Math. Program., Ser. B manuscript No.
(will be inserted by the editor)

A Unified Algorithm for Degree Bounded Survivable Network Design

Lap Chi Lau · **Hong Zhou**

Received: date / Accepted: date

Abstract We present an approximation algorithm for the minimum bounded degree Steiner network problem that returns a Steiner network of cost at most two times the optimal and the degree on each vertex v is at most $\min\{b_v + 3r_{\max}, 2b_v + 2\}$, where $r_{\max}$ is the maximum connectivity requirement and b_v is the given degree bound on v. This unifies, simplifies, and improves the previous results for this problem.

Keywords Approximation Algorithm · Network Design · Minimum Degree Bounded Steiner Network

Mathematics Subject Classification (2000) 68W40 · 68Q25 · 05C85

1 Introduction

In the *minimum bounded degree Steiner network* problem, we are given an undirected graph $G = (V,E)$, a cost c_e on each edge $e \in E$, a degree bound b_v on each vertex $v \in V$, and a connectivity requirement r_{uv} for each pair of vertices $u,v \in V$. A subgraph H of G is called a *Steiner network* if there are at least r_{uv} edge-disjoint paths from u to v in H for all $u,v \in V$. The task of the minimum bounded degree Steiner network problem is to find a Steiner network H with minimum total cost such that $d_H(v) \leq b_v$ for each $v \in V$. This is a general problem of interest to algorithm design, computer networks, graph theory, and operations research.

Lap Chi Lau
Department of Computer Science and Engineering, The Chinese University of Hong Kong
Shatin, Hong Kong
Tel.: +852-39434263
Fax.: +852-26035024
E-mail: chi@cse.cuhk.edu.hk

Hong Zhou
Department of Computer Science and Engineering, The Chinese University of Hong Kong
Shatin, Hong Kong
E-mail: hzhou@cse.cuhk.edu.hk

It is NP-hard to determine whether there is a Steiner network satisfying all the degree bounds, even if we do not consider the cost of the Steiner network, as the Hamiltonian cycle problem is a special case. Thus, researchers focus on designing bicriteria approximation algorithms for the problem that minimize both the total cost and the degree violation. We say an algoithm is an $(\alpha, f(b_v))$-approximation algorithm for the minimum bounded degree Steiner network problem if it returns a Steiner network H of cost at most $\alpha \cdot \text{opt}$ and $d_H(v) \leq f(b_v)$ for each $v \in V$, where opt is the cost of an optimal Steiner network that satisfies all the degree bounds.

The first bicriteria approximation algorithm for this problem is a $(2, 2b_v + 3)$-approximation algorithm by Lau, Naor, Salavatipour, and Singh [12], and it was improved to $(2, 2b_v + 2)$ by Louis and Vishnoi [15]. There are also bicriteria approximation algorithms with additive violation on the degrees in terms of the maximum connectivity requirement. Let $r_{\max} = \max_{u,v}\{r_{u,v}\}$. Lau and Singh [14] gave a $(2, b_v + 6r_{\max} + 3)$-approximation algorithm for the problem, and a $(2, b_v + 3)$-approximation algorithm in the special case when $r_{\max} = 1$. The special case when $r_{\max} = 1$ is known as the *minimum bounded degree Steiner forest* problem.

The main result of this paper is the following theorem.

Theorem 1 *There is a polynomial time algorithm for the minimum bounded degree Steiner network problem that returns a Steiner network H of cost at most* 2opt *and* $deg_H(v) \leq \min\{b_v + 3r_{max}, 2b_v + 2\}$ *for all v.*

Theorem 1 improves the $(2, b_v + 6r_{\max} + 3)$ result in [14] when $r_{\max} \geq 2$ and recovers the $(2, b_v + 3)$ result in [14] for the minimum bounded degree Steiner forest problem. Besides, it achieves the $(2, 2b_v + 2)$ result in [15] simultaneously, while previously there was no such guarantee[1]. Furthermore, both our algorithm and its analysis are simpler[2] as we will discuss in Section 2. We believe that our result unifies what can be achieved using existing techniques. We show an example where our algorithm fails to give a $(2, b_v + 2)$-approximation algorithm for the minimum bounded degree Steiner forest problem in Section 6.

1.1 Related Work

Jain [9] introduced the iterative rounding method to give a 2-approximation algorithm for the minimum Steiner network problem, improving on a line of research that applied primal-dual methods to these problems. Later, the iterative rounding method has been applied to obtain the best known approximation algorithms for network design problems for element-connectivity [4, 3], vertex-connectivity [3, 2], and directed edge-connectivity [7].

The iterative relaxation method was introduced in [12] to adapt Jain's method to degree bounded network design problems, which are well-studied especially in the special case of spanning trees [6, 8]. Later, this method has been applied to obtain

[1] For instance, it was not known how to combine the results in [15, 14] to obtain a $(2, \min\{b_v + 6r_{\max} + 3, 2b_v + 2\})$-approximation algorithm for the problem.

[2] In particular, the analysis of the $(2, b_v + 3)$ result is significantly simpler than that in [14].

the best known approximation algorithms for the degree bounded network design problem, including spanning trees [17,1,11], Steiner networks [12,14,15], directed edge-connectivity [12,1], element-connectivity and vertex-connectivity [10,16,5]. See [13] for a survey on this approach.

2 Technical Overview

Since this work is tightly connected to previous work, we give a high-level overview to describe the previous work and highlight where the improvement comes from.

2.1 Iterative Rounding and Relaxation

All the previous results on this problem are based on the iterative rounding method introduced by Jain [9] for the minimum Steiner network problem. This method is based on analyzing the extreme point solutions to a linear programming relaxation for the problem. Let us first formulate the linear programming relaxation for the minimum bounded degree Steiner network problem. For a subset $S \subseteq V$, we let $\delta(S)$ be the set of edges with one endpoint in S and one endpoint in $V - S$ in the graph and let $d(S) := |\delta(S)|$. In the linear program, there is one variable x_e for each edge, where the intended value is one if this edge is used in the solution and zero if this edge is not used. For a subset of edges $E' \subseteq E$, we write $x(E') = \sum_{e \in E'} x_e$. For a subset of vertices $S \subseteq V$, we define $f(S) := \max_{u \in S, v \notin S}\{r_{uv}\}$ to be the maximum connectivity requirement crossing S. To satisfy the connectivity requirement, we should have $x(\delta(S)) \geq f(S)$ for each $S \subseteq V$. The following is a linear programming relaxation for the minimum bounded degree Steiner network problem. It has exponentially many constraints, but there is a polynomial time separation oracle to determine whether a solution is feasible or not, and thus it can be solved in polynomial time by the ellipsoid method.

$$
\begin{aligned}
\text{(LP)} \quad \text{minimize} \quad & \sum_{e \in E} c_e x_e \\
\text{subject to} \quad & x(\delta(S)) \geq f(S) \quad && \forall S \subset V \\
& x(\delta(v)) \leq b_v && \forall v \in V \\
& x_e \geq 0 && \forall e \in E
\end{aligned}
$$

When there are no degree constraints, Jain [9] proved that there exists an edge e with $x_e \geq \frac{1}{2}$ in any extreme point solution to the above linear program. We call such an edge a *heavy* edge. He used this to obtain an *iterative rounding* algorithm for the minimum Steiner network problem, by repeatedly picking a heavy edge and recomputing an optimal extreme point solution to the residual problem. When there are degree constraints, Lau et al. [12] showed that either there is a heavy edge or there is a degree constrained vertex with at most four nonzero edges incident to it. They then introduced an extra relaxation step to remove the degree constraint on such a vertex in the latter case, leading to a $(2, 2b_v + 3)$-approximation algorithm for the minimum bounded degree Steiner network problem.

Roughly speaking, all the later improvements are based on proving the existence of a heavy edge with additional properties. To improve the degree violation, Louis and Vishnoi [15] proved that in any extreme point solution either there is an edge of integral value (zero or one), or a vertex v with at most $2b_v + 2$ edges incident to it, or a heavy edge *with no endpoint having a degree bound at most one*. They showed that using this iteratively would imply a $(2, 2b_v + 2)$-approximation algorithm for the problem. Note that in the above algorithms, after we pick a heavy edge, we need to decrease the degree bound by half in order to achieve the guarantee on the degree violation, and thus they have to consider a slightly more general problem where the degree bounds are half-integral and some subtle issue arose as we will discuss later.

To obtain additive violation on the degree bounds, Lau and Singh [14] proved that in any extreme point solution either there is an edge of integral value, or a vertex v with at most four edges incident to it, or a heavy edge *between two vertices with fractional degree at most* $6r_{\max}$, where the last condition guarantees that the degree violation is bounded when we picked edges with value at least half. For the minimum bounded degree Steiner forest problem, they proved that in any extreme point solution either there is an integral edge, or a vertex v with at most $b_v + 3$ edges incident to it, or a heavy edge *with no degree constraint on its endpoints*. They showed that these would lead to a $(2, b_v + 6r_{\max} + 3)$-approximation algorithm for the Steiner network problem and a $(2, b_v + 3)$-approximation algorithm for the Steiner forest problem. The algorithm for Steiner forest is simpler, as it just removes the degree constraint on a vertex when it has at most $b_v + 3$ edges, and does not need to update the degree constraint to a half-integral value, as it only picks edges with value at least half when both endpoints have no degree constraints.

Our algorithm is very similar to that for the Steiner forest problem in [14] (see Algorithm 1). We prove that either there is an edge of integral value, or there is a vertex with at most $\min\{b_v + 3r_{\max}, 2b_v + 2\}$ nonzero edges incident to it, or there is a heavy edge with *no degree constraints on its endpoints*. The resulting algorithm is quite simple, in the first case we delete an edge when $x_e = 0$ or pick an edge when $x_e = 1$, in the second case we remove the degree constraint on that vertex, and in the final case we pick such a heavy edge. Note that we only update the degree constraints when we pick an edge with $x_e = 1$, and thus we can maintain the invariant that the degree bounds are integral, and this will simplify the analysis for the $2b_v + 2$ bound.

2.2 Analysis

To analyze the extreme point solutions, an uncrossing technique is used to show that the extreme point solutions are defined by a set of constraints with a special structure. A function $f : 2^V \to \mathbb{R}$ is *skew supermodular* if for any $X, Y \subseteq V$ either $f(X) + f(Y) \leq f(X \cup Y) + f(X \cap Y)$ or $f(X) + f(Y) \leq f(X - Y) + f(Y - X)$. It is known that the function f defined by the connectivity requirements is a skew supermodular function. For a set $S \subseteq V$, the corresponding constraint $x(\delta(S)) \geq f(S)$ defines a vector in $\mathbb{R}^{|E|}$: the vector has a one corresponding to each edge $e \in \delta(S)$ and a zero otherwise. We call this vector the *characteristic vector* of $\delta(S)$ and denote it by $\chi_{\delta(S)}$. A family of sets $\mathscr{L}$ is *laminar* if $X, Y \in \mathscr{L}$ implies that either $X \cap Y = \emptyset$, or $X \subseteq Y$, or $Y \subseteq X$. Using the assumption that f is skew supermodular, it follows from standard uncrossing technique that any extreme point solution of (LP) is characterized by a laminar family of tight constraints.

Lemma 1 ([12], Lemma 2.3) *Suppose that the connectivity requirement function f of (LP) is skew supermodular. Let x be an extreme point solution to (LP) such that $0 < x_e < 1$ for all edges $e \in E$. Then there exist a laminar family $\mathscr{L}$ of sets and set $T \subseteq V$ such that x is the unique solution to*

$$\{x(\delta(S)) = f(S) \mid S \in \mathscr{L}\} \cup \{x(\delta(v)) = b_v \mid v \in T\}$$

that satisfies the following properties:

1. *The characteristic vectors $\chi_{\delta(S)}$ for $S \in \mathscr{L}$ and $\chi_{\delta(v)}$ for $v \in T$ are linearly independent.*
2. *$|E| = |\mathscr{L}| + |T|$.*

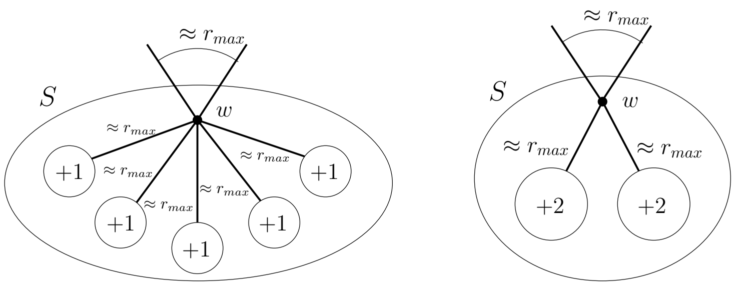
The structure of a laminar set family $\mathscr{L}$ can be seen as a forest in which nodes correspond to sets in $\mathscr{L}$ and there is an edge from set R to set S if R is the smallest set containing S. We call R the *parent* of S and S is a *child* of R. A set without any parent is a *root* and a set without any child is a *leaf*. The *subtree rooted at* a set S consists of S and all its descendants.

Lau et al. [12], following Jain [9], used this forest structure in a counting argument to prove that either there exists an edge with integral value, or a vertex with degree at most four, or an edge with value at least $\frac{1}{2}$. First, each edge is assigned two tokens, one to each endpoint, for a total of $2|E|$ tokens. Assuming none of the conditions hold, i.e. $0 < x_e < \frac{1}{2}$ for each edge and every vertex is of degree at least five, then it can be shown that the tokens can be redistributed such that each member of $\mathscr{L}$ and each vertex in T get at least two tokens, and there are still some extra tokens left. This implies that there are more than $2|\mathscr{L}| + 2|T|$ tokens, contradicting property 2 of Lemma 1. The redistribution is done inductively using the following lemma.

Lemma 2 ([12], Lemma 2.4) *For any subtree of $\mathscr{L}$ rooted at node S, we can reassign tokens collected from child nodes of S and endpoints owned by S such that each vertex in $T \cap S$ gets at least two tokens, each node in the subtree gets at least two tokens, and the root S gets at least three tokens. Moreover, S gets exactly three tokens only if* $\text{coreq}(S) = \frac{1}{2}$, *where* $\text{coreq}(S) := \sum_{e \in \delta(S)} (\frac{1}{2} - x_e)$.

The proof in [12] is almost the same as Jain's proof, but with the presence of degree constraints. Note that each vertex with degree constraint gets at least five tokens in the initial token assignment. Intuitively, we can think of each degree constraint as a singleton set (a leaf set in the laminar family), and thus having five tokens is more than enough for Jain's proof to go through. And one may think that it is already enough if every degree constraint gets at least four tokens to satisfy the induction hypothesis in Lemma 2, and this would imply a $(2, 2b_v + 2)$-approximation algorithm. Unfortunately, the subtle point here is that the degree bounds are half-integral, but for Jain's proof to work they need to be integral. To overcome this problem, Louis and Vishnoi [15] needed to modify the algorithm and the analysis to obtain a $(2, 2b_v + 2)$-approximation algorithm.

The new idea in the Steiner network algorithm in [14] is to only pick heavy edges when both endpoints are of low degree. In the analysis, with the presence of heavy edges that are not allowed to be picked (when some endpoint is of "high" degree), there could be some sets S with $d(S) = 2$ and $x(\delta(S)) = 1$, and thus the counting argument as above would not work in the base case for those sets. For the same induction hypothesis in Lemma 2 to work, a new rule is added to the initial token assignment: if (w,v) is a heavy edge with $b_w \geq 6r_{\max}$ and v is not degree constrained, then v gets two tokens from the edge (w,v) while w gets none. The counting argument would work in the base case with this new rule, but w may not receive any token for the induction step to work. The assumption $b_w \geq 6r_{\max}$ is used to ensure that w can *get back* the tokens in the induction step. To illustrate this, consider a worst case scenario in Figure 1(a). In the figure, there is a degree constrained vertex w where all its incident edges are heavy, and we need to collect two tokens for w and four tokens for S. Each child contributes only one token but "consumes" the degree bound of w by $r_{\max}$. This is where the assumption that $b_w \geq 6r_{\max}$ is used to guarantee that S has at least five children, so that each can contribute at least one token for w and S (and use some additional argument to collect one more token).



(a) For the algorithm in [14]   (b) For our algorithm

Fig. 1: Worst cases for counting arguments.

In this paper, we slightly change the algorithm to remove any vertex with degree at most $\min\{b_v + 3r_{\max}, 2b_v + 2\}$. We used the same initial token assignment rule as in [14] and the same induction hypothesis in Lemma 2 for the counting argument. First, using a simple argument (see Lemma 6), we show that any vertex with degree at least $2b_v + 3$ can receive at least four tokens, and this allows the induction to work and recovers the $(2, 2b_v + 2)$ result by Louis and Vishnoi. As mentioned before, this is possible because our algorithm maintains the invariant that all the degree bounds are integral.

Our improvement to $b_v + 3r_{\max}$ comes from the concept of the *gap* of the degree of a set S, defined as $d(S) - x(\delta(S))$. To illustrate this, consider the same scenario in Figure 1. In the new algorithm, the vertex w with degree bound b_w but having more than $b_w + 3r_{\max}$ edges incident to it has a gap of $3r_{\max}$ on its degree. The main observation is that the heavy edges from a child having only three tokens (one extra token) can only contribute $\frac{1}{2}$ to the gap of w (see Lemma 7), while a child having at least four tokens could contribute $r_{\max}$ to the gap of w. This observation basically allows us to rule out children with only three tokens in the worst case scenarios of the counting argument, as it contributes one token but only consumes $\frac{1}{2}$ of the gap. This allows us to assume that each child can contribute two tokens instead of one token, and this reduces the degree violation from $6r_{\max}$ to $3r_{\max}$ (see Figure 1(b)), with some additional arguments. An additional advantage of our algorithm is that we also avoid the additive term $+3$ in the previous algorithm [14]. The proof for $r_{\max} \geq 2$ is quite short (see Section 4.1). The proof for $r_{\max} = 1$ still has some case analysis (see Section 4.2), but is considerably shorter than that in [14], as a complicated induction hypothesis was used in [14] that caused many more case analyses.

3 Algorithm and Analysis

In the following, let W be the set of vertices with degree constraints.

Algorithm 1: Minimum Bounded Degree Steiner Network

1. Initialization: $H = (V, \emptyset), W \leftarrow V, f'(S) \leftarrow f(S)$ for all $S \subseteq V$.
2. **while** *H is not a Steiner network* **do**
   - (a) Compute an optimal extreme point solution x to (LP) and remove all edges e with $x_e = 0$.
   - (b) For each vertex $v \in W$ with degree at most $\min\{b_v + 3r_{max}, 2b_v + 2\}$, remove v from W.
   - (c) For each edge $e = (u, v)$ with $x_e = 1$, add e to H and remove e from G, and decrease b_u, b_v by one.
   - (d) For each edge $e = (u, v)$ with $x_e \geq \frac{1}{2}$ and $u, v \notin W$, add e to H and remove e from G.
   - (e) For each $S \subset V$, set $f'(S) \leftarrow f(S) - d_H(S)$.
3. Return H.

Given that f is initially a skew supermodular function, it is known that f' in any later iteration is still a skew supermodular function [9]. So, the residual LP in any iteration is still in the original form and it has a polynomial time separation oracle [9]. Assuming the algorithm always makes progress in each iteration, then we can prove Theorem 1 by a standard inductive argument as in [12, 13]. It remains to prove the following lemma to complete the proof of Theorem 1.

Lemma 3 *Let x be an extreme point solution to (LP) and W be the set of vertices with degree constraints. Then at least one of the following is true.*

1. *There exists a vertex $v \in W$ with $d(v) \leq \min\{b_v + 3r_{max}, 2b_v + 2\}$.*
2. *There exists an edge e with $x_e = 0$ or $x_e = 1$.*
3. *There exists an edge $e = (u, v)$ with $x_e \geq \frac{1}{2}$ and $u, v \notin W$.*

We prove Lemma 3 by contradiction. Assuming none of the three conditions holds, then we have

1. $d(v) \geq \min\{b_v + 3r_{max} + 1, 2b_v + 3\}$ for $v \in W$,
2. $0 < x_e < 1$ for $e \in E$,

3. if $x_{uv} \geq \frac{1}{2}$ then either u or v is in W.

We will use a token counting argument to derive a contradiction with Lemma 1. Let $\mathscr{L}$ be the laminar family and T be the set of vertices with tight degree constraints as defined in Lemma 1. In the token counting argument, we first assign two tokens to each edge, for a total of $2|E|$ tokens. Then, using the assumptions above, we show that these tokens can be redistributed such that each member in $\mathscr{L}$ and each vertex in T gets at least two tokens and there are some tokens left, but this contradicts with $|E| = |\mathscr{L}| + |T|$ from Lemma 1.

Initial token assignment rule: Each edge receives two tokens. If $e = (u, v)$ is a heavy edge with $u \in W$ and $v \notin W$, then v gets two tokens from e and u gets no token. For every other edge e, each endpoint of e gets one token.

We will redistribute the tokens inductively using the forest structure of the laminar family $\mathscr{L}$. We need some definitions to state the induction hypothesis. We say a vertex v is *owned by* a set $S \in \mathscr{L}$ if S is the smallest set in $\mathscr{L}$ that contains v. Given an extreme point solution x, we say an edge e is a *heavy edge* if $x_e \geq 1/2$, otherwise we say e is a *light edge*. Let $\delta^h(S) = \{e \in \delta(S), x_e \geq 1/2\}$ and $\delta^l(S) = \{e \in \delta(S), x_e < 1/2\}$ be the set of heavy edges and light edges in $\delta(S)$ respectively. The *corequirement* of a set S is defined as

$$\mathrm{coreq}(S) = \sum_{e \in \delta^h(S)} (1 - x_e) + \sum_{e \in \delta^l(S)} (1/2 - x_e) = |\delta^h(S)| + \frac{|\delta^l(S)|}{2} - x(\delta(S)).$$

We will prove the following lemma which shows that the tokens can be redistributed to obtain a contradiction, proving Lemma 3.

Lemma 4 *For any subtree of $\mathscr{L}$ rooted at node S, we can reassign tokens collected from child nodes of S and endpoints owned by S such that each vertex in $T \cap S$ gets at least two tokens, each node in the subtree gets at least two tokens, and the root S gets at least three tokens. Moreover, S gets exactly three tokens only if* $\mathrm{coreq}(S) = \frac{1}{2}$.

To prove Lemma 4, the most challenging part is handling the case when S owns some vertices in W. We show that in such case S can get at least four tokens, which enables us to go through the induction. In the induction step, we assume that the induction hypothesis holds for each child of S. We say a child of S is a *rich* child if it gets at least four tokens, and say a child of S is a *poor* child if it gets exactly three tokens. Note that a child only needs two tokens and thus has some *excess* tokens, i.e., each rich child of S has at least two excess tokens and each poor child of S has one excess token. The following lemma is the technical core of this paper.

Lemma 5 *Let $S \in \mathscr{L}$. Suppose that the induction hypothesis in Lemma 4 holds for each child of S and S owns $k' \geq 1$ vertices in W. Assume k of these k' vertices are in T, where $0 \leq k \leq k'$, then the number of excess tokens from the child nodes of S, plus the number of tokens collected from endpoints owned by S is at least* $2k + 4$.

Lemma 5 handles the case when S owns some vertices in W, to guarantee that S gets at least four tokens and each vertex in T owned by S gets two tokens for the induction hypothesis. The remaining cases can be handled exactly as in the proof in [14]. In Section 5, we show the proof of Lemma 4 assuming Lemma 5, following the proof in [14] for completeness.

We will prove Lemma 5 in the following section. We present the proof of Lemma 5 when $r_{\max} \geq 2$ in Section 4.1, which improves the result in [14] about Steiner networks. In Section 4.2, we show the proof of Lemma 5 when $r_{\max} = 1$, which recovers the Steiner forest result in [14] with a considerably simpler proof.

4 Proof of Lemma 5

Before we assume $r_{\max} \geq 2$, let's prove two useful lemmas. The first lemma takes care of those vertices $w \in W$ with $d(w) \geq 2b_w + 3$.

Lemma 6 *If $d(w) \geq 2b_w + 3$ for $w \in W$, then w receives at least four tokens in the initial token assignment.*

Proof Assume that there are h heavy edges incident and l light edges incident to w. If $l \geq 4$, then w receives at least four tokens in the initial token assignment. Suppose to the contrary that $l \leq 3$. Then $h \geq 2b_w$ as $d(w) = h + l \geq 2b_w + 3$. If $h > 2b_w$, then $x(\delta(w)) \geq \frac{h}{2} > b_w$, contradicting that x is a feasible solution to (LP). Otherwise, if $h = 2b_w$, since each light edge has positive value, we have $x(\delta(w)) > \frac{h}{2} = b_w$, again contradicting that x is a feasible solution to (LP). □

Lemma 6 says that any vertex $w \in W$ with $d(w) \geq 2b_w + 3$ gets at least four tokens in the initial assignment. Together with the fact that b_w is an integer, then any $w \in W$ with $d(w) \geq 2b_w + 3$ is a singleton set $\{w\}$ with $x(\delta(w))$ integral and has at least four tokens, and thus it behaves the same as a rich child in the proof of Lemma 5. Recall the assumption that $d(w) \geq \min\{b_w + 3r_{max} + 1, 2b_w + 3\}$, henceforth, we can assume that $b_w + 3r_{\max} + 1 < 2b_w + 3$ for each $w \in W$, and thus

$$b_w > 3r_{\max} - 2 \text{ and } d(w) \geq b_w + 3r_{\max} + 1 \text{ for } w \in W. \tag{1}$$

The second lemma takes care of the poor children using the concept of gap of the degree of a set. For an edge e with $0 < x_e < 1$, let $y_e = 1 - x_e \in (0,1)$ be the *gap of* e. For a subset of edges $E' \subseteq E$, let $y(E') := \sum_{e \in E'} y_e$.

Lemma 7 *Suppose the induction hypothesis in Lemma 4 holds for each child of $S \in \mathscr{L}$. Let $R \in \mathscr{L}$ be a poor child of S. Then*

$$y(\delta^h(R)) \leq \frac{1}{2}.$$

Proof Note that

$$\frac{1}{2} = \text{coreq}(R) = \sum_{e \in \delta^h(R)} (1 - x_e) + \sum_{e \in \delta^l(R)} (\frac{1}{2} - x_e) = \sum_{e \in \delta^h(R)} y_e + \sum_{e \in \delta^l(R)} (\frac{1}{2} - x_e).$$

Since $1/2 - x_e > 0$ for each light edge e, we have $\sum_{e \in \delta^h(R)} y_e \leq 1/2$. □

Let $w_1,\dots,w_{k'}$ be the vertices in W owned by S. Suppose the first k vertices $w_1,\dots,w_k$ are in T. The main idea in the proof of Lemma 5 is to consider $Y := \sum_{i=1}^{k'} y(\delta(w_i))$. It follows from (1) that $y(\delta(w_i)) = d(w_i) - x(\delta(w_i)) = d(w_i) - b_{w_i} \geq 3r_{\max}+1$ and thus

$$Y \geq (3r_{\max}+1)k'.$$

By Lemma 7, the heavy edges from a poor child can only contribute very little to this sum, and this will allow us to rule out the existence of a poor child in S.

4.1 Proof of Lemma 5 when $r_{\max} \geq 2$

Now, we are ready to prove Lemma 5 when $r_{\max} \geq 2$.

First, we count the number of tokens that S can collect. Consider the edges in $F := \cup_{i=1}^{k'} \delta(w_i)$. Let $F_2 \subseteq F$ be the subset of edges of F with both endpoints owned by S, and $F_1 := F - F_2$ be the subset of edges of F with exactly one endpoint owned by S. Note that each edge in F_2 can contribute two tokens to S, regardless of whether it is heavy or light. Let ℓ be the number of light edges in F_1. Then each such edge can contribute one token to S. Let γ be the number of rich children of S and ρ be the number of poor children of S. By the induction hypothesis, the children can contribute at least $2\gamma+\rho$ tokens to S. Therefore, S can collect at least $2\gamma+\rho+\ell+2|F_2|$ tokens from its children and the endpoints that it owns. If $2\gamma+\rho+\ell+2|F_2| \geq 2k+4$, then we are done. So we assume to the contrary that

$$2\gamma+\rho+\ell+2|F_2| \leq 2k+3. \tag{2}$$

Next, we consider the contribution to Y. Each endpoint of an edge e can contribute strictly less than one to Y, as $y_e = 1 - x_e < 1$ for each edge. So, the edges in F_2 and the light edges in F_1 can contribute strictly less than $2|F_2|+\ell$ to Y. It remains to count the contribution from the heavy edges in F_1. The heavy edges from a rich child R can contribute at most $r_{\max}$ to Y, because each heavy edge can contribute at most $\frac{1}{2}$ to Y and $|\delta^h(R)| \leq 2r_{\max}$ as $x(\delta(R)) \leq r_{\max}$. The heavy edges from a poor child can contribute at most $\frac{1}{2}$ to Y by Lemma 7. Finally, the heavy edges in $F_1 \cap \delta^h(S)$ can contribute at most $r_{\max}$ to Y, because $|\delta^h(S)| \leq 2r_{\max}$ as $x(\delta(S)) \leq r_{\max}$. These count all the contributions to Y. Therefore, we must have

$$(3r_{\max}+1)k' \leq Y \leq (\gamma+1)\cdot r_{\max} + \frac{1}{2}\rho + \ell + 2|F_2|. \tag{3}$$

To satisfy (2) as an equality, we must have $\rho+\ell+2|F_2| \geq 1$ since γ is an integer. If $\ell+2|F_2| \geq 1$, then the second inequality in (3) is a strict inequality. Otherwise, if $\rho \geq 1$ or (2) is not an equality, by plugging in (2), we also have the following strict inequality for any nonnegative integer r_{max}.

$$\begin{aligned}(3r_{\max}+1)k' &< (\gamma+1)\cdot r_{\max} + 2k + 3 - 2\gamma \\ &\Longrightarrow (3k'-1)r_{\max} < 2k - k' + 3 + \gamma(r_{\max}-2).\end{aligned} \tag{4}$$

As $\gamma \leq k+1$ by (2) and $k \leq k'$, when $r_{\max} \geq 2$ we have

$$(3k'-1)r_{\max} < 2k - k' + 3 + (k+1)(r_{\max}-2) \implies r_{\max}(3k'-k-2) < 1-k'$$
$$\implies r_{\max}(2k'-2) < 1-k',$$

which is impossible for $k' \geq 1$. This contradiction shows that (2) cannot hold, and thus S can collect $2k+4$ tokens as claimed.

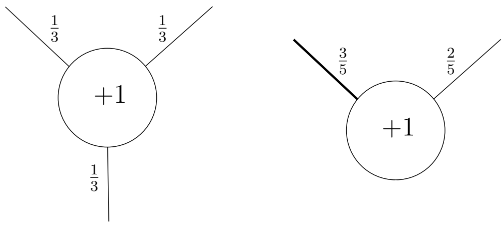
4.2 Proof of Lemma 5 when $r_{\max} = 1$

Until (4), we have not used the assumption that $r_{\max} \geq 2$. Plugging in $r_{\max} = 1$ in (4), we get that $4k' - 2k < 4 - \gamma$. Since $k \leq k'$, we have $2k' < 4 - \gamma$, which implies that $k' = 1$ (recall that $k' \geq 1$). Then, k can only be 0 or 1. If $k = 0$, then $4k' - 2k < 4 - \gamma$ becomes $4 < 4 - \gamma$, which is impossible. Thus, we only need to consider the case of $k' = k = 1$. Henceforth, we assume that S only owns one vertex $w \in T$.

To prove Lemma 5, we need to show that S can collect six tokens from the excess tokens of its children and the endpoints that it owns.

The following lemma characterizes the poor children of S when $r_{max} = 1$.

Lemma 8 *Let R be a poor child of S. When $r_{max} = 1$, either $\delta(R)$ has three light edges (type 1), or $\delta(R)$ has one heavy edge and one light edge (type 2).*



(a) Type 1 (b) Type 2

Fig. 2: Poor child when $r_{\max} = 1$.

Proof Let h be the number of heavy edges in $\delta(R)$, l be the number of light edges in $\delta(R)$. Since R is a poor child, $\frac{1}{2} = \text{coreq}(R) = h + l/2 - x(\delta(R))$. As $r_{max} = 1$, we must have $x(\delta(R)) = 1$ as $R \in \mathscr{L}$, and thus $2h + l = 3$. Therefore, either $h = 1$ and $l = 1$, or $h = 0$ and $l = 3$. □

The next lemma counts the number of tokens that S can collect. We introduce some definitions for the lemma. For a subset $R \subseteq S$, we let $d(w,R)$ be the number of edges from w to a vertex in R. Note that each light edge in $\delta(w)$ contributes one token to S.

- Let $O := \{v \in S - w \mid S \text{ owns } v\}$ be the set of vertices in $S - w$ that S owns. Let o be the number of edges with one endpoint in O. For each such edge e, if the other endpoint of e is w, then e contributes two tokens to S (regardless of whether e is heavy or light), otherwise e contributes one token to S.
- Let ρ^+ be the number of type 1 poor children of S plus the number of type 2 poor children of S with no heavy edge to w. Each such poor child R and the light edges from R to w contribute at least $d(w,R)+1$ tokens to S, as R has one excess token and no heavy edge to w. Any poor child R and the light edges from R to w contribute at least $d(w,R)$ tokens to S, as there is at most one heavy edge from a poor child.
- Let γ^+ be the number of rich children of S with at most one heavy edge to w. Each such child R and the light edges from R to w contribute at least $d(w,R)+1$ tokens to S, as R has two excess tokens and at most one heavy edge to w. Any rich child R and the light edges from R to w contribute at least $d(w,R)$ tokens to S, as there are at most two heavy edges from a rich child.
- Let h^- be the number of heavy edges in $\delta(w) \cap \delta(S)$. Note that $h^- \leq 2$ as $x(\delta(S)) \leq r_{\max} = 1$. Each light edge in $\delta(w) \cap \delta(S)$ contributes one token to S.

Lemma 9 *When $r_{\max} = 1$ and w is the only vertex in T that S owns, then S can collect at least $d(w) - h^- + o + \rho^+ + \gamma^+$ tokens.*

Proof Using the above definitions, we get the lemma by summing the contributions from the edges in $\delta(w)$, the excess tokens of the children of S, and the endpoints that S owns. □

Lemma 9 will be used to show that we can collect six tokens for S in most cases, and the remaining cases will be removed by using the linear independence of the characteristic vectors in Lemma 1.

Recall from (1) that $b_w > 3r_{\max} - 2 = 1$ and $d(w) \geq b_w + 3r_{\max} + 1 = b_w + 4$. We will do a case analysis based on the value of b_w. By Lemma 9, S can collect at least $d(w) - h^- \geq b_w + 2$ tokens. When $b_w \geq 4$, we are done. It remains to consider the case when $b_w = 3$ or $b_w = 2$.

4.2.1 $b_w = 3$

We have $d(w) \geq b_w + 4 = 7$. By Lemma 9, S can collect at least $d(w) - h^- + o + \rho^+ + \gamma^+$ tokens. If this is less than six, the only possibility left is $d(w) = 7, h^- = 2, o = 0, \rho^+ = 0$, and $\gamma^+ = 0$, but we will show that this is impossible.

Since $b_w \leq 3$ (for the same argument to be reused when $b_w = 2$) and $d(w) \geq 7$, there are at most five heavy edges incident to w, and thus at least two light edges e_1, e_2 incident to w. Since $h^- = 2$, $e_1, e_2 \notin \delta(S)$. Since $o = 0$, the other endpoint of e_1, e_2 must be in a child of S. Since $\gamma^+ = 0$, the other endpoint of e_1, e_2 must be in a poor child of S. Since $\rho^+ = 0$, the only possiblility left is that there are two type 2 poor children R_1, R_2 of S, $e_1 \in \delta(R_1)$, $e_2 \in \delta(R_2)$, $\delta(R_1) \subseteq \delta(w)$ and $\delta(R_2) \subseteq \delta(w)$. As $d(w) = 7$, there is still one edge f incident to w. But then $3 \geq b_w = x(\delta(w)) = x(\delta(w) \cap \delta(S)) + x(\delta(w) \cap \delta(R_1)) + x(\delta(w) \cap \delta(R_2)) + x_f = x(\delta(S)) + x(\delta(R_1)) + x(\delta(R_2)) + x_f = 3 + x_f$, implying that $x_f \leq 0$, a contradiction.

4.2.2 $b_w = 2$

We have $d(w) \geq b_w + 4 = 6$. If $d(w) \geq 7$, then we can use the same argument as in the case of $b_w = 3$ to collect enough tokens for S. Hence, we assume that $d(w) = 6$. Note that there are at most three heavy edges incident to w. By Lemma 9, S can collect at least $d(w) - h^- + o + \rho^+ + \gamma^+$ tokens. We are done if $h^- = 0$. In the following, we consider the cases when $h^- = 2$ or $h^- = 1$.

$h^- = 2$*:* By Lemma 9, if S does not have six tokens yet, then we must have $o + \rho^+ + \gamma^+ \leq 1$. Since $h^- = 2$, there is at most one heavy edge left in $\delta(w) - \delta(S)$ and $\delta(S) \subseteq \delta(w)$. This implies that S has at most two children, as otherwise $\rho^+ + \gamma^+ \geq 2$, because each child not having a heavy edge to w contributes at least one to $\rho^+ + \gamma^+$.

- Suppose S contains no children. Then S owns at least four endpoints as $d(w) = 6$, and thus S can collect enough tokens.
- Suppose S has only one child R. Recall that $o + \rho^+ + \gamma^+ \leq 1$ for S has not collected six tokens yet. If $o = 1$, then $|\delta(w) \cap \delta(R)| \geq 3$ as $d(w) = 6$, but then either R is poor or rich would imply that $\rho^+ + \gamma^+ \geq 1$, a contradiction. If $o = 0$, since $\delta(S) \subseteq \delta(w)$ as $h^- = 2$, we must have $\delta(R) = \delta(w) \cap \delta(R)$, but this implies that $\chi_{\delta(S)} = \chi_{\delta(w)} - \chi_{\delta(R)}$, a contradiction to the linear independence of the characteristic vectors.
- Suppose S has exactly two children R_1 and R_2. Since there is at most one heavy edge left in $\delta(w) - \delta(S)$, only a type 2 poor child R with a heavy edge to w will not contribute to $\rho^+ + \gamma^+$. So, for $o + \rho^+ + \gamma^+ \leq 1$, we must have a type 2 poor child R_1 with one heavy edge to w. Hence, there is no heavy edge from R_2 to w, and thus R_2 must contribute one to $\rho^+ + \gamma^+$, and we must have $o = 0$ for $o + \rho^+ + \gamma^+ \leq 1$. The remaining situation is shown in Figure 3. Since $x(\delta(R_1)) = x(\delta(R_2)) = x(\delta(w) - \delta(S)) = 1$, we must have $a + c = a + b = b + c = 1$ and thus $a = b = c = \frac{1}{2}$. However, as R_1 is a type 2 poor child, it implies that both edges are heavy, a contradiction.

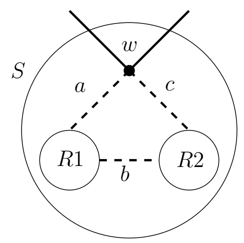
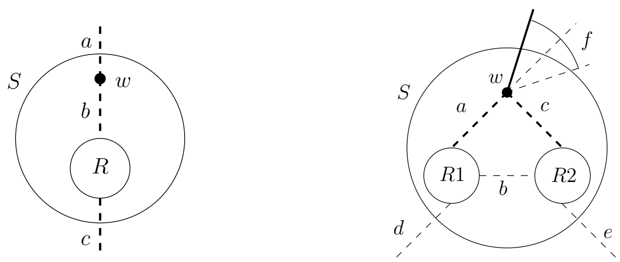


Fig. 3: $b_w = 2$, $h^- = 2$, S has two children, and $o = 0$.

$h^- = 1$*:* By Lemma 9, S can collect at least $d(w) - h^- + o + \rho^+ + \gamma^+$ tokens. The only possibility left is $d(w) = 6, h^- = 1, o = 0, \rho^+ = 0$, and $\gamma^+ = 0$. Since $b_w = 2$, there are at most two heavy edges left in $\delta(w) - \delta(S)$. This implies that S has at most two children, as otherwise $\rho^+ + \gamma^+ \geq 1$.

- Suppose S has no children. Then, since $o = 0$, we must have $\chi_{\delta(S)} = \chi_{\delta(w)}$, contradicting the linear independence of the characteristic vectors.
- Suppose S contains only one child R as shown in Figure 4(a). Since $x(\delta(R)) = x(\delta(S)) = 1$ and $x(\delta(w)) = 2$ and $o = 0$, we have $b+c = a+c = 1$ and $a+b = 2$ and thus $a = b = 1$ and $c = 0$. But this implies that $\chi_{\delta(S)} = \chi_{\delta(w)} - \chi_{\delta(R)}$, a contradiction to the linear independence of the characteristic vectors.
- Suppose S has exactly two children R_1 and R_2. Since there are at most two heavy edges left in $\delta(w) - \delta(S)$, for $\rho^+ = \gamma^+ = 0$, we must have that both R_1 and R_2 are type 2 poor children and each has a heavy edge to w. Since $o = 0$, the situation is as shown in Figure 4(b).
  - Suppose $b = 0$. By the connectivity constraints and the degree constraint, we have $a+b+d = 1, b+c+e = 1, d+e+f = 1$ and $a+c+f = 2$, which implies that $d+e = f = \frac{1}{2}$. So, there is only one edge in $\delta(w) \cap \delta(S)$. Since R_1 and R_2 are type 2 poor children, it follows that $d(w) = |\delta(w) \cap \delta(S)| + |\delta(w) \cap \delta(R_1)| + |\delta(w) \cap \delta(R_2)| \leq 1+2+2 = 5$, contradicting the assumption that $d(w) = 6$.
  - Suppose $b \neq 0$. Then $d = e = 0$, as R_1 and R_2 are type 2 poor children. By the connectivity constraints and the degree constraint, we have $f = 1, a+b = 1, b+c = 1$ and $a+c+f = 2$, which imply that $a = b = c = \frac{1}{2}$, but this contradicts with the fact that type 2 poor children have only one heavy edge.



(a) S has one child R (b) S has two children

Fig. 4: $b_w = 2$, $h^- = 1$ and $o = 0$.

We have considered all the cases and this completes the proof of Lemma 5.

5 Proof of Lemma 4

For completeness, in this section we present exactly the same proof as in [14] for Lemma 4.

For the base case, let $S \in \mathscr{L}$ be a leaf node. If $S \cap W \neq \emptyset$, then by Lemma 5, the leaf node S and vertices in T owned by S get enough tokens. Now, assume $S \cap W = \emptyset$, then S can get at least $|\delta(S)|$ tokens from the vertices owned by S. Since, we assume there is no 1-edge and $x(\delta(S))$ is an integer, $|\delta(S)| \geq 2$. If $|\delta(S)| \geq 4$, then S gets at least four tokens. If $|\delta(S)| = 3$ and $\delta(S)$ contains a heavy edge, then S also gets four tokens, since the heavy edge will give two tokens to S in the initial token assignment. If $\delta(S)$ does not contain heavy edge, then S receives three tokens and $\mathrm{coreq}(S) = \frac{1}{2}$. If $|\delta(S)| = 2$, then at least one edge is a heavy edge. If both of them are heavy edges, then S gets four tokens. If only one of them is heavy, then S gets three tokens and $\mathrm{coreq}(S) = \frac{1}{2}$. Thus, we finished the proof of base case.

Proof of the induction step: The presence of heavy edges with $x_e \geq \frac{1}{2}$ introduces some difficulties in carrying out the inductive argument in Jain [9]. We need to prove some lemmas which work with the new notion of corequirement and the presence of heavy edges.

For any set S, let $wdeg(\delta(S)) = |\delta^l(S)| + 2|\delta^h(S)|$ be the *weighted degree* of S. This definition is in keeping with the idea of regarding each edge with $x_e \geq \frac{1}{2}$ as two parallel edges. Observe that for any $v \notin W$, it receives exactly $wdeg(v)$ tokens in the initial assignment as it gets one token for each edge and two tokens for all heavy edges incident to it. S can take all the tokens for all the vertices it owns which are not in W. We call these the *tokens owned by* S. Let $G' = (V, E')$ be the graph formed by replacing each heavy edge e by two edges e' and e'' such that $x_{e'} = x_{e''} = x_e/2$. Observe that

$$\mathrm{coreq}(S) = \sum_{e \in \delta^l(S) \cap E} (\frac{1}{2} - x_e) + \sum_{e \in \delta^h(S) \cap E} (1 - x_e) = \sum_{e \in \delta(S) \cap E'} (\frac{1}{2} - x_e),$$

and $wdeg(\delta(S)) = |\delta'(S)|$ where $\delta'(S) = \{e \in E' : e \in \delta(S)\}$. Clearly, $\mathrm{coreq}(S)$ is integral or *semi-integral* (half-integral but not integral) depending on whether $|\delta'(S)|$ is even or odd. We first prove the same technical lemma as in Jain [9] and Lau and Singh [14] with the new definition of corequirement and weighted degree.

Claim Let S be a set in $\mathscr{L}$ which owns α tokens and has β children where $\alpha + \beta = 3$, and furthermore S does not own any vertex of W. If each child R of S (if any) has $\mathrm{coreq}(R) = \frac{1}{2}$, then $\mathrm{coreq}(S) = \frac{1}{2}$.

Proof Since each child R of S is of $\mathrm{coreq}(R) = \frac{1}{2}$, this implies that $|\delta'(R)|$ is odd. Note that we assume S does not own any vertex of W, thus each token owned by S is incident to exactly one edge in E'. We call a child of S or a token owned by S an *item* of S, then each item of S is incident with odd number of edges. Since there are $\alpha + \beta = 3$ items in S, the sum of number of edges incident to each item is odd. While each edge between two items is counted twice, it implies that the number of edges incident to only one item of S is odd. Since the set of such edges is exactly $\delta'(S)$, $|\delta'(S)|$ is odd. Hence, the corequirement of S is semi-integral. Then we show that $\mathrm{coreq}(S) < 3/2$ to finish the proof of this Claim. Note that

$$\mathrm{coreq}(S) = \sum_{e \in \delta'(S)} (\frac{1}{2} - x_e) \leq \sum_R \mathrm{coreq}(R) + \sum_e (\frac{1}{2} - x_e),$$

where the first sum is over all β children of S, the second sum is over all edges in $\delta'(S)$ for which S owns a token. Since $\alpha + \beta = 3$, there are a total of three terms in the sum. Since any term in the first sum is $\frac{1}{2}$ and in the second sum is strictly less then $\frac{1}{2}$, if $\alpha > 0$, we have $\text{coreq}(S) < 3/2$, we finished the proof. Therefore, assume $\alpha = 0$, S does not own any token. In this case, edges incident to children of S cannot all be incident to S, otherwise it will cause linear dependence. Therefore, we have $\text{coreq}(S) < \sum_R \text{coreq}(R) = 3/2$, completed the proof. □

Now, we are ready to prove the induction step. Let $S \in \mathscr{L}$ be a non-leaf node with at least one child. If S owns some vertices in W, then by Lemma 5, the node S and all vertices in T owned by S get enough tokens. Therefore, we assume S owns no vertex in W.

- S has at least four children. Then S gets at least four tokens by induction.
- S has exactly three children. If any child of S is a rich child or S owns a token then S can get four tokens. Otherwise, each of the three children of S is a poor child and has corequirement of $\frac{1}{2}$ and S owns no token. Then by the above Claim, $\text{coreq}(S) = \frac{1}{2}$, three tokens are enough.
- S has exactly two children R_1 and R_2. If both of them are rich children, then S gets at least four tokens. Otherwise, without loss of generality, assume R_1 is a poor child with $\text{coreq}(R_1) = \frac{1}{2}$. We claim that S must own a token. Suppose to the contrary that S does not own any token. Then, if there are α edges between R_1 and R_2, we have
$$|\delta'(S)| = |\delta'(R_1)| + |\delta'(R_2)| - 2\alpha$$
As $\text{coreq}(R_1) = \frac{1}{2}$, $|\delta'(R_1)|$ is odd and hence $|\delta'(S)|$ and $|\delta'(R_2)|$ have different parity (see Figure 5), and hence S and R_2 have different corequirement. Since there is no token owned by S, the corequirement of S and R_2 can differ by at most $\text{coreq}(R_1) = \frac{1}{2}$. Because $\chi_{\delta'(S)} \neq \chi_{\delta'(R_1)} + \chi_{\delta'(R_2)}$, there must be an edge between R_1 and R_2, which implies $\text{coreq}(S) < \text{coreq}(R_2) + \frac{1}{2}$. Similarly, since $\chi_{\delta'(R_2)} \neq \chi_{\delta'(R_1)} + \chi_{\delta'(S)}$, there must be an edge in $\delta'(S) \cap \delta'(R_1)$, which implies $\text{coreq}(R_2) < \text{coreq}(S) + \frac{1}{2}$. This shows that $\text{coreq}(S) = \text{coreq}(R_2)$, which is a contradiction. Thus, S owns at least one token.

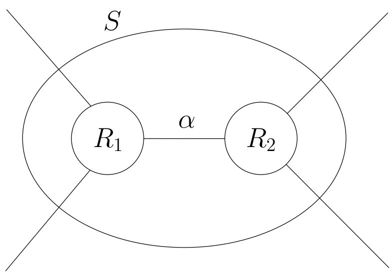


Fig. 5: $|\delta'(S)|$ and $|\delta'(R_2)|$ have different parity

If S owns at least two tokens or R_2 is a rich child, then S can get four tokens. Otherwise, S owns only one token and both R_1, R_2 are poor children, by the above Claim, $\text{coreq}(S) = \frac{1}{2}$, three tokens are enough for S.

- S has exactly one child R. Since $\chi_{\delta'(S)}$ and $\chi_{\delta'(R)}$ are linearly independent, S must own at least one token. However, both $x(\delta'(S)) = f'(S)$ and $x(\delta'(R)) = f'(R)$ are integers and there is no edge of integral value, this simply implies that S cannot own exactly one token, and thus S owns at least two tokens. If S owns three tokens or R is a rich child, then S can get four tokens. Otherwise, S owns exactly two tokens and exactly one child R with $\text{coreq}(R) = \frac{1}{2}$. Then, by the above Claim, $\text{coreq}(S) = \frac{1}{2}$, S only needs three tokens.

Thus we complete the proof of the induction step and also Lemma 4. □

To prove Lemma 3, it only remains to show that we have extra tokens left. If V does not own any vertex in T, then the top nodes of $\mathscr{L}$ already give us extra tokens by Lemma 4. If V owns some vertices in T, then with similar arguments as in Lemma 5 (by taking $S = V$, we have $\delta(S) = \emptyset$, which only makes the argument easier), we will get the desired extra tokens. Thus, we can finish the proof of Lemma 3.

6 A hard example for the algorithm

A natural question is that, for the minimum bounded degree Steiner forest problem, whether we can improve our algorithm further by only relaxing vertices with degree at most $b_v + 2$. This would imply a $(2, b_v + 2)$-approximation algorithm for the problem, matching the known integrality gap for this linear program [12].

In the example shown in Figure 6, some vertices have a degree bound equal to two, but there are five edges incident to these vertices. This is an extreme point solution to (LP) as the characteristic vectors are linearly independent. Our algorithm will get stuck in this example, and it is not clear to us how to modify the algorithm to deal with it. We believe that some new ideas are needed to obtain a $(2, b_v + 2)$-approximation algorithm for this problem.

Acknowledgement

We thank the anonymous reviewers for comments that improve the presentation of the paper. This research is partially supported by HK RGC grant 2150701.

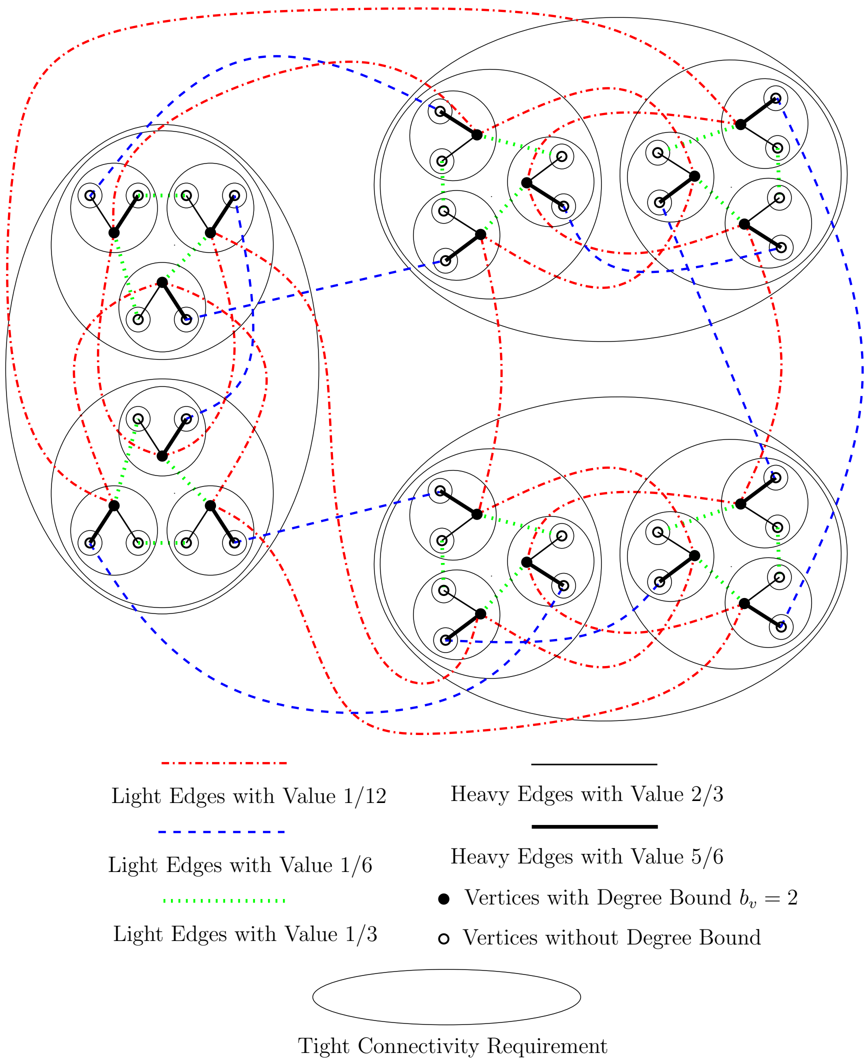


Fig. 6: A hard example

References

1. Bansal, N., Khandekar, R., Nagarajan, V.,: Additive guarantees for degree bounded directed network design. SIAM Journal on Computing 29, 1413–1431 (2009)
2. Cheriyan, J., Vegh, L.: Approximating minimum-cost k-node connected subgraphs via independence-free graphs. In: Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS) (2013)
3. Cheriyan, J., Vempala, S., Vetta, A.: Network design via iterative rounding of setpair relaxations. Combinatorica 26(3), 255–275, (2006)

4. Fleischer, L., Jain, K., Williamson, D.P.: Iterative rounding 2-approximation algorithms for minimum-cost vertex connectivity problems. J. Comput. Syst. Sci. 72(5), 838–867 (2006)
5. Fukunaga, T., Ravi, R.: Iterative rounding approximation algorithms for degree-bounded node-connectivity network design. In: Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS), 263–272 (2012)
6. Fürer, M., Raghavachari, B.: Approximating the minimum-degree Steiner tree to within one of optimal. J. of Algorithms 17(3), 409–423 (1994)
7. Gabow, H.: On the L_∞-norm of extreme points for crossing supermodular directed network LPs. In: Proceedings of the 11th International Conference on Integer Programming and Combinatorial Optimization (IPCO), 392–406 (2005)
8. Goemans, M.X.: Minimum Bounded-Degree Spanning Trees. In: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science, 273–282 (2006)
9. Jain, K.: A factor 2 approximation algorithm for the generalized Steiner network problem. Combinatorica 21(1), 39–60 (2001)
10. Khandekar, R., Kortsarz, G., Nutov, Z.: On some network design problems with degree constraints. Journal of Computer and System Sciences 79(5), 725–736 (2013)
11. Király, T., Lau, L.C., Singh, M.: Degree Bounded Matroids and Submodular Flows. In: Proceedings of the 13th International Conference on Integer Programming and Combinatorial Optimization (IPCO), 259-272 (2008)
12. Lau, L.C., Naor, S., Salavatipour, M., Singh, M.: Survivable network design with degree or order constraints. SIAM Journal on Computing 39(3), 1062–1087 (2009)
13. Lau, L.C., Ravi, R., Singh, M.: Iterative Methods in Combinatorial Optimization. Cambridge University Press, New York (2011)
14. Lau, L.C., Singh, M.: Additive approximation for bounded degree survivable network design. SIAM Journal on Computing 42(6), 2217-2242 (2013)
15. Louis, A., Vishnoi, N.K.: Improved algorithm for degree bounded survivable network design problem. In: Proceedings of the 12th Scandinavian Symposium and Workshops on Algorithm Theory, 408–419 (2010)
16. Nutov, Z.: Degree-constrained node-connectivity. In: Proceedings of the 10th Latin American Symposium on Theoretical Informatics (LATIN), 582–593 (2012)
17. Singh, M., Lau, L.C.: Approximating minimum bounded degree spanning trees to within one of optimal. In: Proceedings of the 39th ACM Symposium on Theory of Computing (STOC), 661–670 (2007)