A simplified 1.5-approximation algorithm for augmenting edge-connectivity of a graph from 1 to 2

Guy Kortsarz†
Rutgers University, Camden
guyk@crab.rutgers.edu

Zeev Nutov
The Open University of Israel
nutov@openu.ac.il

Abstract

The Tree Augmentation Problem (TAP) is given a connected graph $G = (V, E)$ and an edge set $E$ on $V$ disjoint to $E$, find a minimum size subset of edges $F \subseteq E$ such that $(V, E \cup F)$ is 2-edge-connected. In [5] and [6] a 1.8 and 1.5 approximation were given for the problem. The proof of the 1.5 was cut into two papers, as our proof then was very complex and very long. In the current paper we present a substantially simplified, substantially different, self contained and much shorter proof of the 1.5 ratio. The new ideas in this paper may find future applications. Our paper also corrects a typo from the IPL paper [6] that gave a wrong definition of locking trees.

1 Introduction

1.1 Problem definition and our result

A graph (possibly with parallel edges) is $k$-edge-connected if there are $k$ pairwise edge-disjoint paths between every pair of its nodes. We study the following fundamental connectivity augmentation problem: given a connected undirected graph $G = (V, E)$ and a set of additional edges (called “links”) $E$ on $V$ disjoint to $E$, find a minimum size edge set $F \subseteq E$ so that $G + F = (V, E \cup F)$ is 2-edge-connected. The 2-edge-connected components of the given graph $G$ form a tree. It follows that by contracting these components, one may assume that $G$ is a tree. Hence, our problem is:

<table>
<thead>
<tr>
<th>Tree Augmentation Problem (TAP)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Instance:</strong> A tree $T = (V, \mathcal{E})$ and a set of links $E$ on $V$ disjoint to $\mathcal{E}$.</td>
</tr>
<tr>
<td><strong>Objective:</strong> Find a minimum size subset $F \subseteq E$ of edges such that $T \cup F$ is 2-edge-connected.</td>
</tr>
</tbody>
</table>

†Partially supported by NSF grant number 1218620.
TAP is sometimes posed as the problem of covering a laminar family (see e.g. [1]). Namely, given a laminar family $\mathcal{E}$ on a groundset $V$, and an edge set $E$ on $V$, find a minimum size $F \subseteq E$ such that for every $S \in \mathcal{E}$, there is an edge in $F$ with one endpoint in $S$ and the other in $V - S$. TAP is also equivalent to the problem of augmenting the edge-connectivity from $k$ to $k + 1$ for any odd $k$; this is since the family of minimum cuts of a $k$-connected graph with $k$ odd is laminar.

The first 2-approximation for TAP was given by Frederickson & Jájá [7]. TAP is APX-hard even if the set $E$ of links forms a cycle on the leaves of $T$ [1]. Achieving ratio below 2 was posed by Khuller [10] as one of the main open problems in connectivity augmentation. Nagamochi [14] presented a $(1.875 + \varepsilon)$-approximation scheme for TAP, but his algorithm and analysis are very complicated. In the conference version [4] we sketched a proof of a 1.5-approximation algorithm for the problem. The journal version was cut into two papers [5] and [6]. The need for two papers was because our the proof was too complex and too long. We give a substantially simplified, substantially different, self contained and substantially shorter proof for the 1.5 ratio. The new ideas in our analysis may have further applications. Its important to note that there is a typo in the IPL paper regarding the definition of locking trees (see below). We correct it here.

**Theorem 1.1** TAP admits a 1.5-approximation algorithm.

Several ideas introduced by Nagamochi in [14] are used in this paper (e.g., minimally leaf-closed trees and some parts of the lower bound). Key to our analysis is a credit scheme for using the different parts of the lower bound. This credit scheme has a “static” component and a “dynamic” component. The static credit is computed at the beginning of the algorithm and is distributed to different parts of the graph. The dynamic credit is available only after the algorithm “reveals” it by proving that the optimum solution had to be larger than the initial static estimate. We believe that this technique has merit of its own and may be useful for other connectivity problems.

1.2 Related work

There are few 2-approximation algorithms for weighted TAP. The first algorithm, by Frederickson and Jájá [7] was simplified later by Khuller and Thurimella [11]. These algorithms are based on constructing a directed graph and computing a minimum weight arborescence. The primal-dual algorithm of [8] is a combinatorial 2-approximation algorithm for the problem. The iterative rounding algorithm of Jain [9] is an LP-based 2-approximation algorithms. If the LP solution is half integral a 4/3 ratio is given for weighted TAP in [1]. However, there are instances in which the basic solutions are not half integral. The approximation ratio of 2 for all these algorithms is tight even for TAP. Breaking the ratio of 2 for weighted TAP is a major open problem in approximation theory. In [3] an $f(D)n^{O(1)}$ time, $1 + \ln 2$ ratio is give fo weighted TAP, with $D$ the diameter of the tree.

A 1.42 ratio is given for the special case of TAP with all links being leaf to leaf [13]. In addition, the same paper gives a $5/3$ approximation ratio for the same problem, so that the ratio is with
respect to the natural LP value. A slightly better than 1.8 approximation algorithm for TAP with respect to the value of a Positive Semi Definite program, was given recently by Cheriyan and Gao, using the Lasserre lift and project method. The simple LP for unweighted TAP was shown to have integrality gap $3/2$ in [2].

For many problems related to TAP, see the surveys of [10] and [12].

2 Preliminaries

Let $T = (V, E)$ be a tree. For $u, v \in V$ let $(u, v) \in E$ denote the edge in $T$ and $uv$ the link in $E$ between $u$ and $v$. Let $P(uv)$ denote the path between $u$ and $v$ in $T$. A link $uv$ covers all the edges along the path $P(uv)$. We designate a node $r$ of $T$ as the root, and refer to the pair $T, r$ as a rooted tree (we do not mention the root when it is clear from the context). The choice of $r$ defines a partial order on $V$: $u$ is a descendant of $v$ (or $v$ is an ancestor of $u$) if $v$ belongs to $P(ru)$; if, in addition, $(u, v) \in T$, then $u$ is a child of $v$, and $v$ is the parent of $u$. The leaves of $T$ are the nodes in $V \setminus \{r\}$ that have no descendants. We denote the leaf set of $T$ by $L(T)$, or simply by $L$, when the context is clear. The rooted subtree of $T$ induced by $v$ and its descendants is denoted by $T_v$ ($v$ is the root of $T_v$). A subtree $T'$ of $T$ is called a rooted subtree of $T$ if $T' = T_v$ for some $v \in V$.

To contract a rooted subtree $T'$ of $T$ is to combine all nodes in $T'$ into a single node $v$. The edges and links with both endpoints in $T'$ are deleted. The edges and links with one endpoint in $T'$ now have $v$ as their new endpoint. If we add a link $uv$ to a partial solution $I$, then the nodes along the path $P(uv)$ belong to the same 2-edge-connected component of the augmented graph $(V, E \cup I)$. Hence, we may contract some or all the edges of $P(uv)$. For a set of links $I \subseteq E$, let $T/I$ denote the tree obtained by contracting every 2-edge-connected component of $T \cup I$ into a single node. Since all contractions are induced by subsets of links, we refer to the contraction of every 2-edge-connected component of $T \cup I$ into a single node simply as the contraction of the links in $I$.

**Definition 2.1 (shadow)** A link $u'v'$ is a shadow of a link $uv$ if $P(u'v') \subseteq P(uv)$. A cover $F$ of $T$ is shadows minimal if for every link $uv \in F$ replacing $uv$ by a proper shadow of $uv$ results in a set of links that does not cover $T$.

Every TAP instance can be rendered closed under shadows by adding all shadows of existing links. We refer to the addition of all shadows as shadow completion. Shadow completion does not affect the optimal solution size, since every shadow can be replaced by some link covering all edges covered by the shadow. Thus we may assume the following.

**Assumption 2.1** The set of links $E$ is closed under shadows, that is, if $uv \in E$ and $P(u'v') \subseteq P(uv)$ then $u'v' \in E$.

The up-link $up(a)$ of a node $a$ is the link $au$ such that $u$ is as close as possible to the root; under Assumption 2.1, $u$ is an ancestor of $a$. For a rooted subtree $T'$ of $T$ and a node $a \in T'$, we say that
Figure 1: Illustration to Definitions 2.2 and to the proof of Claim 2.4. The twin-link and the locking link are shown by dotted lines, links of $F$ are shown by solid thin lines. Some edges may be paths, and the $uv$-path may have length zero.

$T'$ is $a$-closed if $up(a)$ belongs to $T'$, and $T'$ is $a$-open otherwise.

**Definition 2.2 (twin link and stem)** A link between leaves $a, b$ is a twin link if its contraction results in a new leaf; $a, b$ are called twins and their least common ancestor $s$ is called a stem.

**Definition 2.3 (locking link and locked leaf)** A leaf $a$ is locked by a link $bb'$, and $bb'$ is a locking link of $a$, if (see Figure 1(a)) there exists a rooted subtree $T_v$ of $T$ such that $v \neq r$, $L(T_v) = \{a, b, b'\}$, $ab$ is a twin link, and $T_v$ is $a$-closed.

For $X, Y \subseteq V$ and a link set $F$, let $F(X, Y) = \{xy \in F : x \in X, y \in Y\}$ denote the set of links in $F$ that have one endpoint in $X$ and the other in $Y$; for $x \in V$ let $d_F(x) = |F(x, V)|$ be the degree of $x$ w.r.t. $F$. For the rest of the paper we fix $F$ to be some optimal shadows minimal cover of $T$ with maximal number of twin links. We now give some properties of $F$ that we use later.

**Claim 2.2** $d_F(a) = 1$ for every leaf $a \in L$ of $T$.

**Proof:** If $ax, ay \in F$ for $x \neq y$, then replacing the link $ax$ by its shadow $a'x$, where $a'$ is the parent of $a$, results in a cover of $T$, contradicting the shadows minimality of $F$. □

**Claim 2.3** Let $a, b$ be twins with stem $s$, and let $ax, by \in F$. Then either $ab \in F$ and $d_F(s) = 1$, or $x, y \notin T_s$ and $d_F(s) = 0$. Furthermore, none of $P(sx), P(sy)$ contains the other.

**Proof:** If $ab \in F$ then by contracting $ab$ we get a new leaf $s'$, and $deg_F(s') = 1$, by Claim 2.2, which implies $deg_F(s) = 1$. Suppose that $ab \notin F$. One of $x, y$ is not in $T_s$, say $x \notin T_s$; otherwise $(F \setminus \{ax, by\}) \cup \{ab\}$ is a cover of $F$ of size smaller than $|F|$. We cannot have $y \in T_s$ since then $F \setminus \{ax, by\} \cup \{ab, sx\}$ is a shadow minimal cover of $T$ of size $|F|$ with more twin links than $F$, contradicting our choice of $F$. The second statement follows from the shadows-minimality of $F$. □

**Claim 2.4** Consider a locking leaf $a$ and a tree $T_v$ as in Definition 2.3, and suppose that $ax \in F$ for some $x \notin \{b, b'\}$. Then $bb' \in F$, $x$ is a proper ancestor of $u$, and there is a link $xz \in F$ such that $z \notin T_v$ and $z$ is not a locked leaf.

**Proof:** Let $by$ be the link in $F$ incident to $b$. Note that $x, y \notin P(b'u)$. Otherwise, if say $x \in P(b'u)$ (see Figure 1(b)), then optimality and shadows-minimality of $F$ implies that the unique link in $F$
incident to $b'$ must be $b'x$; but then $(F \setminus \{ax, b'x\}) \cup \{ab, b's\}$ is a cover of $T$ of size $|F|$ with more twin links than $F$. Since $x \notin P(b'u)$, Claim 2.3 implies that $x$ is a proper ancestor of $s$. It can’t be that we cannot have $y \notin T_v$ (see Figure 1(c)) as then $P(sy)$ contains $P(sx)$. In addition, $y$ cannot be an ancestor of $s$ (see Figure 1(d)). Thus $y = b'$ (see Figure 1(e)). Now consider the link $x'z \in F$ that covers the edge between $x$ and its parent, where $x' \in T_v$ and $z \notin T_v$. Note that $x' \notin \{a, b, b'\}$, by Claim 2.2. By shadow minimality of $F$, we must have $x' = x$. Any rooted subtree that contains $z$ is either $z$-open or contains $T_v$, hence $z$ cannot be a locked leaf. □

3 The lower bound and the algorithm

3.1 The Lower Bound

Let $S$ denote the set of stems of $T$ and let $X = V \setminus (L \cup S)$.

Lemma 3.1 Let $W$ be the set of twin and locking links, $M$ a maximum matching in $E(L,L) \setminus W$, and $U$ the set of leaves unmatched by $M$. Let $N = \{bb' \in M : \text{each of } b, b' \text{ is unmatched by } M\}$ where $M_F = F(L,L) \setminus W$. Let $J$ be the set of links in $F$ not incident to a locked leaf. Then:

$$\frac{3}{2} |F| \geq \frac{3}{2} |M| + \frac{1}{2} |U| + \frac{1}{2} \sum_{x \in X} \min\{2, d_J(x)\}.$$ (1)

Proof: Define a weight function $w$ on $E(L,V)$ by:

$$w(e) = \begin{cases} 
3/2 & \text{if } e \in E(L,L) \setminus W \\
2 & \text{if } e \in W \\
1 & \text{if } e \in E(L,V \setminus L)
\end{cases}$$

We prove the following two inequalities, that imply (1):

$$\frac{3}{2} |F| \geq w(F(L,V)) + \frac{1}{2} \sum_{x \in X} \min\{d_J(x), 2\}$$ (2)

$$w(F(L,V)) \geq \frac{3}{2} |M| + |U| + \frac{1}{2} |N|$$ (3)

We prove (2). Assign $3/2$ tokens to every $e \in F$. We will show that these tokens can be reassigned such that: every link in $F(L,L) \setminus W$ keeps its $3/2$ tokens, every link in $W \cap F$ gets 2 tokens, every link in $F(L,V \setminus L)$ keeps 1 token from its $3/2$ initial tokens, and every $x \in X$ gets 1/2 token for each link in $F \setminus J$ incident to $x$. Such an assignment is achieved as follows.

For every $e \in F$, move 1/2 token from the $3/2$ tokens of $e$ to each non-leaf endnode of $e$, if any. Let $e \in W \cap F$. Note that $e$ keeps its $3/2$ tokens, and we will assign to $e$ additional 1/2 token moved earlier to some non-leaf node by some other link, as follows:

- If $e$ is a twin link, then $e$ gets 1/2 token located at its stem $s$, that were moved to $s$ from the link in $F$ incident to $s$; such a link exists and is unique, by Claim 2.3.
If $e$ is a locking link, then $e$ gets $1/2$ token that were moved to $x$ from the link $ax \in F$ incident to the leaf $a$ locked by $e$; such a link $ax$ exists and is unique, and $x \in X$, by Claim 2.2.

We prove (3). By Claim 2.2 $F(L, L)$ is a matching, hence:

$$w(F(L, V)) = \frac{3}{2}|M_F| + 2|F(L, L) \cap W| + (|L| - 2|M_F| - 2|F(L, L) \cap W|) = |L| - \frac{1}{2}|M_F|$$

It is not hard to see that $|M| - |M_F| \geq |N|$, and clearly $\frac{3}{2}|M| + |U| = L - \frac{1}{2}|M|$. Thus we have

$$w(F(L, V)) - \left(\frac{3}{2}|M| + |U|\right) = \frac{1}{2}(|M| - |M_F|) \geq \frac{1}{2}|N|,$$ as claimed in (3). \qed

We prove the following statement that implies Theorem 1.1.

**Theorem 3.2** There exists a polynomial time algorithm that given an instance of TAP computes a solution $I$ of size at most the right-hand size of (1). Thus $|I| \leq 1.5 \cdot |F|$.

### 3.2 The credit scheme and the invariants of the algorithm

**The Credit Scheme.** Initially, the algorithm assigns coupons to unmatched leaves and to links in $M$, such that the total number of coupons assigned equals the sum of the first two terms in the right hand side of (1). For technical reasons, we also assign 1 coupon to the root $r$.

**Initial assignment of coupons:**

- Every link in $M$ gets $3/2$ coupons.
- Every unmatched leaf gets 1 coupon, and $r$ gets 1 coupon.

Our algorithm iteratively contracts certain subtrees of $T/I$. We refer to the nodes created by contractions as compound nodes. Each compound node always owns 1 coupon. Non-compound nodes are referred to as original nodes (of $T$). Each time a contraction takes place, the new compound node gets 1 coupon, which together with the links added is paid by the total credit of the contracted subtree. For technical reasons, $r$ is also considered as a compound node. Our algorithm is designed to maintain the following invariant.

**Invariant 3.3 (Coupons Invariant)** Every unmatched leaf and every (unmatched) compound node of $T/I$ (including $r$) owns 1 coupon, and every link in $M$ owns $3/2$ coupons.

To charge the term $\frac{1}{2}|N| + \frac{1}{2} \sum_{x \in X} \min\{2, d_J(x)\}$. in (1), we introduce tickets. A ticket worth 1/2 coupon. Suppose that we want to contract a subtree $T'$ of $T/I$, and to claim a ticket (1/2 coupon) in $T'$. Note that we do not need to specify the node or the link on which the ticket is claimed. All we care about is that $T'$ contains a ticket, for any possible choice of $F$. However, some care is needed to claim tickets properly. We will discuss this issue later.

**Greedy Contractions.** The following simple operation maintains the Coupons Invariant.
Definition 3.1 Let $uv \in E$ such that $\text{coupons}(p(uv)) \geq 2$ and either both $u, v$ are unmatched by $M$, or $uv \in M$. Then adding $uv$ to the partial solution $I$ and assigning 1 coupon to the obtained compound node is called a greedy contraction.

One of the steps of the algorithm is to apply greedy contractions exhaustively; clearly, this can be done in polynomial time. The order of greedy contractions can be arbitrary, with one exception.

Rule 3.4 (Contracting all links incident to a locked leaf) Consider a leaf $a$ locked by a link $bb'$ as in Definition 2.3, and suppose that $a, b, b'$ are all unmatched by $M$. Then in greedy contractions, $bb'$ is contracted first and $up(a)$ second. In the case that $b$ is also a locked leaf, we perform the same operation, assuming that $up(b)$ is a shadow of $up(a)$. Note that then, all links incident to $a$ (and to $b$, if $up(b)$ is a shadow of $up(a)$) are contracted into the new compound leaf.

We have the following property of the matching $M$.

Lemma 3.5 If all greedy contractions are exhausted then no contraction of a link in $M$ creates a new leaf.

Proof: Let $bb' \in M$. Note that in the original tree $T$ the contraction of $bb'$ does not create a leaf, since $M$ contains no twin link. Thus if the contraction of $bb'$ creates a leaf in $T/I$, then $P(bb')$ has a compound node in $T/I$. This holds as some structure that hanged out of a node of $P(bb')$ and contained a leaf is now gone (this structure was avoiding the contraction of $bb'$ in $T$ from creating a new leaf). But then $bb'$ gives a greedy contraction, contradicting the assumption. □

Matching Invariants. We summarize the essential properties of the matching $M$ as follows.

Invariant 3.6 (Matching Invariant) $M$ is a matching on leaves of $T/I$ such that:

(M1) No contraction of a link in $M$ creates a new leaf (in particular, $M$ has no twin link).

(M2) Every stem has at least one child that is matched by $M$.

(M3) Every node $b$ matched by $M$ is an original leaf of $T$, thus $d_F(b) = 1$ (by Claim 2.2).

In the Weak Matching Invariant property (M3) is replaced by the weaker property

(M3') Every node $b$ matched by $M$ has $d_F(b) = 1$.

Properties (M1) and (M2) hold after the greedy contractions are exhausted. (M1) holds by Lemma 3.5. (M2) holds since if two children of a stem are unmatched by $M$, then the link between them gives a greedy contraction. (M3) follows from the definition of $M$, and it is not affected by greedy contractions. Note that (M3) implies (M3'). While property (M3) is essential, in the analysis, we need to consider a modification of $M$, when a node $b$ matched by $M$ is a compound node obtained by contracting a twin link. This modification will satisfy (M3') instead of (M3).

Claiming tickets properly. To claim a ticket in a rooted subtree $T'$ of $T/I$, we prove that for any choice of $F$, one of the following must hold:
\* \textbf{N-ticket:} There is a link $bb' \in M$ with $b, b' \in T'$ such that each of $b, b'$ is an endnode of a twin link in $F$; note that indeed $bb' \in N$, by Claim 2.2.

\* \textbf{X-ticket:} There is a link $e = a'x \in F$ with $x \in X \cap T'$, such that either $a'$ is an unmatched leaf of $T'$ and $T'$ is $a'$-closed, or $a' \notin T'$.

In the case of an X-ticket, we need to verify that the original endnode of $e$ contained in $a'$ is not a locked leaf. Note that Claim 2.4 implies that every $x \in X$ with $d_F(x) \geq 1$ has at least one ticket. This suffices if we need to show existence of only one ticket in $T'$. In the case when $T'$ has no matched pair and no non-leaf compound nodes, we will need to show existence of two tickets. However in this case Rule 3.4 and our choice of $T'$ will guarantee that if a compound leaf of $T'$ contains a locked leaf, then it contains all the links incident to this locked leaf. Summarizing, we use the following rule to claim tickets properly.

\textbf{Rule 3.7 (Claiming tickets properly)}

\begin{itemize}
\item \text{(i) No ticket is claimed twice: Immediately after claiming tickets in a rooted subtree $T'$ of $T/I$, $T'$ is contracted into a new compound node, while tickets are never claimed on compound nodes.}
\item \text{(ii) No X-ticket is claimed for a link incident to a locked leaf: $T'$ has no non-leaf compound nodes. If $T'$ has no matched pair, then any leaf of $T'$ that contains a locked leaf, contains all the links incident to this locked leaf. Otherwise, at most one ticket is claimed in $T'$.}
\end{itemize}

\section{The algorithm}

We use the following notation for the credit distributed in a rooted subtree $T'$ of $T/I$. Let $\text{coupons}(T')$ denote the total number of coupons owned by $T'$; this includes the coupons owned by nodes of $T'$ (every unmatched leaf and every compound node of $T'$ owns 1 coupon), and $3/2$ coupons for every link in $M$ with both endnodes in $T'$. Let $\text{tickets}(T') = |N(T')| + \sum_{x \in X \cap T'} \deg_F(x)$ denote the number of tickets in $T'$. Let $\text{credit}(T') = \text{coupons}(T') + \frac{1}{2} \text{tickets}(T')$.

\textbf{Definition 3.2} A rooted subtree $T'$ of $T/I$ is $M$-compatible \textit{if for any $bb' \in M$ either both $b, b'$ belong to $T'$, or none of $b, b'$ belongs to $T'$. A set of links $B'$ is an exact cover of $T'$ if the set of edges of $T/I$ that is covered by $B'$ equals the set of edges of $T'$.}

The high level idea of our algorithm is as follows. We repeatedly find an $M$-compatible rooted subtree $T'$ of $T/I$ and an exact cover $B'$ of $T'$, add $B'$ to $I$, contract $T'$, and to assign a coupon on the resulting compound node (in order to satisfy the Coupons Invariant 3.3). Note that since $T'$ is $M$-compatible and $B'$ is an exact cover of $T'$, property (M3) continues to hold during the algorithm. If we require not to over spend the credit provided by the right hand side of (1), we need $\text{credit}(T') \geq |B'| + 1$ to hold. If we iteratively cover trees in this way until $T$ is completely covered, then the number of links we use is at most the right hand side of Inequality (1), as claimed in Theorem 3.2. In the next section we will prove the following key statement.

\textbf{Lemma 3.8} Suppose that the Coupons Invariant and the Matching Invariant hold for $T, M, I$ and
that $M$ has no locking links. Then there exists a polynomial time algorithm that finds an $M$-
compatible subtree $T'$ of $T/I$ and an exact cover $B' \subseteq E$ of $T'$ such that $\text{credit}(T') \geq |B'| + 1$.

Algorithm Tree-Cover (Algorithm 1) initiates $I \leftarrow \emptyset$ as a partial cover. It computes a
maximum matching $M$ in $E(L, L) \setminus W$ and initiates the credit scheme as described in Section 3.2.
In the main loop, the algorithm iteratively exhausts greedy contractions, then computes $T', B'$ as
in Lemma 3.8, adds $B'$ to $I$, contracts $T'$, and leaves a coupon on the created compound leaf. The
stopping condition is when $I$ covers $T$, namely, when $T/I$ is a single node.

It is easy to see that all the steps in the algorithm can be implemented in polynomial time, and
that each time the greedy contractions are exhausted, the Coupons Invariant and the Matching
Invariant hold for $T, M, I$. The credit scheme used implies that the algorithm computes a solution
$I$ of size at most $1.5$ times the right-hand size of (1). Hence it only remains to prove Lemma 3.8.

Algorithm 1: Tree-Cover($T = (V, E), E$) (A 1.5-approximation algorithm)

1. initialize: $I \leftarrow \emptyset$
2. $M \leftarrow$ maximum matching in $E(L, L) \setminus W$.
3. Give 1 coupon to every unmatched leaf and to $r$, and give $3/2$ coupons to every link in $M$.
4. while $T/I$ has more than one node do
   5. Exhaust greedy contractions and update $I$ and $M$ accordingly.
   6. Find a subtree $T'$ of $T/I$ and a cover $B'$ of $T'$ as in Lemma 3.8.
   7. Contract $T'$, assign 1 coupon to the new compound leaf, and set $I \leftarrow I \cup B'$.
8. return $I$

4 Proof of Lemma 3.8

4.1 Semi-closed trees and their properties

For a node set $U \subseteq V$, we let $\text{up}(U) = \{\text{up}(u) : u \in U\}$.

Definition 4.1 ([14]) Let $U$ be a subset of nodes of $T$. A rooted subtree $T'$ of $T$ is $U$-closed if
there is no link in $E$ from $U \cap T'$ to $T \setminus T'$. $T'$ is leaf-closed if it is $L(T)$-closed. A leaf-closed $T'$
is minimally leaf-closed if any proper rooted subtree of $T'$ is not leaf-closed.

Proposition 4.1 ([14]) If $T'$ is minimally leaf-closed, then $\text{up}(L(T'))$ is an exact cover of $T'$.

A naive approach to find $T', B'$ as in Lemma 3.8 is by taking $T'$ to be minimally leaf-closed and
$B' = \text{up}(L(T'))$. However, we do not have enough credit for that, since $|L(T')|$ can be much larger
than $\text{coupons}(T')$, not to mention the one extra unit of credit. This is since every link $bb' \in M$
owns $3/2$ coupons, while taking the up-links of $b, b'$ requires $2$ coupons. Hence we define $T'$ and
its cover $B'$ so that $\text{coupons}(T') \geq |B'|$ holds, and some extra credit is provided with the help of
tickets. Our main new idea is to use tickets, and minimally semi-closed trees, defined below, which
Definition 4.2 (Semi-closed tree) A rooted subtree $T'$ of $T/I$ is semi-closed (w.r.t. a matching $M$ on the leaves of $T/I$) if it is $M$-compatible and closed w.r.t. its unmatched leaves. $T'$ is minimally semi-closed if $T'$ is semi-closed but any proper subtree of $T'$ is not semi-closed.

Note that a semi-closed $T'$ may not leaf-closed (this why we called it “semi-closed”); $T'$ is closed with respect to every unmatched leaf, but matched leaves may have links to nodes outside $T'$. In what follows, let us use the following notation:

- $L' = L(T')$ is the set of leaves of $T'$.
- $M' = M(T')$ is the set of links in $M$ with both endnodes in $T'$.
- $U' = U(T')$ is the set of unmatched leaves of $T'$.
- $S'$ is the set of stems of $T'$.
- $S'_1 = \{ s \in S' : \text{the twin-link between the children of s is in } F \}$.
- $C'$ is the set of non-leaf compound nodes of $T'$ (recall that this includes $r$, if $r \in T'$).
- $B(T') = M' \cup up(U')$ (hence $|B(T')| = |M'| + |U'| = |L'| - |M'|$).

The following statement explains how we intend to cover minimally semi-closed trees.

Lemma 4.2 If $T'$ is minimally semi-closed then $B(T')$ is an exact cover of $T'$.

Proof: Let $T''$ be obtained from $T'$ by contracting $M'$. Note that $L(T'') = U'$. Otherwise, if $T''$ has a leaf $a$ that is not a leaf of $T'$, then the subtree of $T'$ that was contracted into $a$ is a semi-closed tree (with no unmatched leaves), contradicting the minimality of $T'$. Note also that $T''$ is minimally leaf-closed, since $T'$ is minimally semi-closed. Thus $up(U')$ is an exact cover of $T''$ (if $T'$ has no unmatched leaves then $T''$ is a single node), by Proposition 4.1. As a link in $M'$ can cover only edges in $T'$, the statement follows. \qed

Definition 4.3 (Deficient tree) A semi-closed tree $T'$ is deficient if $\text{credit}(T') < |B(T')| + 1$.

In the next section we characterize deficient trees in terms of how our optimal solution $F$ covers them. As this characterization depends on $F$, it does not provide a polynomial time algorithm for checking whether a given semi-closed tree is deficient. We thus consider the following family of trees, that includes the deficient trees, and can be recognized in polynomial time.

Definition 4.4 (Dangerous tree) A semi-closed tree $T'$ is dangerous if $|C'| = 0$, $|M'| = 1$, and one of the following holds:

(i) $|L'| = 3$, $|S'| \leq 1$, and $T'$ is as in Figure 2(a) with all the links depicted present in $E$. Namely, there exists an ordering $a,b,b'$ of $L'$ such that $bb' \in M$, $ab' \in E$, the contraction of $ab'$ does not create a new leaf, and there exists a link $bz \in E$ with $z \notin T'$ (namely, $T'$ is $b$-open).

(ii) $|L'| = 4$, $|S'| \in \{1,2\}$, and $T'$ is as in Figure 2(b,c) with the links depicted present in $E$. Namely, $T'$ is obtained from a 3-leaf dangerous tree by the following operation: choose one of $b,b'$, say $b$, add a stem $s$ as a parent of $b$ (namely, subdivide the parent edge of $b$ by a stem $s$)
Figure 2: Dangerous trees. The dashed arc shows the matched pair \( bb' \) and solid thin lines show links in \( E \) that must exist. In (b,c) \( s \) is a stem, and in (a,c) \( w \) may be a stem. Some of the edges of \( T' \) can be paths, and \( u = v \) or/and \( u = w \) may hold. Note that any 4-leaf dangerous tree has a stem \( s \) such that contracting the twin link between its children results in a 3-leaf dangerous tree.

Figure 3: Bad trees. The solid thin lines show the links in \( F \), and \( s, s' \) are stems.

*with additional child \( a' \), and replace the endnode \( b \) by \( s \) for the link that is incident to \( b \).*

Note that the property of a tree being dangerous depends only on the structure of the tree and existence/absence of certain links in \( E \) and \( M \), and thus can be tested in polynomial time. In the next section we will prove the following.

**Lemma 4.3** Suppose that the Coupons Invariant and the Matching Invariant hold for \( T, M.I \) and that \( M \) has no locking links. Then any deficient tree is dangerous.

Before proving Lemma 4.3, we explain the reason for the absence of locking links in \( M \) and the existence of \( N \)-tickets. These are needed to avoid two specific “bad trees” depicted in Figure 3 of being deficient, where the thin lines now show the links in \( F \). If \( M \) could include locking links, a deficient tree \( T' \) could be as in Figure 3(a); this tree may not be dangerous – e.g., if \( ab' \notin E \). However, the existence of the link \( ab' \) is essential for our algorithm. Note that if property (M3) holds, then \( b' \) is an original leaf, and thus the link \( bb' \) locks \( a \). Thus this case cannot occur if (M3) holds; consequently, \( T' \) is not deficient. Similarly, without the \( N \)-tickets, a deficient tree \( T' \) may be as in Figure 3(b), and such \( T' \) may not be dangerous – e.g., if \( as, a's' \notin E \); however, the existence of one of these links is essential for our algorithm. Note that if our solution \( F \) indeed covers the tree \( T' \) in Figure 3(b), then \( T' \) has an \( N \)-ticket. Thus \( \text{credit}(T') = 3.5 + 0.5 = 4 \), while \( |B(T')| = 3 \); consequently, \( T' \) is not deficient. Note that the contraction of the twin link \( a'b' \) in (b) gives the tree in (a); then (M3) does not hold as \( b' \) is not an original leaf, but (M3') holds as \( d_F(b') = 1 \).
The reader may note that in the tree in Figure 2(a) \( w \) cannot be a stem (otherwise, \( bb' \) is a locking link in \( M \)). This is true if (M3) holds, but if only (M3') holds, then \( b' \) may be a compound node, and then \( bb' \) is not a locking link. Specifically, this is so if the tree in Figure 2(a) is obtained from the tree in Figure 2(c) (with \( w \) being a stem) by contracting the link \( a'b' \). For technical reasons, we need the property that any 4-leaf dangerous tree (in particular, if it has 2 stems) can be reduced to a 3-leaf dangerous tree by contraction of a twin link. Hence our family of dangerous trees will include the tree in Figure 2(a) with \( w \) being a stem, even if such a tree cannot be deficient.

4.2 Proof of Lemma 4.3

In this section, let \( T' \) be a deficient tree with root \( v \). We will show that then \( T' \) is as in Lemma 4.3. For that, we start by deriving some properties of \( T' \), and in particular show that \(|L'| \in \{3, 4\} \), see Claims 4.4 – 4.7. Then in Claim 4.8 we will show that if \( T' \) has 3 leaves, then it is either dangerous, or it is a "bad tree" and \( F \) covers it as depicted in Figure 3(a). All this is done relying on the Weak Matching Invariant only. By applying condition (M3) and using the fact that \( M \) has no locking link, we exclude the case that \( T' \) is a bad tree as in Figure 3(a), proving Lemma 4.3 for 3-leaf trees. Then, relying on our proof for 3-leaf trees, we will prove Lemma 4.3 for 4-leaf trees.

Note that \( |B(T')| = |U'| + |M'| \) and that \( coupons(T') = |U'| + |C'| + \frac{3}{2}|M'| \). Hence

\[
credit(T') - |B(T')| - \frac{1}{2}tickets(T') = coupons(T') - |B(T')| = |C'| + \frac{1}{2}|M'|.
\]

From this we immediately get the following.

**Claim 4.4** \(|C'| = 0 \) (so \( r \notin T' \)) and \(|M'| + tickets(T') \leq 1\).

**Claim 4.5** If \(|M'| = 0 \) then \( tickets(T') \geq |L'| + 1 \geq 2 \). Thus \(|M'| = 1 \), \(|L'| \geq 3 \), and \(|S'| \leq 2\).

**Proof:** By (M2), \(|S'| \leq 2|M'| \), hence \(|M'| = 0 \) implies \(|S'| = 0\). By Rule 3.4, \(|M'| = 0 \) implies that no link incident to a leaf of \( T' \) was originally incident to a locked leaf. Let \( F' \) be obtained by picking for every \( a \in U' \) one link \( ax \in F \) that covers \( a \), and let \( yz \in F \) be a link that covers the edge between \( v \) and its parent (note that \( C' = \emptyset \) implies \( v \neq r \)), where \( z \notin T' \). For every \( ax \in F' \), \( x \in T' \setminus U' \) (as \( T' \) is \( L' \)-closed and no greedy contraction applies), hence \( ax \) contributes a ticket at \( x \) and \( |F'| = |L'| \). Note that \( y \in T' \setminus U' \) (since \( T' \) is \( L' \)-closed), hence the link \( yz \) contributes a ticket at \( y \). Consequently, the links in \( F' \cup \{yz\} \) contribute \( |F'| + 1 = |L'| + 1 \) tickets in \( T' \). The second statement follows from the observation that if \(|L'| \leq 2 \) then \(|M'| = 0 \) (by (M1)) and that \(|S'| \leq 2|M'| \) (by (M2)). \( \square \)

Let \( e = bb' \) be the unique link in \( M' \). Note that so far we used only properties (M1) and (M2), but now we will assume that (M3') holds as well, and thus \( \deg_F(b) = \deg_F(b') = 1 \). Also note that Claims 4.4 and 4.5 imply \( tickets(T') = 0 \).

**Claim 4.6** \(|L'| \leq |S'| + 3\).
Figure 4: Illustration to the proof of Claim 4.8.

**Proof:** Let $F'$ and $yz$ be as in the proof of Claim 4.5. Note that $T'$ has a ticket, unless every link in $F' \cup \{yz\}$ has an endnode in $V' = (L' \setminus U') \cup S'_1$. Since $\deg_F(b) = 1$ for every $b \in V'$ and since $T'$ has no ticket, we must have $|F' \cup \{yz\}| \leq |V'|$. Now note that $|M'| = 1$ implies $|L'| = |U'| + 2$, and thus $|F' \cup \{yz\}| = |L'| - 1$ and $|V'| = |S'_1| + 2$. Thus $|L'| - 1 \leq |S'_1| + 2$, as claimed. □

**Claim 4.7** $|S'_1| \leq 1$.

**Proof:** By Claim 4.5, $|S'_1| \leq |S'| \leq 2$. We cannot have $|S'_1| = 2$, since then $T'$ gets an $N$-ticket from $e = bb'$ (each one of $b, b'$ is an endnode of a twin link in $F$). □

**Claim 4.8** Suppose that the Coupons Invariant and the Weak Matching Invariant hold. If $|L'| = 3$ then either $T'$ is a bad tree depicted in Figure 3(a), or $T'$ is dangerous (see Figure 3(b,c)).

**Proof:** Consider a 3-leaf tree as in Figure 4. If $\deg_F(w) \geq 1$, then $w$ must be a stem (otherwise $w$ has a ticket); thus by Claim 2.3, $ab \in F$ and $\deg_F(w) = 1$. If $wb' \in F$, then any link that covers the edge between $v$ and its parent gives a ticket. Otherwise, to avoid a ticket, $T'$ must be as in Figure 4(a), which is the bad tree from Figure 3(a).

Suppose that $\deg_F(w) = 0$. Let $ax \in F$ be the link that covers $a$, and let $yz \in F$ be a link that covers the edge between $v$ and its parent, where $y \in T'$. Since $T'$ is $a$-closed, $y \neq a$. Thus $x, y \in \{b, b'\}$, as otherwise $T'$ has a ticket at $x$ or at $y$. Also, $x \neq y$, by (M3’). Consequently, if $F'$ is the set links in $F$ with at least one endnode in $\{a, b, b'\}$, then either $F' = \{ab, b'z\}$, or $F' = \{ab', bz\}$. If $u \neq w$, then the former case $F' = \{ab, b'z\}$ is not possible; this is since there must be a link in $F$ covering the edge between $w$ and its parent, but this link cannot be incident to one of $a, b$, and thus gives a ticket. Thus $F' = \{ab', bz\}$, and we arrive at the case in Figure 4(b), which is the 3-leaf dangerous tree in Figure 2(a). If $u = w$ (see Figure 4(c)), then there is no difference in the roles of $b, b'$, and we can have either $ab, b'z \in F$ or $ab', bz \in F$, obtaining in both cases a 3-leaf dangerous tree. □

Now we apply (M3) and the fact that $M$ has no locking link, to conclude that the case in Figure 4(a) cannot occur – $w$ cannot be a stem, as otherwise the leaf $a$ is locked by the link $bb' \in M$. Thus Lemma 4.3 for 3-leaf trees follows from Claim 4.8.

**Claim 4.9** If $|L'| = 4$ then $T'$ is dangerous (see Definition 4.4(ii)).
Proof: From Claims 4.6 and 4.7 we get that $|S'_1| = 1$; namely, $T'$ has a unique stem, say $s$, such that the twin link between its children, say $f$, is in $F$. One of the children of $s$ must be matched by $M$, by (M2). Let $I' = I \cup \{f\}$ and consider the tree $T/I'$ and its subtree $T''$ obtained from $T/I$ and $T'$, respectively, by contracting $f$. The contraction of $f$ creates a new leaf $b''$ that is now matched by $M$. Note that this contraction can be paid by the coupon of the unmatched child of $s$, and that $b$ does not need a coupon since it is matched by $M$. Consequently, the Coupons Invariant holds, without overspending the credit. By Claims 2.2 and 2.3, $\text{deg}_F(b) = 1$; other parts of the Weak Matching Invariant just continue to hold after the contraction of $f$. Note that $T''$ has 3 leaves, hence it must be as in Claim 4.8 as if $T''$ has a ticket, then so is $T'$. Now note that $T''$ cannot be a bad tree as in Figure 4(a), since then we will have $|S'_1| = 2$, a case refuted in Claim 4.7 by existence of an $N$-ticket. Using for $T''$ our usual notation of Definition 4.4(i), where the matched pair of a deficient tree is denoted by $bb'$, we get that either $b'' = b$ or $b'' = b'$. Consequently, it is not hard to verify that $T'$ must be dangerous as in Definition 4.4(ii). □

4.3 Finding a good tree when all minimally semi-closed trees are dangerous

We remind the reader that the property of a tree being dangerous depends only on the structure of the tree and existence/absence of certain links in $E$ and $M$, and thus can be tested in polynomial time. Note that we can compute all semi-closed and dangerous subtrees of $T/I$ in polynomial time, as every semi-closed or dangerous tree is a rooted subtree of $T/I$, and since we can check in polynomial time whether a given subtree is semi-closed or dangerous. If $T/I$ has a minimally semi-closed subtree $T'$ that is not dangerous, then $T'$ is not deficient, so $T'$ and $B(T')$ satisfy the requirement of Lemma 3.8. Thus in this section we will assume that all minimally semi-closed subtrees of $T/I$ are dangerous. In this case, we will show that Algorithm 2 below finds an $M$-compatible tree $T'$ and its cover $B'$ such that $\text{credit}(T') \geq |B'| + 1$.
Algorithm 2: Find-Tree\(T = (V, E), E, M\) (Finds a non-deficient tree \(T'\) and its cover \(B'\) such that \(|B'| = |B(T')|\), when all minimally semi-closed trees are dangerous.)

1 Let \(\bar{I}\) be the link set obtained by picking for every 4-leaf dangerous tree a twin-link (see Figure 2(c,d)) whose contraction results in a 3-leaf dangerous tree.

2 \(\bar{T} \leftarrow T/(I \cup \bar{I}) = (T/I)/\bar{I}\) (namely, \(\bar{T}\) is obtained from \(T/I\) by contracting the links in \(\bar{I}\)).
   \(\triangleright\) Comment: Every 3-leaf dangerous tree of \(T/I\) remains dangerous in \(\bar{T}\), while every 4-leaf dangerous tree of \(T/I\) is transformed into a 3-leaf dangerous tree of \(\bar{T}\).

3 Let \(\hat{M}\) be a matching on the leaves of \(\hat{T}\) obtained from \(M\) by replacing the link \(e = bb'\) by the link \(\hat{e} = ab'\) in every 3-leaf dangerous tree \(T_0\) of \(\hat{T}\) (see Figure 4(c,d)).

4 Let \(\hat{T}'\) be a minimally semi-closed tree in \(\hat{T}\) w.r.t. the matching \(\hat{M}\).

5 Let \(\hat{B}' = M(\hat{T}') \cup \text{up}(\hat{U}')\), where \(\hat{U}'\) is the set of unmatched leaves of \(\hat{T}'\) w.r.t. \(\hat{M}\).

6 Let \(T'\) be the subtree of \(T/I\) and \(\hat{I}' \subseteq \bar{I}\) such that \(T'/\hat{I}' = \hat{T}'\).
   \(\triangleright\) Comment: \(\hat{I}'\) is the set of twin links in \(\bar{I}\) contained in the leaves of \(\hat{T}'\), and \(T'\) is obtained from \(\hat{T}'\) by “uncontracting” the links in \(\hat{I}'\).

7 \textbf{return} \(T'\) and \(B' = \hat{B}' \cup \hat{I}'\)

We now consider the tree \(\hat{T}'\) and its cover \(\hat{B}'\) computed at lines 4 and 5 of the algorithm.

Claim 4.10 For any dangerous tree \(T_0\) in \(\hat{T}\), either \(\hat{T}'\) properly contains \(T_0\), or \(\hat{T}', T_0\) are node disjoint. Consequently, \(\hat{T}'\) is not dangerous.

Proof: Note that \(\hat{T}\) has no 4-leaf dangerous trees, hence \(T_0\) has 3 leaves. As \(\hat{T}', T_0\) are rooted subtree of \(\hat{T}\), they are either node disjoint or one of them contains the other. If \(\hat{T}', T_0\) are not node disjoint, then they share a leaf. Suppose that \(T_0\) is as in Figure 4(b,c). Note that we must have \(b \in \hat{T}'\), since \(a \in \hat{T}'\) or \(b' \in \hat{T}'\) implies \(ab' \in \hat{T}'\) (since \(ab' \in \hat{M}\) and since \(\hat{T}'\) is semi-closed w.r.t. \(\hat{M}\)), hence the least common ancestor \(u\) of \(a, b'\), and all its descendants also belong to \(\hat{T}'\). Now let \(\text{up}(b) = bu\). Since \(\hat{T}'\) is \(b\)-closed, the root of \(\hat{T}'\) is an ancestor of \(u\). Since \(T_0\) is \(b\)-open, \(u\) is a proper ancestor of the root of \(T_0\). Hence \(T_0\) is a proper subtree of \(\hat{T}'\).

Let \(f: M \rightarrow \hat{M}\) be defined by \(f(e) = \hat{e}\) if \(e \in M \setminus \hat{M}\) and \(f(e) = e\) otherwise, where \(\hat{e}\) is as in Line 3 in Algorithm 2. Since the dangerous subtrees of \(\hat{T}\) are node disjoint, \(f\) is a bijection. Now we prove the following.

Claim 4.11 For any \(e \in M\), either each of \(e, f(e)\) has both endnodes in \(\hat{T}'\), or none of \(e, f(e)\) has an endnode in \(\hat{T}'\).

Proof: Let \(e \in M\). If \(f(e) = e\) then the statement holds, so assume that \(f(e) = \hat{e} = ab'\) is as in Line 4 in Algorithm 2 (see Figure 4(b,c)), and \(T_0\) is the dangerous tree with leaves \(a, b, b'\). By Claim 4.10, either \(\hat{T}'\) properly contains \(T_0\), or \(\hat{T}', T_0\) are node disjoint. In the former case, each of \(e, f(e)\) has both endnodes in \(\hat{T}'\), while in the later case none of \(e, f(e)\) has an endnode in \(\hat{T}'\).

Claim 4.12 \(\hat{T}'\) is semi-closed (w.r.t. \(M\)), \(\hat{B}'\) is an exact cover of \(\hat{T}'\), and \(|\hat{B}'| = |B(\hat{T}')|\).
Proof: By Claim 4.11, $\tilde{T}′$ is $M$-compatible. Hence to show that $\tilde{T}′$ is semi-closed we need to show that $\tilde{T}′$ is $a$-closed for any its leaf $a$ unmatched by $M$. If $a$ is unmatched by $\tilde{M}$, then this is so since $\tilde{T}′$ is semi-closed w.r.t. $\tilde{M}$. Otherwise, $a$ is a leaf in a dangerous tree $T_0$ as in Figure 4(b,c), and $\tilde{T}′$ contains $T_0$, by Claim 4.10. As $T_0$ is $a$-closed, so is $\tilde{T}′$.

By the definition, $\tilde{T}′$ is a minimally semi-closed tree w.r.t. $\tilde{M}$, and $\tilde{B}′ = \tilde{M}(\tilde{T}′) \cup \up(\tilde{U}′)$, where $\tilde{U}′$ is the set of unmatched leaves of $\tilde{T}′$ w.r.t. $\tilde{M}$. It is easy to see that condition (M1) holds $\tilde{M}$, since it holds for $M$. Applying Lemma 4.2 on $\tilde{T}′$ and $\tilde{M}$, we get that $\tilde{B}′$ is an exact cover of $\tilde{T}′$.

To see that $|\tilde{B}′| = |B(\tilde{T}′)|$, note that by Claim 4.11, $|M(\tilde{T}′)| = |M(\tilde{T}′)|$; namely, the number links in $\tilde{M}$ and in $M$ with both endnodes in $\tilde{T}′$ is the same. As $|\tilde{B}′| = |L(\tilde{T}′)| - |\tilde{M}(\tilde{T}′)|$ and $|B(\tilde{T}′)| = |L(\tilde{T}′)| - |M(\tilde{T}′)|$, the last statement follows. □

Corollary 4.13 Algorithm 2 returns a pair $T′, B′$ such that the following holds: $T′$ is $M$-compatible, $B′$ is an exact cover of $T′$, $|B′| = |B(T′)|$, and $\text{credit}(T′) ≥ |B(T′)| + 1 = |B′| + 1$.

Proof: Recall that $\tilde{T}′$ is obtained from $T′$ by contracting the links in $\tilde{I}$ (see in Figure 2(b,c) the links $a′b$ in (b) and $a′b′$ in (c)). By Claim 4.12, $\tilde{T}′$ is semi-closed, and thus is $M$-compatible. From this it is easy to see that $T′$ is also $M$-compatible (but $T′$ as in Figure 2(b,c) may be $a′$-open and not semi-closed).

By Lemma 4.2, $\tilde{B}′$ is an exact cover of $\tilde{T}′$, and $\tilde{T}′$ is obtained from $T′$ by contracting $\tilde{I}′$. Thus $B′ = \tilde{B}′ \cup \tilde{I}′$ is an exact cover of $T′$.

To see that $|B′| = |B(T′)|$, note that $|\tilde{B}′| = |B(\tilde{T}′)|$ (by Claim 4.12) and that every link in $\tilde{I}′$ can be paid by the coupon of the corresponding unmatched leaf $a′$ of $T′$.

To prove that $\text{credit}(T′) ≥ |B(T′)| + 1$ it is sufficient to show that $T′$ is not dangerous (and thus is not deficient). Suppose to the contrary that $T′$ is dangerous. Note that every 3-leaf dangerous tree of $T′/I$ remains dangerous in $\tilde{T}$, while every 4-leaf dangerous tree of $T′/I$ is transformed into a 3-leaf dangerous tree of $\tilde{T}$. Thus $\tilde{T}′$ is a dangerous tree (with 3 leaves). But $\tilde{T}′$ is a semi-closed tree w.r.t. $\tilde{M}$, which contradicts Claim 4.10. □

Acknowledgment: We thank André Linhares, Joseph Cheriyan, Zhihan Gao, Chaitanya Swamy, and anonymous referees for useful comments, and Guy Even, Jon Feldman, and Samir Khuller for useful discussions.

References


