}_+^E$, let E_x denote $\{e \in E \mid x(e) > 0\}$. We say that polytope $P_T(I)$ 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 $P_T(I)$ is $(1, \beta)$-bounded. By this and the fact that $P_T(I)$ 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 $P_T(I)$ 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 only see that the $(1 + \beta(1 + \theta) + \kappa)$-approximability of the algorithm, due to the space limitation.

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. 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

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

also holds for $i \geq 1$.

Fix v as an arbitrary vertex. 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. $\qquad\square$

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 $\omega \leq$ OPT, we have $\epsilon' \leq \epsilon$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 $R \leq$ 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 $d_w(v; T) \leq \beta b_{R'}(v) \leq \beta(R + \epsilon') \leq \beta(1 + \epsilon)$OPT for any $v \in V$. This implies that T is a $\beta(1 + \epsilon)$-approximate solution to problem MINIMUMWDTREE.
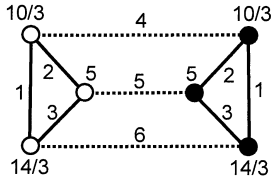
Figure 1: A counterexample for $(1,2)$-boundedness of $P_T(I)$

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 $d_w(v;T) \leq \beta b_{R'}(v) \leq \beta\text{OPT}$ for any $v \in V$, which implies that T is a β-approximate solution. $\square$

Now we see that $P_T(I)$ is $(1,3)$-bounded. First let us observe that the key property of tight constraints observed in [14] also holds in our setting. We omit the proof due to the space limitation.

Lemma 1. *For any extreme point (x^*, y^*) of $P_T(I)$, 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 $|E_{x^*}| \leq |\mathcal{L}| + |X|$.* $\square$

Theorem 3. *Polytope $P_T(I)$ 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. $\square$

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 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. $\square$

It is a natural question to ask whether the $(1,3)$-boundedness of $P_T(I)$ 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 [14]. Singh and Lau [14] 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 $P_T(I)$. 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 $P_T(I)$ such that $x^*(e) = 2/3$ for edges represented by solid lines, and $x^*(e) = 1/3$ for edges represented by dotted lines. Clearly, $x^*(e) < 1$ for any edge $e \in E$ and $\min_{v \in A = V} |\delta(v; E_{x^*})| = 3$.

Acknowledgements

References

[1] 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.

[2] 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.

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

[4] 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.

[5] 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.

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

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

[8] 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.

[9] 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.

[10] 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.

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

[12] 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.

[13] 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.

[14] 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.