An LP 7/4-approximation for the Tree Augmentation Problem

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

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

Abstract

In the Tree Augmentation Problem (TAP) the goal is to augment a tree T by a minimum size edge set F from a given edge set E such that $T \cup F$ is 2-edge-connected. The best known approximation ratio for the problem is 1.5 [5, 12]. Several paper analysed integrality gaps of LP and SDP relaxations for the problem. In [13] is given a simple LP with gap 1.5 for the case when every edge in E connects two leaves of T. Recently, Cheriyan and Gao [1] showed that a certain SDP relaxation obtained by the Lasserre lift-and-project method has integrality gap $7/4 + \epsilon$. Here we show, by a simple and short proof, that a modification of the LP used in [13] has integrality gap 7/4, which is achieved by the combinatorial algorithm of [12].

1 Introduction

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 $G = (V, \mathcal{E})$ and a set of additional edges (called "links") E on V disjoint to $\mathcal{E}$, find a minimum size edge set $F \subseteq E$ such that $G + F = (V, \mathcal{E} \cup F)$ 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:

Tree Augmentation Problem (TAP)
Instance: A tree $T = (V, \mathcal{E})$ and a set of edges (called *links*) E on V disjoint to $\mathcal{E}$.
Objective: Find a minimum size subset $F \subseteq E$ of links such that $T \cup F$ is 2-edge-connected.

TAP can be formulated as the problem of covering a laminar family as follows. Root T at some node r. Every edge of T partitions T into two parts T' and $T \setminus T'$, where $r \notin T'$; let $\hat{\mathcal{E}}$ denote the set family obtained by picking for each edge the part T' that does not contain r. Then $\hat{\mathcal{E}}$ is laminar, and $F \subseteq E$ is a feasible solution for TAP if and only if F covers $\hat{\mathcal{E}}$, namely, for every $T' \in \hat{\mathcal{E}}$ there is a link in F from T' to $T \setminus T'$. TAP is also equivalent to the problem of augmenting

*Partially supported by NSF grant number 0728787.

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 best known approximation ratio for TAP is 1.5 [5, 12]. Several paper analysed integrality gaps of LP and SDP relaxations for the problem. In [13] is given a simple LP with gap 1.5 for LL-TAP – a special case of TAP where every link in E connects two leaves of T. In [13] is also given a 17/12-approximation algorithm for LL-TAP. Recently, Cheriyan and Gao [1] showed that for TAP, a certain SDP relaxation obtained by the Lasserre lift-and-project method has integrality gap $7/4+\epsilon$. This is the first approximation for general TAP with respect to some mathematical program. Here we show, by a simple and short proof, that a modification of the LP used in [13] has integrality gap 7/4, which is achieved by (a modification of) the combinatorial algorithm of [12].

In the more general Weighted TAP problem, the links in E have weights $\{w_e : e \in E\}$ and the goal is to find a minimum weight augmenting edge set $F \subseteq E$ such that $T \cup F$ is 2-edge connected. This problem admits several 2-approximation algorithms [6, 10, 7, 8]. The approximation ratio of 2 for all these algorithms is tight even for TAP. These algorithms achieve ratio 2 w.r.t. to the standard LP-relaxataion, that seeks to minimize $\sum_{e \in E} w_e x_e$ over the polyhedron:

$$\left\{x \in \mathbb{R}^E : x(\delta(A)) \geq 1 \ \forall A \in \hat{\mathcal{E}}, x_e \geq 0 \ \forall e \in E\right\}$$

where $\delta(A)$ denotes the set of links with exactly one endnode in A and $x(F) = \sum_{e \in E} x_e$. This LP has integrality gap at least 1.5 for TAP [3], and at most 5/3 for LL-TAP [13].

Breaking the ratio of 2 for Weighted TAP is a major open problem in network design, see the surveys [9, 11]. Several algorithms are known for special cases. In [2] is shown how to round a half-integral solution to the above LP within ratio 4/3. However, there are instances that do not have an LP optimal solutions which is half integral. In [4] is given an algorithm with ratio $(1+\ln 2)$ and running time $n^{f(D)}$ where D is the diameter of T. Studying various LP-relaxations for TAP is motivated by the hope that these may lead to breaking the ratio of 2 for Weighted TAP.

For $u, v \in V$ let $(u, v) \in \mathcal{E}$ denote the edge in T and uv the link in E between u and v. A link uv *covers* all the edges along the path $P(uv)$. The choice of the root 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 r' and its descendants is denoted by $T_{r'}$ (r' is the root of $T_{r'}$). A subtree T' of T is called a *rooted subtree* of T if $T' = T_{r'}$ for some $r' \in V$.

Definition 1.1 (shadow) *Let $P(uv)$ denote the path between u and v in T. A link $u'v'$ is a* shadow *of a link uv if $P(u'v') \subseteq P(uv)$.*

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 that the set of links E is closed under shadows.

Definition 1.2 (twin link and stem) *A link between two leaves of T is a* twin link *if its contraction results in a new leaf; the least common ancestor of the endnodes of a twin link is called a* stem. *Let W denote the set of twin links, and for $e \in W$ let s_e denote the stem of e.*

For $A, B \subseteq V$ and $F \subseteq E$ let $\delta_F(A, B)$ denote the set of links in F with one end in A and the other end in B, and let $\delta_F(A) = \delta_F(A, V \setminus A)$ denote the set of links in F with exactly one endnode in A. The default subscript in the above notation is E. Let L denote the set of leaves of T and let $\mathcal{O}_L = \{A \subseteq V : |A \cap L| \text{ is odd}\}$. For a function x on E and $F \subseteq E$ let $x(F) = \sum_{e \in F} x_e$.

Let Π be the polyhedron defined by the following set of linear constraints:

$$
\begin{aligned}
x(\delta(A)) &\geq 1 && \forall A \in \mathcal{E} && (1)\\
x(\delta(v)) &= 1 && \forall v \in L && (2)\\
x(\delta(A, V)) &\geq \lceil |A \cap L|/2 \rceil && \forall A \in \mathcal{O}_L && (3)\\
x_e - x(\delta(s_e)) &= 0 && \forall e \in W && (4)\\
x_e &\geq 0 && \forall e \in E && (5)
\end{aligned}
$$

Inequalities (1) and (5) are the constraints of a standard LP-relaxation for TAP. Validity of the constraints (2) and (3) was already observed in [13, 12], and we add over it the constraints in (4). Specifically, without the constraints in (4) the above LP was shown to have integrality gap 3/2 for LL-TAP, but for TAP the constraints in (4) are crucial.

Now we explain why the above LP is a relaxation for TAP.

Definition 1.3 (exact cover) *An edge set F is an* exact cover *of L if $|\delta_F(v)| = 1$ for all $v \in L$.*

Lemma 1.1 ([12]) *Given a* TAP *instance with shadow completion, let F be an optimal shadows-minimal solution with $|F \cap W|$ maximal. Then F is an exact cover of L, and for any $e = ab \in W$, either $e \in F$ and $|\delta_F(s_e)| = 1$, or $e \notin F$ and $|\delta_F(s_e)| = 0$.*

Let Π_L be the polyhedron defined by (2), (3), and (5). The extreme points of Π_L are the characteristic vectors of the exact edge-covers of L, see [14]; thus by Lemma 1.1, these constraints are valid. The validity of the constraints (4) follows also from Lemma 1.1. Consequently, the LP $\tau = \min\{x(E) : x \in \Pi\}$ is a relaxation for TAP. Combining techniques from [13, 12] and using some new methods, we prove the following.

Theorem 1.2 TAP *admits a polynomial time algorithm that computes a solution F such that $|F| \leq \frac{7}{4}\tau$.*

2 Proof of Theorem 1.2

Let $S = \{s_e : e \in W\}$ be the set of stems of T and let $R = V \setminus (L \cup S)$. Let $\rho \geq 1.5$ be a parameter, set later to $\rho = 7/4$. Define a weight function w on $E(L, V)$ by:

$$w_e = \begin{cases} \rho & \text{if } e \in \delta(L, L) \setminus W \\ \rho - \frac{1}{2} & \text{if } e \in \delta(L, V \setminus L) \\ \rho + \frac{1}{2} & \text{if } e \in W \end{cases}$$

Lemma 2.1 *Let F_L be a minimum weight exact cover of L and $x \in \Pi$ such that $x(E) = \tau$. Then:*

$$\rho\tau \geq w(F_L) + \frac{1}{2} \sum_{v \in R} x(\delta(v)). \tag{6}$$

Proof: Let x' be defined by $x'_e = x_e$ if $e \in \delta(L, V)$ and $x'_e = 0$ otherwise. Note that $x' \in \Pi_L$, since x satisfies (2), (3), and (5). Since F_L is an optimal (integral) exact cover of L with respect to the weights w_e and $x' \in \Pi_L$, $x' \cdot w \geq w(F_L)$. Thus to prove the lemma, it is sufficient to prove the following:

$$\rho\tau \geq x' \cdot w + \frac{1}{2} \sum_{v \in R} x(\delta(v)) \ .$$

Assign ρx_e tokens to every $e \in E$. The total amount of tokens is exactly $\rho x(E) = \rho\tau$. We will show that these tokens can be moved around such that the following holds:

(i) every $e \in \delta(L, L)$, and thus every $e \in W$, keeps its initial ρx_e tokens.

(ii) every $e \in \delta(L, V \setminus L)$ keeps $(\rho - \frac{1}{2})x_e$ tokens from its initial ρx_e tokens.

(iii) every $v \in R$ gets $\frac{1}{2}x_e$ tokens for each $e \in \delta(v)$.

(iv) every $e \in W$ gets additional $\frac{1}{2}x_e$ tokens, to a total of $(\rho + \frac{1}{2})x_e$ tokens.

This distribution of tokens is achieved in two steps. In the first step, for every $e \in E$, move $\frac{1}{2}x_e$ token from the ρx_e tokens of e to each non-leaf endnode of e, if any. Note that after this step, (i) and (ii) hold, and every $v \in V \setminus L$ gets $\frac{1}{2}x_e$ token for each $e \in \delta(v)$. In the second step, every $e \in W$ gets all the tokens moved at the first step to its stem s_e by the links in $\delta(s_e)$. The amount of such tokens is $\frac{1}{2}x(\delta(s_e)) = \frac{1}{2}x_e$, by (4). This gives an assignment of tokens as claimed, and thus the proof of the lemma is complete. $\square$

To prove Theorem 1.2 we prove the following.

Theorem 2.2 *For $\rho = 7/4$ there exist a polynomial time algorithm that given an instance of* TAP *computes a solution I of size at most the right-hand size of* (6). *Thus $|I| \leq \frac{7}{4}\tau$.*

In the rest of the paper we prove Theorem 2.2. The algorithm as in Theorem 2.2 is an adjustment of the 1.5-approximation algorithm of [12] for TAP. Let F_L be a minimum w-weight exact cover of L. Let $M = \delta_{F_L}(L, L)$. Initially, the algorithm assigns *coupons* to leaves unmatched by M and to links in M, such that the total number of coupons assigned equals $w(F_L)$. For technical reasons, we also assign 1 coupon to the root r of T.

Initial assignment of coupons: Every link in $M \setminus W$ gets ρ coupons, every link in $M \cap W$ gets $\rho + \frac{1}{2}$ coupons, every unmatched leaf gets $\rho - \frac{1}{2}$ coupons, and r gets 1 coupon. By Lemma 2.1, the total amount of coupons plus $\frac{1}{2}\sum_{v\in R} x(\delta(v))$ is at most $\rho\tau$.

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. 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.

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.

Definition 2.1 *Let $uv \in E$ such that 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. We have following properties of the matching M on leaves of T/I.

Lemma 2.3 *After all greedy contractions are exhausted, M is a matching on the original leaves (thus $x(\delta(v)) = 1$ for every node v matched by M) such that:*

(i) *No contraction of a link in M creates a new leaf; in particular, $M \cap W = \emptyset$.*
(ii) *If two leaves a, b are unmatched by M then $ab \notin E$.*

Proof: Property (M1) is proved in [12]. (M2) holds since any link $ab \in E$ such that a, b are unmatched by M gives a greedy contraction. □

Definition 2.2 ([12]) *A rooted subtree T' of T/I is M-*compatible *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'.*

We use the following notation for the credit distributed in a rooted subtree T' of T/I. Let *coupons*(T') denote the total number of coupons owned by nodes of T' and by links in M with both endnodes in T'. Let $R' = R \cap T'$ and $credit(T') = coupons(T') + \frac{1}{2}\sum_{v\in R'} x(\delta(v))$. In the next section we will prove the following key statement.

Lemma 2.4 *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 credit$(T') \geq |B'| + 1$.

Algorithm Tree-Cover (Algorithm 1) initiates $I \leftarrow \emptyset$ as a partial cover. It computes a minimum w-weight exact edge cover F_L of L, sets $M = \delta(F_L, F_L)$ and initiates the described credit scheme. In the main loop, the algorithm iteratively exhausts greedy contractions, then computes T', B' as in Lemma 2.4, 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. The credit scheme used implies that the algorithm computes a solution I of size at most ρ times the right-hand size of (6). Hence it only remains to prove Lemma 2.4.

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

initialize: $I \leftarrow \emptyset$
$F_L \leftarrow$ minimum w-weight exact edge cover of L, $M \leftarrow \delta_{F_L}(L, L)$.
Give ρ coupons to every link in $M \setminus W$, give $\rho + \frac{1}{2}$ coupons to every link in $M \cap W$, give $\rho - \frac{1}{2}$ coupons to every unmatched leaf, and give 1 coupon to r.
while *T/I has more than one node* **do**
  Exhaust greedy contractions and update I and M accordingly.
  Find a subtree T' of T/I and a cover B' of T' as in Lemma 2.4.
  Contract T', assign 1 coupon to the new compound leaf, and set $I \leftarrow I \cup B'$.
return I

3 Proof of Lemma 2.4

The *up-link* $up(a)$ of a node a is the link au such that u is as close as possible to the root; assuming shadows completion, u is an ancestor of a. For $U \subseteq V$, we let $up(U) = \{up(u) : u \in U\}$. A rooted subtree T' of T is *a-closed* if $up(a)$ belongs to T', and T' is *a-open* otherwise.

Definition 3.1 (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.*

In what follows, let us use the following notation:

- $L' = L(T')$ is the set of leaves of T' and $S' = S(T')$ is the set of stems of T'.
- $R' = V(T') \setminus (L' \cup S')$ and $\Sigma = \sum_{v \in R'} x(\delta(v))$.
- $M' = M(T')$ is the set of links in M with both endnodes in T'.
- U' is the set of leaves of T' unmatched by M, and U'_0 is the set of the original leaves in U'.
- C' is the set of *non-leaf* compound nodes of T' (recall that this includes r, if $r \in T'$).
- $B(T') = M' \bigcup up(U')$ (hence $|B(T')| = |M'| + |U'| = |L'| - |M'|$).

Lemma 3.1 ([12]) *If T' is minimally semi-closed then $B(T')$ is an exact cover of T'.*

Definition 3.2 (deficient tree) *A semi-closed tree T' is* deficient *if $credit(T') < |B(T')| + 1$.*

Definition 3.3 (dangerous tree) *A semi-closed tree T' is* dangerous *if $|C'| = |S'| = 0$, $|M'| = 1$, $|L'| = 3$, and 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 T' is b-open.*

We prove the following.

Lemma 3.2 *Any deficient tree is dangerous.*

Note that we can check in polynomial time whether a given rooted subtree of T/I is 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 2.4. Thus we will assume that all minimally semi-closed subtrees of T/I are dangerous. In this case, it is shown in [12] that Algorithm 2 below finds an M-compatible tree T' and its cover B' such that $credit(T') \geq |B'| + 1$.

Algorithm 2: FIND-TREE($T = (V, \mathcal{E}), E, M$) (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 $\tilde{M}$ be a matching on the leaves of T/I obtained from M by replacing the link $e = bb'$ by the link $\tilde{e} = ab'$ in every dangerous tree T_0 of T.
2. Let T' be a minimally semi-closed tree w.r.t. the matching $\tilde{M}$.
3. **return** T' *and* $B' = \tilde{M}(\tilde{T}') \cup up(\tilde{U}')$, where $\tilde{U}'$ is the set of unmatched leaves of T' w.r.t. $\tilde{M}$.

The proof of Lemma 2.4 is now complete. It remains only to prove Lemma 3.2.

4 Proof of Lemma 3.2

Let T' be a deficient tree with root r'. We will show that T' is dangerous. Note that $|B(T')| = |M'| + |U'|$ and that $coupons(T') = \rho|M'| + |U'| + (\rho - \frac{1}{2})|U'_0| + |C'|$. Hence for $\rho = 7/4$ we have:

$$coupons(T') - |B(T')| = (\rho - 1)|M'| + (\rho - \frac{3}{2})|U'_0| + |C'| = \frac{7}{4}|M'| + \frac{1}{4}|U'_0| + |C'| \ . \qquad (7)$$

Claim 4.1 *$|C'| = 0$, $|M'| \leq 1$, and $|S'| = 0$.*

Proof: From (7) we immediately get that $|C'| = 0$, $|M'| \leq 1$, and if $|M'| = 1$ then $|U'_0| = 0$. To see that $|S'| = 0$, note that if $|S'| \geq 1$, then Lemma 2.3 implies the contradiction $|M'| = 1$ and $|U'_0| \geq 1$. ☐

Claim 4.2 *$\Sigma \geq |U'| + 1 - 2|M'|$. Thus $|M'| = 1$, $\Sigma < 1/2$, and $|L'| = 3$.*

Proof: Let $E' = \delta(U') \cup \delta(T')$. Note that $x(E') \geq |U'| + 1$. If $e \in \delta(U')$, then e contributes x_e to Σ, unless e is incident to a matched leaf. However, $x(\delta(b)) = 1$ for every matched leaf b, and the number of matched leaves in T' is $2|M'|$. Hence $\Sigma \geq x(E') \geq |U'| + 1 - 2|M'|$, as claimed.

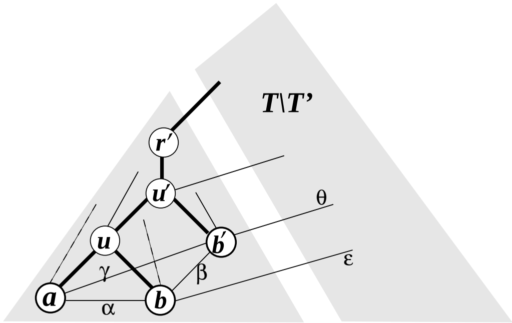

Figure 1: Illustration to the proof of Lemma 3.2 (links in E_1 are shown by dashed lines).

We prove the second statement. If $|M'| \neq 1$, then since $|M'| \leq 1$, we must have $|M'| = 0$; this implies $\Sigma \geq |U'| + 1 \geq 2$, which gives the contradiction $credit(T') - |B(T')| \geq \frac{1}{2}\Sigma \geq 1$. Thus $|M'| = 1$. If $\Sigma \geq \frac{1}{2}$, then $credit(T') - |B(T')| \geq \frac{7}{4}|M'| + \frac{1}{2}\Sigma \geq 1$. We show that $|L'| = 3$. If $|L'| \leq 2$ then $|M'| = 0$, as otherwise we get a contradiction to Lemma 2.3(i). Also, $|L'| \geq 4$ is not possible, since then $|U'| \geq 2$, which implies the contradiction $\Sigma \geq |U'| + 1 - 2|M'| \geq 1$. □

Let bb' be the matched pair and a the unmatched leaf of T'. Let u and u' be the least common ancestor of ab and ab', respectively, and assume w.l.o.g. that u is a descendant of u' (see Fig. 1). Let $x_{ab} = \alpha$, $x_{bb'} = \beta$, $x_{ab'} = \gamma$, $x(\delta(b, T \setminus T') = \epsilon$, and $x(\delta(b', T \setminus T') = \theta$. To finish the proof of the lemma, it is sufficient to show the following:

- If $u \neq u'$, then $\gamma > 0$ and $\epsilon > 0$.
- If $u = u'$, then at least one of the following holds: $\gamma, \epsilon > 0$ or $\alpha, \theta > 0$.

Indeed, if $u \neq u'$, then $\gamma > 0$ implies that the link ab' exists, and $\epsilon > 0$ implies that T' is b-open. Thus, by the definition, T' is dangerous. The same holds if $u = u'$ and $\gamma, \epsilon > 0$. If $u = u'$ and $\alpha, \theta > 0$, then ab exists (since $\alpha > 0$) and T' is b'-open (since $\theta > 0$); thus by exchanging the roles of b, b' we get that T' is dangerous, by the definition. Consider the following links sets:

- $E_1 = \delta(\{a, b\}, R') = x(\delta(a, R')) + x(\delta(b, R'))$ are the links from a, b to R'.
- $E_2 = \delta(T_u \setminus \{a, b\}, T \setminus T_u) = \delta(T_u \cap R', T \setminus T_u)$ are the links from $T_u \cap R'$ to nodes outside T_u.
- $E_3 = \delta(R', T \setminus T')$ are the links from R' to nodes outside T'.

Any $e \in E_1 \cup E_2 \cup E_3$ contributes x_e to Σ. Recalling that $x(\delta(b)) = x(\delta(b')) = 1$ and $x(\delta(a)) \geq 1$, it is easy to verify that:

(i) $\Sigma \geq x(E_1) + x(E_2) = x(\delta(T_u)) - \gamma \geq 1 - \gamma$ if $u \neq u'$.
The first inequality is since every $e \in E_1 \cup E_2$ contributes x_e to Σ and since $E_1 \cap E_2 = \emptyset$.

(ii) $\Sigma \geq x(E_1) = x(\delta(a, R')) + x(\delta(b, R')) \geq x(\delta(a, R')) \geq x(\delta(a)) - \alpha - \gamma \geq 1 - \alpha - \gamma$.

(iii) $\Sigma \geq x(E_3) \geq x(\delta(T', T \setminus T')) - \theta - \epsilon \geq 1 - \theta - \epsilon$.

(iv) $\alpha + \epsilon \leq 1$ (since $x(\delta(b)) = 1$) and $\gamma + \theta \leq 1$ (since $x(\delta(b')) = 1$).

Suppose that $u \neq u'$. Then $1/2 > \Sigma \geq 1 - \gamma$, by (i); hence $\gamma > 1/2$. If $\epsilon = 0$ then (iii) implies $\theta > 1/2$, and we obtain a contradiction to (iv) $\gamma + \theta > 1$. Thus, $\gamma, \epsilon > 0$ in this case, as claimed.

Suppose that $u = u'$. By (ii) and (iii), $\alpha + \gamma > 1/2$ and $\theta + \epsilon > 1/2$. One can easily verify that combined with (iv) this gives that we must have $\gamma > 0$ and $\epsilon > 0$, or $\alpha > 0$ and $\theta > 0$, as claimed.

This concludes the proof of Lemma 3.2.

References

[1] J. Cheriyan and Z. Gao. Approximating unweighted tree augmentation via lift-and-project. Manuscript, 2014.

[2] J. Cheriyan, T. Jordán, and R. Ravi. On 2-coverings and 2-packing of laminar families. In *ESA*, pages 510–520, 1999.

[3] J. Cheriyan, H. Karloff, R. Khandekar, and J. Koenemann. On the integrality ratio for tree augmentation. *Oper. Res. Lett.*, 36(4):399–401, 2008.

[4] N. Cohen and Z. Nutov. A $(1 + \ln 2)$-approximation algorithm for minimum-cost 2-edge-connectivity augmentation of trees with constant radius. *Theor. Comput. Sci.*, 489-490:67–74, 2013.

[5] G. Even, J. Feldman, G. Kortsarz, and Z. Nutov. A 3/2-approximation for augmenting a connected graph into a two-connected graph. In *APPROX*, pages 90–101, 2001.

[6] G. N. Frederickson and J. Jájá. Approximation algorithms for several graph augmentation problems. *SIAM J. Computing*, 10:270–283, 1981.

[7] M. Goemans, A. Goldberg, S. Plotkin, E. T. D. Shmoys, and D. Williamson. Improved approximation algorithms for network design problems. In *SODA*, pages 223–232, 1994.

[8] K. Jain. A factor 2 approximation algorithm for the generalized steiner network problem. *Combinatorica*, 21(1):39–60, 2001.

[9] S. Khuller. Approximation algorithms for finding highly connected subgraphs (chapter 6). In *Approximation algorithms for NP-hard problems (Ed. D. S. Hochbaum)*. PWS, Boston, 1996.

[10] S. Khuller and R. Thurimella. Approximation algorithms for graph augmentation. *J. of Algorithms*, 14:214–225, 1993.

[11] G. Kortsarz and Z. Nutov. Approximating minimum cost connectivity problems (chapter 58). In *Handbook of Approximation Algorithms and Metahueristics (Ed. T. F. Gonzales)*. Chapman & Hall/CRC, 2007.

[12] G. Kortsarz and Z. Nutov. A simplified 1.5-approximation algorithm for augmenting edge-connectivity of a graph from 1 to 2. Manuscript, 2014.

[13] Y. Maduel and Z. Nutov. Covering a laminar family by leaf to leaf links. *Discrete Applied Mathematics*, 158(13):1424–1432, 2010.

[14] A. Schrijver. *Combinatorial Optimization, Polyhedra and Efficiency.* Springer-Verlag Berlin, Heidelberg New York, 2004.