Approximating Minimum Bounded Degree Spanning Trees to within One of Optimal

Mohit Singh*
Tepper School of Business
Carnegie Mellon University
Pittsburgh, PA USA
mohits@andrew.cmu.edu

Lap Chi Lau†
CSE Dept.
The Chinese Univ. of Hong Kong
Shatin, Hong Kong
chi@cse.cuhk.edu.hk

ABSTRACT

In the MINIMUM BOUNDED DEGREE SPANNING TREE problem, we are given an undirected graph with a degree upper bound B_v on each vertex v, and the task is to find a spanning tree of minimum cost which satisfies all the degree bounds. Let OPT be the cost of an optimal solution to this problem. In this paper, we present a polynomial time algorithm which returns a spanning tree T of cost at most OPT and $d_T(v) \leq B_v + 1$ for all v, where $d_T(v)$ denotes the degree of v in T. This generalizes a result of Furer and Raghavachari [8] to weighted graphs, and settles a 15-year-old conjecture of Goemans [10] affirmatively. The algorithm generalizes when each vertex v has a degree lower bound A_v and a degree upper bound B_v, and returns a spanning tree with cost at most OPT and $A_v - 1 \leq d_T(v) \leq B_v + 1$ for all v. This is essentially the best possible. The main technique used is an extension of the iterative rounding method introduced by Jain [12] for the design of approximation algorithms.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Non Numerical Algorithms and Problems—*Computations on discrete structures*; G.2.2 [**Discrete Mathematics**]: Graph Theory—*Network Problems, Trees.*

General Terms

Algorithms, Theory

Keywords

Approximation algorithms, Spanning Trees, Bounded degree, Iterative rounding.

*Supported by NSF ITR grant CCR-0122581 (The ALADDIN project) and NSF Grant CCF- 0430751.

†This work was partly done during a visit to Egerváry Research Group on Combinatorial Optimization (EGRES) in Budapest. Supported by European MCRTN Adonet, Contract Grant No. 504438.

STOC'07, June 11–13, 2007, San Diego, California, USA.

1. INTRODUCTION

The MINIMUM BOUNDED DEGREE SPANNING TREE problem (MBDST) is defined as follows: Given a simple undirected graph $G = (V, E)$, a cost function $c : E \to \mathbb{R}$ and a degree upper bound B_v for each vertex $v \in V$, find a spanning tree of minimum cost which satisfies all the degree bounds. Let OPT be the cost of an optimal solution to this problem. An $(\alpha, f(B_v))$-approximation algorithm[1] is an algorithm which returns a spanning tree T with cost at most $\alpha \cdot$ OPT and $d_T(v) \leq f(B_v)$ for all v, where $d_T(v)$ denotes the degree of v in T. When all degree bounds are 2 (i.e. $B_v = 2$ for all v), the MBDST problem specializes to the MINIMUM COST HAMILTONIAN PATH problem, and thus is NP-hard. In unweighted graphs, Furer and Raghavachari [8] gave an elegant $(1, B_v + 1)$-approximation algorithm for the MBDST problem. Goemans [10] conjectured that this result can be generalized to weighted graphs.

CONJECTURE 1.1. *In polynomial time, one can find a spanning tree of maximum degree at most $k + 1$ whose cost is no more than the cost of a minimum cost tree with maximum degree at most k.*

Note that the above conjecture is formulated in the special case where $B_v = k$ for all v. Recently, Goemans [10] made a major step towards this conjecture by giving a polynomial time $(1, B_v + 2)$-approximation algorithm for the MBDST problem. In this paper, we settle Conjecture 1.1 positively by proving the following result:

THEOREM 1.2. *There exists a polynomial time $(1, B_v + 1)$-approximation algorithm for the* MINIMUM BOUNDED DEGREE SPANNING TREE *problem.*

Theorem 1.2 also generalizes to the setting when there is a degree lower bound A_v and a degree upper bound B_v for each vertex $v \in V$. In this case, the algorithm returns a spanning tree T such that $A_v - 1 \leq d_T(v) \leq B_v + 1$ and the cost of T is at most OPT, where OPT is the minimum cost of a spanning tree which satisfies all degree (upper and lower) bounds. Note that we do not assume that the cost function satisfies triangle inequalities (or even non-negativity). With this general cost function, it is not possible to obtain any approximation algorithm if we insist on satisfying all the degree upper bounds [9][2]. Thus, Theorem 1.2 is essentially the best possible.

[1]Notice that the first parameter is used to specify the *ratio*, while the second parameter is used to specify the *actual bound*.

[2]Assuming P $\neq$ NP, there is no $(p(n), B_v)$-approximation algorithm for any polynomial $p(n)$ of n where n is the number of vertices.

1.1 Techniques

Polyhedral combinatorics has proved to be a powerful, coherent, and unifying tool in combinatorial optimization (see [20]). In the last two decades, polyhedral methods have also been applied very successfully to the design of approximation algorithms (see [21]). A standard approach to design approximation algorithms is to first formulate the problem as an integer program, and then use the linear relaxation of this program as a way to lower-bound the cost of an optimal solution. We shall also use this approach. Given an undirected graph $G = (V, E)$ and a subset S of vertices, we denote $E(S) = \{e \in E \ : \ |e \cap S| = 2\}$, i.e., edges which have both endpoints in S. We also denote $\delta(S)$ the edges which have exactly one endpoint in S. For $x \ : \ E \to \mathbb{R}^+$ and $U \subseteq E$, we denote $x(U) := \sum_{e \in U} x(e)$. As in Goemans' result [10], we use the following natural linear programming relaxation for the MINIMUM BOUNDED DEGREE SPANNING TREE problem.

$$\begin{array}{lrcll} \text{minimize} & c(x) & = & \sum_{e \in E} c_e \, x_e & & (1) \\ \text{subject to} & x(E(V)) & = & |V| - 1 & & (2) \\ & x(E(S)) & \leq & |S| - 1 & \forall \, S \subset V & (3) \\ & x(\delta(v)) & \leq & B_v & \forall \, v \in V & (4) \\ & x_e & \geq & 0 & \forall \, e \in E & (5) \end{array}$$

Using a polyhedral approach, a general strategy is to construct a spanning tree of cost no more than the optimal value of the above linear program, and in which the degree of each vertex is at most $B_v + 1$. This would prove Theorem 1.2. In fact, this general strategy has been used in previous work, and different techniques have been proposed to "round" the above linear program. An important observation of Goemans is that a *basic feasible solution* (or an *extreme point solution*) of the above linear program is characterized by a *laminar family* (definitions will be provided later) of *tight constraints* (inequalities that are satisfied as equalities), and he exploited this fact cleverly in [10] to design an $(1, B_v + 2)$-approximation algorithm for the MBDST problem.

We note that a very similar observation was made by Jain [12] in his breakthrough work on the SURVIVABLE NETWORK DESIGN problem, where he first introduced the idea of *iterative rounding* to the design of approximation algorithms. This potential connection initiated our approach to the BOUNDED DEGREE SURVIVABLE NETWORK DESIGN problem. Recently, in joint work [15] with Naor and Salavatipour, we have extended Jain's iterative rounding method to give the first constant factor (bi-criteria) approximation algorithm for bounded degree network design problems including the MINIMUM BOUNDED DEGREE STEINER TREE problem, BOUNDED DEGREE SURVIVABLE NETWORK DESIGN problem, etc. Inspired by these results, we attempted Conjecture 1.1 using the iterative rounding method.

The basic setting of the iterative rounding method for network design problems goes as follows. First we solve the linear program to obtain a basic optimal solution x^*. We proceed by adding the edges with the highest fractional value to the integral solution. Then we construct the *residual problem* where the edges added previously are fixed, and update the linear program appropriately. A key feature of the iterative rounding method is to repeat this procedure: solve again the linear program for the residual problem to obtain a basic optimal solution (instead of using x^*), and add the edges with the highest fractional value in this new fractional solution to the integral solution. This procedure is iterated until the integral solution constructed is a feasible solution. In the SURVIVABLE NETWORK DESIGN problem, the crucial theorem in Jain's approach is that the edges picked in each iteration have fractional value at least $1/2$, which ensures that the above algorithm has an approximation ratio of 2. This theorem relies heavily on the properties of a basic solution, as in Goemans' theorem.

The iterative rounding method can also be applied to solve problems optimally. For this purpose, we could only pick an edge e with $x^*_e = 1$ (we call such an edge e an *1-edge*). The above linear program without the degree constraints (constraints from (4)) is the standard linear programming formulation of the MINIMUM SPANNING TREE problem, and this iterative rounding approach (by only picking 1-edges) can be used to construct a minimum spanning tree, as we will show in Section 2.

For the MINIMUM BOUNDED DEGREE SPANNING TREE problem, however, both approaches would not work directly. The former approach of picking an edge e with $x^*_e \geq \frac{1}{2}$ would not work because we could not guarantee the optimality (with respect to the cost of the linear program) of the solution, while the latter approach of picking 1-edges would not work because the algorithm may not make progress in case there is no 1-edge.

We propose a way to combine and extend the ideas of these results. In particular, we show that only adding 1-edges to the solution can also be used to design approximation algorithms via the iterative rounding method. Thus our algorithm *does not round*. Our algorithm would keep adding 1-edges to the solution whenever possible. Of course, we cannot always guarantee the existence of an 1-edge, for otherwise we would have solved the problem optimally and satisfied all the degree bounds. The key insight is that if an 1-edge does not exist, then there must be a vertex with degree upper bound B_v and with at most $B_v + 1$ edges incident at it in the support of a basic feasible solution. We call such a vertex a *special* vertex. To proceed, we *remove* the degree constraints of all special vertices and re-solve the linear program again. The heart of our analysis is to show that there is an 1-edge if there is no special vertex. This is proved by a counting argument similar to that of Jain [12], which relies heavily on the fact that a basic feasible solution is characterized by a laminar family of tight constraints (as in [12, 10]). In this way, eventually we construct a spanning tree by picking only 1-edges, which ensures the optimality of the cost. Observe that by removing the degree constraint of a special vertex, the degree constraint at this vertex could only be violated by at most an additive constant of one, and so Theorem 1.2 follows. We remark that the idea of removing the degree constraint of a special vertex comes from the joint work on bounded degree network design problems [15]. These results demonstrate that the iterative rounding method is quite general and powerful, and we hope that our results will shed light on further applications of this method.

1.2 Related Work

The MINIMUM BOUNDED DEGREE SPANNING TREE problem is a well studied problem and has been attacked using a variety of techniques. Initial efforts on the problem were concentrated on obtaining bi-criteria approximation algorithms. Ravi et al [18] gave an $(O(\log n), O(B_v \log n))$-approximation for the MBDST problem using a matching-based augmentation technique. Konemann and Ravi [13, 14] used a Lagrangian-relaxation based approach to obtain an $(O(1), O(B_v + \log n))$-approximation algorithm. Chaudhuri et al [1, 2] presented an $(1, O(B_v + \log n))$-approximation algorithm, and an $(O(1), O(B_v))$-approximation algorithm based on the push-relabel framework developed for the maximum flow problem. Ravi and Singh [19] considered a variant of the problem in which the tree returned must be a minimum spanning tree. They gave an algorithm that returns an MST in which the degree of any vertex v is at most $B_v + p$, where p is the number of distinct costs in any MST. Recently, Goemans [10] presented an $(1, B_v + 2)$-approximation algorithm using matroid intersection techniques. This was the previous best guarantee for the MBDST problem. In the special case where the graph is unweighted, Furer and Raghavachari [8] developed an algorithm, based on a variant of local search, to return a spanning tree in which the degree of each vertex v is at most $B_v + 1$.

The iterative rounding technique that we use in our algorithm was developed by Jain [12] for the SURVIVABLE NETWORK DESIGN problem and has later been successfully applied to various problems [4, 7]. Recently, this technique has been extended to give constant factor bi-criteria approximation algorithm for the BOUNDED DEGREE SURVIVABLE NETWORK DESIGN problem [15].

1.3 Organization

The rest of the paper is organized as follows. In Section 2, we give an iterative procedure which shows the integrality of the spanning tree polyhedron. Then, in Section 3, we present a simple $(1, B_v + 2)$-approximation algorithm for the MBDST problem via iterative rounding. This matches the previous best result of Goemans [10]. In Section 4, we present the main algorithm and the proof of Theorem 1.2. Finally, in section 5, we extend the algorithm to deal with degree lower bounds.

2. SPANNING TREE POLYHEDRON

In this section, we present an iterative procedure to find a minimum spanning tree from a basic optimal solution of a linear program. This motivates the main result of the paper and illustrates the basic proof techniques. Let $G = (V, E)$ be a graph with a cost function c on edges. A classical result of Edmonds [6] states that the following linear program LP-MST(G) is integral, and a basic optimal solution is always a minimum spanning tree.

$$
\begin{aligned}
\text{minimize} \quad & c(x) = \sum_{e \in E} c_e\, x_e && (6)\\
\text{subject to} \quad & x(E(V)) = |V| - 1 && (7)\\
& x(E(S)) \leq |S| - 1 \qquad \forall\, S \subset V && (8)\\
& x_e \geq 0 \qquad \forall\, e \in E && (9)
\end{aligned}
$$

The following is an iterative procedure to obtain a minimum spanning tree of G.

Iterative MST Algorithm

1. Initialization $F \leftarrow \emptyset$.
2. While $V(G) \neq \emptyset$ do
   (a) Find a basic optimal solution x^* of LP-MST(G) and remove every edge e with $x^*_e = 0$ from G.
   (b) Find a vertex v with at most one edge $e = uv$ incident at it, and update $F \leftarrow F \cup \{e\}$, $G \leftarrow G \setminus \{v\}$.
3. Return F.

Figure 1: MST Algorithm

First assume that the above algorithm terminates. We claim that the solution F returned by the algorithm is a spanning tree of G of cost no more than the cost of the initial LP solution x^*, and hence a minimum spanning tree. The argument will proceed by induction on the number of iterations of the algorithm.

If the algorithm finds a vertex v of degree one (a leaf vertex) in Step 2b with an edge $e = \{u, v\}$ incident at v, then we must have $x^*_e = 1$ since $x(\delta(v)) \geq 1$ is a valid inequality of the LP (subtract the constraint (8) for $S = V \setminus \{v\}$ from the constraint (7)). Intuitively, v is a leaf of the spanning tree. Hence, we add e to the solution F (initially $F = \emptyset$), and remove v from the graph. Note that for any spanning tree T' of $G' = G \setminus \{v\}$, we can construct a spanning tree $T = T' \cup \{e\}$ of G. Hence, the residual problem is to find a minimum spanning tree on $G \setminus v$, and we apply the same procedure to solve the residual problem recursively. Observe that the restriction of x^* to $E(G')$, denoted by x^*_{res}, is a feasible solution to LP-MST(G'). Inductively, the algorithm will return a spanning tree F' of cost at most the optimal value of LP-MST(G'), and hence $c(F') \leq c \cdot x^*_{res}$, as x^*_{res} is a feasible solution to LP-MST(G'). So, we have

$$c(F) = c(F') + c_e \text{ and } c(F') \leq c \cdot x^*_{res}$$

which imply that

$$c(F) \leq c \cdot x^*_{res} + c_e = c \cdot x^*$$

as $x^*_e = 1$. Therefore, the spanning tree returned by the algorithm is of cost no more than the cost of the LP solution x^*, which is a lower bound on the optimal cost. This shows that the algorithm returns a minimum spanning tree of the graph.

It remains to show that the algorithm will terminate, or that we can always finds a vertex v of degree one in Step 2b.

LEMMA 2.1. *For any basic solution x^* of LP-MST(G) with support $E^* = \{e \mid x^*_e > 0\}$, there exists a vertex v such that $deg_{E^*}(v) = 1$.*

A basic solution is defined to be the unique solution of m linearly independent tight constraints (constraints which achieve equality), where m denotes the number of variables in the linear program. For any edge e, if $x^*_e = 0$, we can remove the edge e from the graph and consider only the edges in E^*. Thus we can assume that there is no tight constraints from (9). To prove Lemma 2.1, we shall prove that there are at most $n - 1$ tight constraints from (7)-(8), where n denotes the number of vertices in the graph. This can be shown by an uncrossing technique. For a set $S \subseteq V$, the corresponding constraint $x^*(E(S)) \leq |S| - 1$ defines a vector in $\mathbb{R}^{|E|}$: the vector has a 1 corresponding to each edge $e \in E(S)$, and 0 otherwise. We call this vector the *characteristic vector* of $E(S)$, and denote it by $\chi_{E(S)}$. Let $\mathcal{F} = \{S \mid x^*(E(S)) = |S| - 1\}$ be the set of tight constraints from (7)-(8). Denote by $span(\mathcal{F})$ the vector space generated by the set of vectors $\{\chi_{E(S)} \mid S \in \mathcal{F}\}$. We say two sets X, Y are *intersecting* if $X \cap Y$, $X - Y$ and $Y - X$ are nonempty. A family of sets is *laminar* if no two sets are intersecting. From standard uncrossing arguments (see e.g. Cornuejols et al [5], Jain [12]) it follows that we can obtain a laminar family $\mathcal{L} \subseteq \mathcal{F}$ such that $span(\mathcal{L}) = span(\mathcal{F})$. For completeness we include a proof here to illustrate the uncrossing technique. First we need an "uncrossing" lemma on intersecting sets.

LEMMA 2.2. *[10] If $S, T \in \mathcal{F}$ and $S \cap T \neq \emptyset$, then both $S \cap T$ and $S \cup T$ are in $\mathcal{F}$. Furthermore, $\chi_{E(S)} + \chi_{E(T)} = \chi_{E(S \cap T)} + \chi_{E(S \cup T)}$.*

PROOF. As $S \cap T \neq \emptyset$, we have:

$$
\begin{aligned}
|S| - 1 + |T| - 1 &= |S \cap T| - 1 + |S \cup T| - 1\\
&\geq x^*(E(S \cap T)) + x^*(E(S \cup T))\\
&\geq x^*(E(S)) + x^*(E(T))\\
&= |S| - 1 + |T| - 1
\end{aligned}
$$

and hence we have equality throughout. This implies that $S \cup T$ and $S \cap T$ are both in $\mathcal{F}$, and furthermore there are no edges $e \in E^*$ between $S \setminus T$ and $T \setminus S$. Therefore, $\chi_{E(S)} + \chi_{E(T)} = \chi_{E(S \cap T)} + \chi_{E(S \cup T)}$. $\square$

A basic solution is characterized by a set of linearly independent tight constraints. The following lemma implies that a basic solution of LP-MST(G) is characterized by a laminar family of tight constraints.

LEMMA 2.3. *[12] If $\mathcal{L}$ is a maximal laminar subfamily of $\mathcal{F}$, then $span(\mathcal{L}) = span(\mathcal{F})$.*

PROOF. Let $\mathcal{L}$ be a maximal laminar subfamily of $\mathcal{F}$ and assume that $\chi_{E(S)} \notin span(\mathcal{L})$ for some $S \in \mathcal{F}$. Choose one such set S that intersects as few sets of $\mathcal{L}$ as possible. Since $\mathcal{L}$ is a maximal laminar family, there exists $T \in \mathcal{L}$ that intersects S. From Lemma 2.2, we have that $S \cap T$ and $S \cup T$ are also in $\mathcal{F}$ and that $\chi_{E(S)} + \chi_{E(T)} = \chi_{E(S \cap T)} + \chi_{E(S \cup T)}$. Since $\chi_{E(S)} \notin span(\mathcal{L})$, either $\chi_{E(S \cap T)} \notin span(\mathcal{L})$ or $\chi_{E(S \cup T)} \notin span(\mathcal{L})$. In either case, we have a contradiction because both $S \cup T$ and $S \cap T$ intersect fewer sets in $\mathcal{L}$ than S; this is because every set that intersects $S \cup T$ or $S \cap T$ also intersects S. $\square$

The proof of Lemma 2.1 follows from Lemma 2.3.

Proof of Lemma 2.1: Suppose each vertex has degree at least two. Then $|E^*| \geq \frac{1}{2} \sum_{v \in V} deg_{E^*}(v) = |V|$.

Recall that a basic solution is the unique solution of m linearly independent constraints, where m is the number of variables in the linear program. As x^* is a basic solution and there are no tight constraints from (9), we have $|E^*| = |\mathcal{L}|$. A simple inductive argument shows that a laminar family on a ground set of size n containing no singleton sets has at most $n-1$ sets. Hence, $|\mathcal{L}| \leq |V| - 1$ and so $|E^*| = |\mathcal{L}| \leq |V| - 1$, a contradiction. $\square$

REMARK 2.4. *If x^* is an optimal basic solution to LP-MST(G), then the residual LP solution x^*_{res}, which is x^* restricted to $G' = G \setminus v$, remains an optimal basic solution to LP-MST(G'). Hence, in the MST Algorithm we only need to solve the original linear program once and none of the residual linear programs. Alternatively, Lemma 2.1 shows that $|E^*| = n-1$ and since $x(E^*) = n-1$ and $x(e) \leq 1$ for all edges $e \in E^*$ (by considering constraints (9) for size two sets), we must have $x_e = 1$ for all edges $e \in E^*$ proving integrality of the spanning tree polyhedron.*

3. A +2 APPROXIMATION ALGORITHM

In this section we first present an $(1, B_v + 2)$-approximation algorithm for the MBDST problem via iterative rounding. This algorithm is simple, and it illustrates the idea of removing degree constraints. We use the following standard linear programming relaxation for the MBDST problem, which we denote by LP-MBDST$(G, \mathcal{B}, W)$. In the following we assume that degree bounds are given for vertices only in a subset $W \subseteq V$. Let $\mathcal{B}$ denote the vector of all degree bounds B_v one for each $v \in W$.

$$\begin{array}{llcll} \text{minimize} & c(x) & = & \sum_{e \in E} c_e\, x_e & (10) \\ \text{subject to} & x(E(V)) & = & |V| - 1 & (11) \\ & x(E(S)) & \leq & |S| - 1 \quad \forall\, S \subset V & (12) \\ & x(\delta(v)) & \leq & B_v \quad \forall\, v \in W & (13) \\ & x_e & \geq & 0 \quad \forall\, e \in E & (14) \end{array}$$

Observe that LP-MBDST$(G, \mathcal{B}, W)$ has an exponential number of constraints. Cunningham [3] gave a polynomial time procedure to separate over constraints (11)-(12) and (14). Separating over constraints (13) is clearly in polynomial time. Hence, using the ellipsoid algorithm one can optimize over LP-MBDST$(G, \mathcal{B}, W)$ in polynomial time. An alternative is to write a compact formulation for the above linear program [16] which has polynomially many variables and constraints.

Our $(1, B_v + 2)$-approximation algorithm in Figure 2 is a simple iterative rounding procedure for LP-MBDST$(G, \mathcal{B}, W)$.

MBDST Algorithm

1. Initialization $F \leftarrow \emptyset$.
2. While $V(G) \neq \emptyset$ do
   (a) Find a basic optimal solution x^* of LP-MBDST$(G, \mathcal{B}, W)$ and remove every edge e with $x^*_e = 0$ from G. Let the support of x^* be E^*.
   (b) If there exists a vertex $v \in V$, such that there is at most one edge $e = uv$ incident at v in E^*, then update $F \leftarrow F \cup \{e\}$, $G \leftarrow G \setminus \{v\}$, $W \leftarrow W \setminus \{v\}$, and also update $\mathcal{B}$ by setting $B_u \leftarrow B_u - 1$.
   (c) If there exists a vertex $v \in W$ such that $deg_{E^*}(v) \leq 3$ then update $W \leftarrow W \setminus \{v\}$.
3. Return F.

Figure 2: MBDST Algorithm

Before we prove the correctness of the algorithm, we give a high-level description and some intuition. First we remove all edges e with $x^*_e = 0$ and focus on the edges with positive fractional value, i.e. $x^*_e > 0$. In Step 2b, if v is of degree 1 and $e = uv$ is the only edge incident at v, then v is a leaf of the spanning tree. So, we add e to the solution F (initially $F = \emptyset$), remove v from the graph, and update the LP appropriately. Note that since $x^*_e = 1$, we maintain the optimality of the cost and also do not violate any degree constraint. Of course, we cannot always guarantee such an edge exists, otherwise we would have solved the problem exactly. The crucial observation is that if there is no leaf vertex, then there must exist a vertex v with at most three edges incident at it and the degree constraint for v is present in the linear program (i.e. $v \in W$). In Step 2c, we remove the degree constraint of a vertex v if v has at most three edges incident at it. By doing so, the degree constraint of v is violated by at most an additive constant of two, since in the worst case $B_v = 1$ and all the three edges incident at v are used in the returned solution F. In each iteration, we either remove a degree constraint or include an edge in our solution. Therefore, in a total of at most $n + n - 1 = 2n - 1$ iterations, we construct a spanning tree by including only 1-edges. These steps provide a simple $(1, B_v + 2)$-approximation algorithm for the MBDST problem.

We start the proof by a characterization of a basic feasible solution of LP-MBDST$(G, \mathcal{B}, W)$. We remove all edges with $x^*_e = 0$ and focus only on the support of the basic solution and the tight constraints from (11)-(13). Let $\mathcal{F} = \{S \mid x^*(E(S)) = |S| - 1\}$ correspond to the set of tight constraints from (11)-(12), and let $T = \{v \in W \mid x^*(\delta(v)) = B_v\}$ correspond to the set of tight degree constraints from (13). The proof of the following lemma is based on the uncrossing techniques used in Section 2; we shall also prove it in a more general setting in Lemma 4.3.

LEMMA 3.1. *Let x^* be any basic solution of LP-MBDST$(G, \mathcal{B}, W)$ with support E^*. Then there exists a set $T \subseteq W$ and a laminar family $\mathcal{L}$ such that x^* is the unique solution to the following linear system.*

$$\begin{cases} x^*(\delta(v)) = B_v & \forall v \in T \\ x^*(E(S)) = |S| - 1 & \forall S \in \mathcal{L} \end{cases}$$

Moreover, the characteristic vectors $\{\chi_{E(S)} : S \in \mathcal{L}\} \cup \{\chi_{\delta(v)} : v \in T\}$ are linearly independent. Furthermore, $|E^*| = |\mathcal{L}| + |T|$.

In the next lemma we prove (by a very simple counting argument) that in each iteration we can proceed by applying either Step 2b or Step 2c; this will ensure that the algorithm terminates.

LEMMA 3.2. *Any basic feasible solution x^* of LP-MBDST$(G, \mathcal{B}, W)$ with support E^* must satisfy one of the following.*
(a) There is a vertex $v \in V$ such that $deg_{E^*}(v) = 1$.
(b) There is a vertex $v \in W$ such that $deg_{E^*}(v) \leq 3$.

PROOF. Suppose for sake of contradiction that both (a) and (b) are not satisfied. Then every vertex has at least 2 edges incident at it and every vertex in W has at least 4 edges incident at it. Therefore, $|E^*| \geq (2(n - |W|) + 4|W|)/2 = n + |W|$, where $n = |V(G)|$.

By Lemma 3.1, there is a laminar family $\mathcal{L}$ and a set $T \subseteq W$ of vertices such that $|E^*| = |\mathcal{L}| + |T|$. As $\mathcal{L}$ contains subsets of size at least two, $|\mathcal{L}| \leq n-1$. Hence, $|E^*| = |\mathcal{L}| + |T| \leq n-1+|T| \leq n - 1 + |W|$, a contradiction. □

From Lemma 3.2 and the previous discussion, we obtain the following theorem of Goemans [10].

THEOREM 3.3. **(Goemans [10])** *There exists a polynomial time $(1, B_v+2)$-approximation algorithm for the* MINIMUM BOUNDED DEGREE SPANNING TREE *problem.*

4. A +1 APPROXIMATION ALGORITHM

In this section we present an $(1, B_v + 1)$-approximation algorithm for the MBDST problem. The general approach is similar to the MBDST algorithm in Figure 2. Observe that, in Step 2c of the MBDST algorithm, by removing the degree constraint of a vertex $v \in W$ only when a vertex has degree at most two (or more generally, removing a degree constraint for a vertex v if $deg_{E^*}(v) \leq B_v + 1$), we could ensure that the degree of every vertex is violated by at most one. However, it may no longer be the case that there exists a leaf vertex if every vertex in W just has degree at least three (instead of four), and so the algorithm may not be able to proceed. To overcome this, in Step 2b we not only look for 1-edges incident at leaf vertices but include any 1-edge in our integral solution. However, the residual problem is no longer an MBDST problem, since the endpoints of this edge are not necessarily leaf vertices. Hence, we define the following more general problem which is *self-reducible*, i.e. the problem in a later iteration is still of the same form. We call this problem the MINIMUM BOUNDED-DEGREE CONNECTING TREE (MBDCT) problem, and we present an $(1, B_v + 1)$-approximation algorithm for this more general problem by the iterative rounding method.

The MINIMUM BOUNDED-DEGREE CONNECTING TREE problem is defined as follows. We are given a graph $G = (V, E)$, a degree upper bounds B_v for each vertex v in some subset $W \subseteq V$, a cost function $c : E \to \mathbb{R}$, and a forest F. We assume without loss of generality that $E(F) \cap E(G) = \emptyset$. The task is to find a minimum cost forest H such that $H \cup F$ is a spanning tree of G and $d_H(v) \leq B_v$. We call such a forest H an *F-tree* of G, and a connected component of F a *supernode*; note that an isolated vertex of F is also a supernode. Intuitively, the forest F is the partial solution we have constructed so far, and H is a spanning tree in the graph where each supernode is contracted into a single vertex. We denote this contracted graph by G/F. Observe that when $F = \emptyset$ the MBDCT problem is just the MBDST problem.

We need some notation to define the linear programming relaxation for the MBDCT problem. For any set $S \subseteq V(G)$ and a forest F on G, let $F(S)$ be the set of edges in F with both endpoints in S, i.e., $\{e \in F : |e \cap S| = 2\}$. Note that $F(V)$ is just equal to $E(F)$. We denote $\mathcal{C}(F)$ the sets of supernodes of F. A set S is *non-intersecting with* F if for each $C \in \mathcal{C}(F)$ we either have $C \subseteq S$ or $C \cap S = \emptyset$. We denote $\mathcal{I}(F)$ the family of all subsets which are non-intersecting with F.

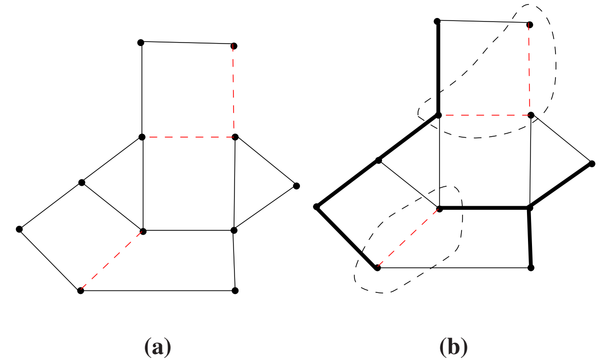

Figure 3: In Figure (a), the dashed edges correspond to F. In Figure (b), the bold edges H form an F-tree of G as $F \cup H$ is a spanning tree of G or equivalently, H is a spanning tree of G/F.

The following is a linear programming relaxation for the MBDCT problem, which we denote by LP-MBDCT$(G, \mathcal{B}, W, F)$. In the linear program we have a variable x_e for each edge e which has at most one endpoint in any one component of forest F. Indeed we assume (without loss of generality) that E does not contain any edge with both endpoints in the same component of F.

$$\begin{array}{rrcll} \text{minimize} & c(x) & = & \sum_{e \in E} c_e x_e & (15) \\ \text{s.t.} & x(E(V)) & = & |V| - |F(V)| - 1 & (16) \\ & x(E(S)) & \leq & |S| - |F(S)| - 1 \quad \forall S \in \mathcal{I}(F) & (17) \\ & x(\delta(v)) & \leq & B_v \quad \forall v \in W & (18) \\ & x_e & \geq & 0 \quad \forall e \in E & (19) \end{array}$$

In the linear program, the constraints from (16)-(17) and (19) are exactly the spanning tree constraints for the graph G/F, the graph formed by contracting each component of F into a singleton vertex. The constraints from (18) are the degree constraints for vertices in W. Hence, from the discussion in Section 3, it follows that we can optimize over LP-MBDCT$(G, \mathcal{B}, W, F)$ using the ellipsoid algorithm in polynomial time.

The algorithm in Figure 4 is an iterative rounding procedure for LP-MBDCT$(G, \mathcal{B}, W, F)$. For clarity of presentation and proof of correctness, we present the algorithm as a recursive procedure.

For the correctness of the MBDCT Algorithm, we shall prove the following key lemma in Section 4.2, which will ensure that the algorithm terminates.

MBDCT Algorithm$(G, \mathcal{B}, W, F)$

1. If F is a spanning tree return $\emptyset$ else let $\hat{F} = \emptyset$
2. Find a basic optimal solution x^* of LP-MBDCT$(G, \mathcal{B}, W, F)$ and remove every edge e with $x^*_e = 0$ from G. Let E^* be the support of x^*.
3. If there exists an edge $e = \{u, v\}$ such that $x^*_e = 1$, then set $\hat{F} \leftarrow \{e\}$, $F \leftarrow F \cup \{e\}$ and $G \leftarrow G \setminus \{e\}$. Also set $B_u \leftarrow B_u - 1$ and $B_v \leftarrow B_v - 1$.
4. If there exists a vertex $w \in W$ such that $deg_{E^*}(w) \leq B_w + 1$, then update $W \leftarrow W \setminus \{w\}$.
5. Return $\hat{F} \bigcup$ MBDCT Algorithm$(G, \mathcal{B}, W, F)$.

Figure 4: MBDCT Algorithm

LEMMA 4.1. *A basic feasible solution x^* of LP-MBDCT$(G, \mathcal{B}, W, F)$ with support E^* must satisfy one of the following.*
(a) There is an edge e with $x^*_e = 1$.
(b) There is a vertex $w \in W$ such that $deg_{E^*}(w) \leq B_w + 1$.

We first prove that Lemma 4.1 implies that the MBDCT Algorithm returns a F-tree with the claimed guarantees.

THEOREM 4.2. *Given a graph G, degree bounds $\mathcal{B}$ for vertices $v \in W$ for some subset $W \subseteq V$, and a forest F, the MBDCT Algorithm returns a F-tree H of cost at most the cost of the optimal solution to LP-MBDCT$(G, \mathcal{B}, W, F)$, and $d_H(v) \leq B_v + 1$ for all $v \in W$.*

PROOF. The proof is by induction on the number of iterations of the algorithm. The base case is trivially true as $H = \emptyset$ is a F-tree of G if F is a spanning tree and H satisfies the degree bounds on each vertex in W. Let x^* be a basic optimal solution to LP-MBDCT$(G, \mathcal{B}, W, F)$ in the first iteration. Suppose, in Step 3, we find an edge $e = (u, v)$ with $x^*_e = 1$. Let $F' = F \cup \{e\}$, $G' = G \setminus \{e\}$ and $\mathcal{B}'$ denote the modified degree bounds as described in Step 3. By the induction hypothesis, the algorithm returns a F'-tree H' of G' whose cost is at most the cost of an optimal solution to LP-MBDCT$(G', \mathcal{B}', W, F')$, and $d_{H'}(w) \leq B'_w + 1$ for all $w \in W$. Consider the F-tree $H = H' \cup \{e\}$ of G. Firstly, observe that x^* restricted to edges of G', say x^*_{res}, is a feasible solution to LP-MBDCT$(G', \mathcal{B}', W, F')$. Therefore,

$$c(H) = c(H') + c_e \leq c \cdot x^*_{res} + c_e = c \cdot x^*$$

as $x^*_e = 1$. Hence, the cost of H is at most the cost of an optimal solution to LP-MBDCT$(G, \mathcal{B}, W, F)$. Now, adding the edge e increases the degree of u and v by 1. As $B_u = B'_u + 1$ and $B_v = B'_v + 1$, we have

$$d_H(u) = d_{H'}(u) + 1 \leq B'_u + 1 + 1 = B_u + 1$$

where the inequality $d_{H'}(u) \leq B'_u + 1$ follows from the induction hypothesis. Similarly, we also have $d_H(v) \leq B_v + 1$. For any other vertex $w \in W \setminus \{u, v\}$, we have

$$deg_H(w) = deg_{H'}(w) \leq B'_w + 1 = B_w + 1$$

where the inequality holds by the induction hypothesis. Hence, the degrees are satisfied within an additive constant of one, as claimed.

Now, suppose we remove a degree constraint for some vertex $w \in W$ in Step 4. Let $W' = W \setminus w$. Clearly, x^* is a feasible solution to LP-MBDCT$(G, \mathcal{B}, W', F)$ since we relaxed the problem by deleting the degree constraint for w. Let x' denote an optimal solution to LP-MBDCT$(G, \mathcal{B}, W', F)$. By the induction hypothesis, the algorithm returns a F-tree H with cost at most $c \cdot x'$ and satisfies that $d_H(v) \leq B_v + 1$ for all $v \in W'$. Clearly, $c \cdot x' \leq c \cdot x^*$, and hence the cost of H is at most the cost of an optimal solution to LP-MBDCT$(G, \mathcal{B}, W, F)$. Moreover, by the induction hypothesis H satisfies the degree bound within additive constant of one for each vertex in $W \setminus \{w\}$. Since $deg_{E^*}(w) \leq B_w + 1$ and $H \subseteq E^*$, we have $d_H(w) \leq B_w + 1$.

In either case we show how to construct a F-tree H with cost at most the cost of an optimal solution to LP-MBDCT$(G, \mathcal{B}, W, F)$, and $d_H(v) \leq B_v + 1$ for all $v \in W$. Lemma 4.1 implies these are the only cases. □

4.1 Characterizing basic solutions

To prove Lemma 4.1, we need a characterization of the basic solutions of LP-MBDCT$(G, \mathcal{B}, W, F)$. The proof of the following lemma is standard but we give it here for completeness.

LEMMA 4.3. *Let x^* be any basic feasible solution of LP-MBDCT$(G, \mathcal{B}, W, F)$ with support E^*. Then there exists a set $T \subseteq W$ and a laminar family $\emptyset \neq \mathcal{L} \subseteq \mathcal{I}(F)$ such that x^* is the unique solution to the following linear system.*

$$\begin{cases} x^*(\delta(v)) = B_v & \forall v \in T \\ x^*(E(S)) = |S| - |F(S)| - 1 & \forall S \in \mathcal{L} \end{cases}$$

Moreover, the vectors $\{\chi_{E(S)} : S \in \mathcal{L}\} \cup \{\chi_{\delta(v)} : v \in T\}$ are linearly independent. Furthermore, $|E^*| = |\mathcal{L}| + |T|$.

PROOF. A basic solution of a linear program is the unique solution of m linearly independent tight constraints, where m denotes the number of variables in the linear program. Let $U = \{v \in W : x^*(\delta(v)) = B_v\}$ and $\mathcal{M} = \{S \subseteq V : \sum_{e \in E(S)} x^*_e = |S| - |F(S)| - 1\}$. For $R, S \in \mathcal{M}$ and $R \cap S \neq \emptyset$, we have that:

$$\begin{aligned} &(|R \cap S| - |F(R \cap S)| - 1) + (|R \cup S| - |F(R \cup S)| - 1) \\ &\geq x^*(E(R \cap S)) + x^*(E(R \cup S)) \\ &\geq x^*(E(R)) + x^*(E(S)) \\ &= |R| - |F(R)| - 1 + |S| - |F(S)| - 1 \\ &= (|R \cap S| - |F(R \cap S)| - 1) + (|R \cup S| - |F(R \cup S)| - 1), \end{aligned}$$

where the last equality holds because $E(F) \cap \delta(R) = \emptyset$ and $E(F) \cap \delta(S) = \emptyset$ by the definition of $\mathcal{I}(F)$. So equality holds everywhere and thus both $R \cap S$ and $R \cup S$ are also in $\mathcal{M}$. This also implies that there are no edges in E between $R \setminus S$ and $S \setminus R$, and hence the linear dependency $\chi_{E(R \cap S)} + \chi_{E(R \cup S)} = \chi_{E(R)} + \chi_{E(S)}$.

Now, from standard uncrossing arguments (as in the proof of Lemma 2.3), it follows that there exists a maximal linearly independent laminar family $\mathcal{L}$ in $\mathcal{M}$ such that the characteristic vectors in $\{\chi_{E(S)} : S \in \mathcal{L}\}$ span all the characteristic vectors in $\{\chi_{E(S)} : S \in \mathcal{M}\}$. Let T be a maximal subset of U such that $\chi_{\delta(v)}$ for $v \in T$ and $\chi_{E(S)}$ for $S \in \mathcal{L}$ are linearly independent. Then, the inequalities corresponding to vertices in T and the inequalities corresponding to sets in $\mathcal{L}$ define a basic solution x^* proving the first claim, satisfying the second claim, and the final claim follows. □

REMARK 4.4. *Another proof of Lemma 4.3 can be obtained by observing that in LP-MBDCT$(G, \mathcal{B}, W, F)$, the constraints from (2)-(3) and (5) correspond to spanning tree constraints of G/F, which is the graph formed by contracting each component of F into a singleton vertex. Let $\mathcal{F} = \{S \in \mathcal{I}(F) : x^*(E(S)) = |S| - |F(S)| - 1\}$. Observe that each $S \in \mathcal{F}$ corresponds to a subset $S' \subseteq V(G/F)$ after we contract each component of F contained in S (observe that $S \in \mathcal{I}(F)$ implies that S does* *not intersect any component of F). Let $\mathcal{F}'$ be the family consisting of subsets of $V(G/F)$ corresponding to subsets in $\mathcal{F}$. From Lemma 2.3, it follows that there is a laminar family $\mathcal{L}' \subseteq \mathcal{F}'$ such that $span(\mathcal{L}') = span(\mathcal{F}')$. Now, uncontracting each supernode inside each subset $S' \in \mathcal{L}'$, we get the desired laminar family $\mathcal{L} \subseteq \mathcal{F}$.*

4.2 A counting argument

We are ready to prove Lemma 4.1. Suppose for sake of contradiction that both (a) and (b) of Lemma 4.1 are not satisfied. Then each vertex $v \in W$ has degree at least 3, and degree of $v \in W$ is exactly 3 only if $B_v = 1$. Now, let $\mathcal{L} \neq \emptyset$ be the laminar family and T be the vertices defining the solution x^* as in Lemma 4.3. As in the proof of Lemma 3.2, we shall derive that $|\mathcal{L}| + |T| < |E^*|$. This contradicts Lemma 4.3 and completes the proof.

We call a vertex v *active* if there is some edge incident at v. Clearly, all vertices in T are active. The laminar family $\mathcal{L}$ defines a directed forest L in which nodes correspond to sets in $\mathcal{L}$ and there exists an edge from set R to set S if R is the smallest set containing S. We call R the *parent* of S and S the *child* of R. A parent-less node is called a *root* and a childless node is called a *leaf*. Given a node R, the *subtree rooted at* R consists of R and all its descendants.

The strategy in the counting argument is similar to that used by Jain [12]. For each active node $v \in V$, we assign one token to v for each edge incident at v. For every edge we have assigned exactly two tokens, and hence the total tokens assigned is exactly $2|E^*|$. We shall redistribute these tokens such that each vertex in T and each subset $S \in \mathcal{L}$ is assigned two tokens, and we are still left with some excess tokens; this will imply $|E^*| > |\mathcal{L}| + |T|$ and contradict Lemma 4.3.

In the initial assignment each active vertex has at least one token, and each vertex in T gets at least three tokens. Vertices in T need two tokens and are assigned at least three tokens; active vertices not in T do not need any tokens but are assigned at least one token. Hence in the initial assignment each active vertex has at least one *excess* token and the following claim follows.

CLAIM 4.5. *If an active vertex v has only one excess token, then either $v \notin T$ and v is of degree one, or $v \in T$ and v is of degree three and $B_v = 1$.*

The following key lemma shows that such a redistribution is possible.

LEMMA 4.6. *For any rooted subtree of the forest $L \neq \emptyset$ with root S, we can distribute the tokens assigned to vertices inside S such that every vertex in $T \cap S$ and every node in the subtree gets at least two tokens and the root S gets at least four tokens.*

PROOF. The proof is by induction on the height of the subtree. First suppose S is a leaf.

1. S contains at least four active vertices, then S can collect at least four tokens by taking one excess token from each active vertex.

2. S contains exactly three active vertices, say $\{u, v, w\}$, then $|E^*(S)| \leq 3$. If any one of the active vertices in S, say u, has two excess tokens, then S can collect four tokens by taking one excess token from each vertex and two excess tokens from u, and we are done. Now suppose that each of $\{u, v, w\}$ has exactly one excess token. Since $x^*(E^*(S)) \geq 1$ and there is no 1-edge, we have $|E^*(S)| \geq 2$.

   Suppose $|E^*(S)| = 2$, say $E^*(S) = \{uv, uw\}$, then this implies that $u \in T$, $B_u = 1$ and u has another neighbor $y \notin S$ (else u would be removed from W in Step 4 of Algorithm 4) . However, $x^*(E^*(S)) = x^*(u,v) + x^*(u,w) = x^*(\delta(u)) - x^*(u,y) = B_u - x^*(u,y) < 1$, a contradiction.

   Hence S contains exactly three edges. This implies that $u, v, w \in T$ and $B_u = B_v = B_w = 1$ by Claim 4.5. Since there are no edges inside a supernode, each of the three active vertices must be in different supernodes. Therefore, $x^*(u,v) + x^*(u,w) + x^*(v,w) = x^*(E^*(S)) \geq 2$, since S contains at least three supernodes. This implies that $\sum_{z \in \{u,v,w\}} x^*(\delta(z)) \geq 4$ which contradicts the fact that degree bound of each of u, v and v is one.

3. S contains at most two active vertices. We show such a case cannot occur. For any $S \in \mathcal{L}$ we have that $x^*(E^*(S)) = k$ for some integer $k > 0$ and since there is no 1-edge, $E^*(S)$ must contain at least two edges. This implies that S contains at least three active vertices.

Now suppose S has at least one child.

1. S has two or more children: By the induction hypothesis, each child has 2 excess tokens, and so S can collect at least 4 tokens by taking the excess tokens.

2. S has only one child: Let the child of S be R. S can take two excess tokens from R by the induction hypothesis. Observe that $S \setminus R$ must contain at least one active vertex as $\chi_{E(S)}$ and $\chi_{E(R)}$ are linearly independent. If $S \setminus R$ has two or more active vertices then we can take one excess token from each and give them to S, and we are done. So suppose $S \setminus R$ has exactly one active vertex, say v. If v has two excess tokens, then we are also done. So assume v has only one excess token. Note that $x^*(E^*(S)) = x^*(E^*(R)) + x^*(\delta(v,R))$, where $\delta(v,R)$ denotes the edges between v and vertices in R. Since $S, R \in \mathcal{L}$, both are tight and $x^*(E^*(S))$ and $x^*(E^*(R))$ are integers, $x^*(\delta(v,R)) \geq 1$. As there is no 1-edge, v is not a degree-1 vertex. Since v has only one excess token, we must have $v \in T$ and $B_v = 1$ by Claim 4.5. Now, $1 = B_v = x^*(\delta(v)) \geq x^*(\delta(v,R)) \geq 1$ and we must have equality throughout and so $\delta(v) = \delta(v,R)$. This implies that $\chi_{E(S)} = \chi_{E(R)} + \chi_{\delta(v,R)} = \chi_{E(R)} + \chi_{\delta(v)}$, which contradicts their linear independence in $\mathcal{L}$.

□

From Lemma 4.6, we obtain that number of tokens is at least $2|T| + 2|\mathcal{L}| + 2$ which shows that $|E^*| > |T| + |\mathcal{L}|$, which contradicts Lemma 4.3. This completes the proof of Lemma 4.1, and hence Theorem 4.2 follows.

5. A ±1 APPROXIMATION ALGORITHM

In this section, we consider an extension of the MBDCT problem in which a degree lower bound A_v and a degree upper bound B_v are given for each vertex v. We present an $(1, A_v - 1, B_v + 1)$-approximation algorithm for the MBDCT problem, where both the degree lower and upper bounds are violated by at most 1. We assume the lower bounds are given on a subset of vertices $U \subseteq V$. Let $\mathcal{A}$ denote the vector of all degree bounds A_v for each $v \in U$. The following is a linear programming relaxation for the MBDCT problem, which is denoted by LP-MBDCT$(G, \mathcal{A}, \mathcal{B}, U, W, F)$.

$$\begin{array}{lrcll} \text{minimize} & c(x) & = & \sum_{e \in E} c_e\, x_e & \\ \text{subject to} & x(E(V)) & = & |V| - |F(V)| - 1 & \\ & x(E(S)) & \leq & |S| - |F(S)| - 1 & \forall\, S \in \mathcal{I}(F) \\ & x(\delta(v)) & \geq & A_v & \forall\, v \in U \\ & x(\delta(v)) & \leq & B_v & \forall\, v \in W \\ & x_e & \geq & 0 & \forall\, e \in E \end{array}$$

Recall that a vertex is active if it has degree at least 1. Notice that if a supernode C has only one active vertex v, we could just contract C into a single vertex c, set $A_c := A_v$ and $B_c := B_v$, and set $c \in U \iff v \in U$, and set $c \in W \iff v \in W$. Henceforth, we call a supernode which is not a single vertex a *nontrivial supernode*. Hence a non-trivial supernode has at least 2 active vertices. The $(1, A_v - 1, B_v + 1)$-approximation algorithm in Figure 5 is an iterative rounding procedure for LP-MBDCT$(G, \mathcal{A}, \mathcal{B}, U, W, F)$.

MBDCT Algorithm2$(G, \mathcal{A}, \mathcal{B}, U, W, F)$

1. If F is a spanning tree then return $\emptyset$ else let $\hat{F} \leftarrow \emptyset$.
2. Find a basic optimal solution x^* of LP-MBDCT$(G, \mathcal{A}, \mathcal{B}, U, W, F)$ and remove every edge e with $x^*_e = 0$ from G.
3. If there exists an edge $e = \{u, v\}$ such that $x^*_e = 1$ then $\hat{F} \leftarrow \{e\}$, $F \leftarrow F \cup \{e\}$ and $G \leftarrow G \setminus \{e\}$. Also update $\mathcal{A}, \mathcal{B}$ by setting $A_u \leftarrow A_u - 1$, $B_u \leftarrow B_u - 1$ and $A_v \leftarrow A_v - 1$, $B_v \leftarrow B_v - 1$.
4. If there exists a vertex $v \in U \cup W$ of degree at most two, then update $U \leftarrow U \setminus \{v\}$ and $W \leftarrow W \setminus \{v\}$.
5. Return $\hat{F} \bigcup$ MBDCT Algorithm2$(G, \mathcal{A}, \mathcal{B}, U, W, F)$.

Figure 5: MBDCT Algorithm 2

For the correctness of the MBDCT Algorithm 2, we shall prove the following key lemma, which will ensure that the algorithm terminates.

LEMMA 5.1. *A basic feasible solution x^* of LP-MBDCT$(G, \mathcal{A}, \mathcal{B}, U, W, F)$ with support E^* must satisfy one of the following.*
(a) There is an edge e such that $x^*_e = 1$.
(b) There is a vertex $v \in U \cup W$ such that $deg_{E^*}(v) = 2$.

In MBDCT Algorithm 2, we only remove a degree constraint on $v \in U \cup W$ if v is of degree 2 and there is no 1-edge. Since there is no 1-edge, we must have $A_v \leq 1$. If $v \in U$, then the worst case is $A_v = 1$ but both edges incident at v are not picked in later iterations. If $v \in W$, then the worst case is $B_v = 1$ but both edges incident at v are picked in later iterations. In either case, the degree bound is off by at most 1. Following the same argument of Theorem 4.2, we have the following extension of Theorem 1.2.

THEOREM 5.2. *There is a polynomial time $(1, A_v - 1, B_v + 1)$-approximation algorithm for the* MINIMUM BOUNDED DEGREE CONNECTING TREE *problem.*

To prove Lemma 5.1, we need a characterization of the basic solutions of LP-MBDCT$(G, \mathcal{A}, \mathcal{B}, U, W, F)$. The proof of the following lemma is the same as the proof of Lemma 4.3.

LEMMA 5.3. *Let x^* be any basic feasible solution of LP-MBDCT$(G, \mathcal{A}, \mathcal{B}, U, W, F)$. Then there exists a set $T_U \subseteq U$, $T_W \subseteq W$ and a laminar family $\emptyset \neq \mathcal{L} \subseteq \mathcal{I}(F)$ such that x^* is the unique solution to the following linear system.*

$$\begin{cases} x^*(\delta(v)) = A_v & \forall v \in T_U \\ x^*(\delta(v)) = B_v & \forall v \in T_W \\ x^*(E(S)) = |S| - |F(S)| - 1 & \forall S \in \mathcal{L} \end{cases}$$

Moreover, the vectors $\{\chi_{E(S)} : S \in \mathcal{L}\} \cup \{\chi_{\delta(v)} : v \in T_U\} \cup \{\chi_{\delta(v)} : v \in T_W\}$ are linearly independent. Furthermore, $|E^*| = |\mathcal{L}| + |T_U| + |T_W|$.

5.1 A counting argument

Now we are ready to prove Lemma 5.1. The set up is very similar to that of Section 4.2. Let $\mathcal{L}$ be the laminar family and $T := T_U \cup T_W$ be the vertices defining the solution x^* as in Lemma 5.3. Suppose that both (a) and (b) of Lemma 5.1 are not satisfied. We shall derive that $|\mathcal{L}| + |T| < |E^*|$, which will contradict Lemma 5.3 and complete the proof.

As before, for each active vertex $v \in V$, we assign one token to v for each edge incident at v. Observe that in the initial assignment each active vertex has at least one excess token, and so a nontrivial supernode has at least two excess tokens. For a vertex v with only one excess token, if $v \notin T$, then v is a degree 1 vertex; if $v \in T$, then v is of degree 3 and $B_v = 1$ or $B_v = 2$.

Suppose every vertex v which is active (and hence has excess tokens) gives all its excess tokens to the supernode it is contained in. We say the number of excess tokens of a supernode is the sum of excess tokens of active vertices in that supernode. Observe that the excess of any supernode is at least one as every supernode has at least one active vertex and each active vertex has at least one excess token.

We call a supernode *special* if its excess is exactly one.

CLAIM 5.4. *A supernode C is special only if it contains exactly one active vertex $v \in T$ and $deg_{E^*}(v) = 3$.*

PROOF. If the supernode C has two or more active vertices then the excess of C is at least two. Hence, it must contain exactly one active vertex with exactly one excess token. Also, there must be at least two edges incident at the supernode as $x^*(\delta(C)) \geq 1$ is a valid inequality. Hence, $deg_{E^*}(C) \geq 2$. If $v \notin T$, then both v and thus C will have at least two excess tokens. This implies $v \in T$ and $deg_{E^*}(v) = 3$. □

We contract a special supernode into a single vertex because it contains only one active vertex. Hence, the only special supernodes are singleton vertices in T with degree exactly three.

The main difference from the proof in Section 4.2 is the existence of special vertices with degree bounds equal to 2, for which we need to revise the induction hypothesis because some node $S \in \mathcal{L}$ may now only get three tokens. The following definition gives a characterization of those sets which only get three tokens.

DEFINITION 5.5. *A set $S \neq V$ is* special *if:*

1. $|\delta(S)| = 3$;
2. $x^*(\delta(S)) = 1$ *or* $x^*(\delta(S)) = 2$;
3. $\chi_{\delta(S)}$ *is a linear combination of the characteristic vectors of its descendants (including possibly $\chi_{E(S)}$) and the characteristic vectors $\chi_{\delta(v)}$ of $v \in S \cap T$;*

Observe that special supernodes satisfy all the above properties. Intuitively, a special set has the same properties as a special supernode. The following lemma will complete the proof of Lemma 5.1, and hence Theorem 5.2.

LEMMA 5.6. *For any rooted subtree of the forest $L \neq \emptyset$ with the root S, we can distribute the tokens assigned to vertices inside S such that every vertex in $T \cap S$ and every node in the subtree gets at least two tokens and the root S gets at least three tokens. Moreover, the root S gets exactly three tokens only if S is a special set or $S = V$.*

PROOF. First we prove some claims needed for the lemma.

CLAIM 5.7. *If $S \neq V$, then $|\delta(S)| \geq 2$.*

PROOF. Since $S \neq V$, $x^*(\delta(S)) \geq 1$ is a valid inequality of the LP. As there is no 1-edge, $|\delta(S)| \geq 2$. □

Let the root be set S. We say a supernode C is a *member* of S if $C \subseteq S$ but $C \not\subseteq R$ for any child R of S. We also say a child R of S is a *member* of S. We call a member R of S *special*, if R is a special supernode (in which the supernode is a singleton vertex in T with degree three from Claim 5.4) or if R is a special set. In either case (whether the member is a supernode or set), a member has exactly one excess token only if the member is special. Special members also satisfy all the properties in Definition 5.5.

Recall that $E(S)$ denotes the set of edges with both endpoints in S. We denote by $D(S)$ the set of edges with endpoints in *different* members of S.

CLAIM 5.8. *If $S \in \mathcal{L}$ has r members then $x^*(D(S)) = r - 1$.*

PROOF. For every member R of S, we have

$$x^*(E(R)) = |R| - |F(R)| - 1,$$

since either $R \in \mathcal{L}$, or R is a supernode in which case both LHS and RHS are zero. As $S \in \mathcal{L}$, we have

$$x^*(E(S)) = |S| - |F(S)| - 1$$

Now observe that every edge of $F(S)$ must be contained in $F(R)$ for some member R of S. Hence, we have the following, in which the sum is over R that are members of S.

$$x^*(D(S)) = x^*(E(S)) - \sum_R x^*(E(R))$$

$$= |S| - |S \cap F| - 1 - \sum_R (|R| - |F(R)| - 1)$$

$$= (|S| - \sum_R |R|) + \sum_R |F(R)| - |F(S)| + \sum_R 1 - 1$$

$$= (\sum_R 1) - 1 = r - 1$$

because $|S| = \sum_R |R|$ and $|F(S)| = \sum_R |F(R)|$. □

CLAIM 5.9. *Suppose a set $S \neq V$ contains exactly three special members R_1, R_2, R_3 and $|D(S)| \geq 3$. Then S is a special set.*

PROOF. Note that $|\delta(S)| = |\delta(R_1)| + |\delta(R_2)| + |\delta(R_3)| - 2|D(S)| = 3 + 3 + 3 - 2|D(S)| = 9 - 2|D(S)|$. Since $S \neq V$, we have $|\delta(S)| \geq 2$ by Claim 5.7. As $|D(S)| \geq 3$, the only possibility is that $|D(S)| = 3$ and $|\delta(S)| = 3$, which satisfies the first property of a special set. Also, we have $x^*(\delta(S)) = x^*(\delta(R_1)) + x^*(\delta(R_2)) + x^*(\delta(R_3)) - 2x^*(D(S))$. As each term on the RHS is an integer, it follows that $x^*(\delta(S))$ is an integer. Moreover, as we do not have a 1-edge, $x^*(\delta(S)) < |\delta(S)| = 3$ and thus $x^*(\delta(S))$ is either equal to 1 or 2, and so the second property of a special set is satisfied. Finally, note that $\chi_{\delta(S)} = \chi_{\delta(R_1)} + \chi_{\delta(R_2)} + \chi_{\delta(R_3)} + \chi_{E(R_1)} + \chi_{E(R_2)} + \chi_{E(R_3)} - 2\chi_{E(S)}$. Here, the vector $\chi_{E(R_i)}$ will be the zero vector if R_i is a special vertex. Since R_1, R_2, R_3 satisfy the third property of a special member, S satisfies the third property of a special set. □

The proof of Lemma 5.6 is by induction on the height of the subtree. In the base case, each member has at least one excess token and exactly one excess token when the member is special. Consider the following cases for the induction step.

1. S has at least four members. Each member has an excess of at least one. Therefore S can collect at least four tokens by taking one excess token from each.

2. S has exactly three members. If any member has at least two excess tokens, then S can collect four tokens, and we are done. Else each member has only one excess token and thus, by the induction hypothesis, is special. If $S = V$, then S can collect three tokens, and this is enough since V is the root of the laminar family. Else, we have $x^*(D(S)) = 2$ from Claim 5.8. Because there is no 1-edge, we must have $|D(S)| > x^*(D(S)) = 2$. Now, it follows from Claim 5.9 that S is special and it only requires three tokens.

3. S contains exactly two members R_1, R_2. If both R_1, R_2 have at least two excess tokens, then S can collect four tokens, and we are done. Else, one of the members has exactly one excess token say R_1. Hence, R_1 is special by the induction hypothesis. We now show a contradiction to the independence of tight constraints defining x^*, and hence this case would not happen.

   Since S contains two members, Claim 5.8 implies $x^*(D(S)) = 1$. There is no 1-edge, therefore we have $|D(S)| = |\delta(R_1, R_2)| \geq 2$. Also, R_1 is special and thus $|\delta(R_1)| = 3$. We claim $\delta(R_1, R_2) = \delta(R_1)$. If not, then let $e = \delta(R_1) \setminus \delta(R_1, R_2)$. Then

   $$x_e^* = x^*(\delta(R_1)) - x^*(\delta(R_1, R_2)) = x^*(\delta(R_1)) - x^*(D(S)).$$

   But $x^*(\delta(R_1))$ is an integer as R_1 is special and $x^*(D(S)) = 1$. Therefore, x_e^* is an integer which is a contradiction. Thus $\delta(R_1, R_2) = \delta(R_1)$. But then

   $$\chi_{E(S)} = \chi_{E(R_1)} + \chi_{\delta(R_1)} + \chi_{E(R_2)}$$

   if R_2 is a set or

   $$\chi_{E(S)} = \chi_{E(R_1)} + \chi_{\delta(R_1)}$$

   if R_2 is supernode. R_1 is special implies that $\chi_{\delta(R_1)}$ is a linear combination of the characteristic vectors of its descendants and the characteristic vectors $\{\chi_{\delta(v)} \colon v \in R_1 \cap T\}$. Hence, in either case $\chi_{E(S)}$ is spanned by $\chi_{E(R)}$ for $R \in \mathcal{L} \setminus \{S\}$ and $\chi_{\delta(v)}$ for $v \in S \cap T$ which is a contradiction to the inclusion of S in $\mathcal{L}$.

This completes the proof of Lemma 5.6 and Theorem 5.2. □

6. CONCLUDING REMARKS AND OPEN QUESTIONS

In this paper we extend the iterative rounding framework to obtain the best possible guarantee for the MBDST problem. A closely related problem is the well studied travelling salesperson problem (TSP). The sub-tour elimination relaxation for TSP is very similar to the LP relaxation for the MBDST problem. Indeed our techniques can be used to give the following polyhedral result: Any solution to the sub-tour elimination polytope can be written as a convex combination of 1-trees each of maximum degree three and average degree two, improving on a similar result of Goemans [10]. Here, a 1-tree is a tree on $V \setminus v$ along with any two edges incident at vertex v. A natural open question is whether the techniques used here can be used to obtain better approximation algorithm for the TSP problem.

Acknowledgements

The collaboration started when the authors visited Microsoft Research in Redmond; we are very grateful to the Theory Group for their hospitality. The first author would like to thank R. Ravi for introducing him to the MBDST problem and innumerable discussions over the past few years. We also thank R. Ravi for help in the preparation of the paper.

7. 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, pp. 26-39, 2005.

[2] K. Chaudhuri, S. Rao, S. Riesenfeld, and K. Talwar. *Push relabel and an improved approximation algorithm for the bounded-degree mst problem* , In Proceedings of 33th International Colloquium on Automata, Languages and Programming, 2006.

[3] William H. Cunningham, *Testing membership in matroid polyhedra,* Journal of Combinatorial Theory, Series B 36(2): 161-188, 1984.

[4] J. Cheriyan, S. Vempala and A. Vetta, *Network design via iterative rounding of setpair relaxations,* Combinatorica 26(3) pp. 255-275, 2006.

[5] G. Cornuejols, D. Naddef and J. Fonlupt *The Traveling Salesman Problem on a Graph and some related Integer Polyhedra*, Mathematical Programming, 33, pp. 1-27, 1985.

[6] J. Edmonds, *Matroids and the Greedy Algorithm.* Mathematical Programming, 1:125–136, 1971.

[7] L. Fleischer, K. Jain and D.P. Williamson, *Iterative rounding 2-approximation algorithms for minimum-cost vertex connectivity problems.*, J. Comput. Syst. Sci. 72:5, 2006.

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

[9] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co., New York, NY, USA, 1979.

[10] M.X. Goemans, *Minimum Bounded-Degree Spanning Trees,* Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science, 273–282, 2006.

[11] D.S. Hochbaum, *Approximation algorithms for NP-hard problems*, PWS Publishing Co., Boston, MA, USA, 1997.

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

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

[14] J. Könemann and R. Ravi, *Primal-Dual meets local search: approximating MSTs with nonuniform degree bounds*, SIAM J. on Computing, 34(3):763-773, 2005.

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

[16] T. Magnanti, L. Wolsey, *Optimal trees*, Network Models, Handbook in Operations Research and Management Science, M.O. Ball et al ed., Amsterdam, North-Holland, pp: 503-615, 1995.

[17] V. Melkonian and E. Tardos, *Algorithms for a Network Design Problem with Crossing Supermodular Demands*, Networks, 43(4):256-265, 2004.

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

[19] 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, 2006.

[20] A. Schrijver, Combinatorial Optimization - Polyhedra and Efficiency, Springer-Verlag, New York, 2005.

[21] V. Vazirani, Approximation Algorithms, Springer, 2001.