LP-Relaxations for Tree Augmentation*

Guy Kortsarz[1] and Zeev Nutov[2]

1 **Rutgers University, Camden, NJ, USA**
guyk@camden.rutgers.edu
2 **The Open University of Israel, Ra'anana, 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 approximation ratio known for TAP is 1.5. In the more general Weighted TAP problem, F should be of minimum weight. Weighted TAP admits several 2-approximation algorithms w.r.t. to the standard cut-LP relaxation. The problem is equivalent to the problem of covering a laminar set family. Laminar set families play an important role in the design of approximation algorithms for connectivity network design problems. In fact, Weighted TAP is the simplest connectivity network design problem for which a ratio better than 2 is not known. Improving this "natural" ratio is a major open problem, which may have implications on many other network design problems. It seems that achieving this goal requires finding an LP-relaxation with integrality gap better than 2, which is an old open problem even for TAP. In this paper we introduce two different LP-relaxations, and for each of them give a simple algorithm that computes a feasible solution for TAP of size at most 7/4 times the optimal LP value. This gives some hope to break the ratio 2 for the weighted case.

1998 ACM Subject Classification G.2.2 Graph Theory: Graph algorithms

Keywords and phrases Tree Augmentation, LP-relaxation, Laminar family, Approximation algorithms

Digital Object Identifier 10.4230/LIPIcs.APPROX-RANDOM.2016.13

1 Introduction

1.1 Problem definition and related problems

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 $G = (V, E_G)$ and a set of additional edges (called "links") E on V disjoint to E_G, find a minimum size edge set $F \subseteq E$ such that $G + F = (V, E_G \cup F)$ is 2-edge-connected. Contracting the 2-edge-connected components of the input graph G results in a tree. Hence, our problem is:

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

* This work was partially supported by NSF grant number 434923.

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2016).
Editors: Klaus Jansen, Claire Matthieu, José D. P. Rolim, and Chris Umans; Article No. 13; pp. 13:1–13:16
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

TAP can be formulated as the problem of covering the edges of a tree by paths. For $u, v \in V$ let $(u, v) \in E_T$ denote the edge in T and uv the link in E between u and v. Let $P(uv) = P_T(uv)$ denote the path between u and v in T. A link uv *covers* all the edges along the path $P(uv)$. Then TAP is the problem of finding a minimum subset of (paths of the) links that cover the edges of T.

TAP can be also formulated as the problem of covering a laminar set family. In what follows, root T at some node r. 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 $P(ru)$. The *rooted subtree* of T induced by v and its descendants is denoted by T_v (v is the root of T_v). Let $\mathcal{T} = \{T_v : v \in V \setminus \{r\}\}$. The family of node sets of the trees in $\mathcal{T}$ is laminar, and $F \subseteq E$ is a feasible solution for TAP if and only if F covers $\mathcal{T}$, namely, for every $T' \in \mathcal{T}$ there is a link in F from T' to $T \setminus T'$.

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.

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. Even a more general problem is the 2-Edge-Connected Subgraph problem, where the goal is to find a spanning 2-edge-connected subgraph of a given weighted graph; Weighted TAP is a particular case, when the input graph contains a connected spanning subgraph of cost zero.

In this paper we introduce new LP-relaxations for TAP (that are also valid for Weighted TAP) and prove that their integrality gap for TAP less than 2.

1.2 Previous and related work

TAP is NP-hard even for trees of diameter 4 [9], or when the set E of links forms a cycle on the leaves of T [3]. The first 2-approximation for Weighted TAP was given 24 years ago in 1981 by Fredrickson and Jájá [9], and was simplified later by Khuller and Thurimella [15]. These algorithms reduce the problem to the Min-Cost Arborescence problem, that is solvable in polynomial time [5], while invoking a factor of 2 in the ratio. The primal-dual algorithm of [12, 11] is another combinatorial 2-approximation algorithm for the problem. The iterative rounding algorithm of Jain [13] is an LP-based 2-approximation algorithms. These algorithms achieve ratio 2 w.r.t. to the standard **cut-LP** that seeks to minimize $\sum_{e \in E} w_e x_e$ over the following polyhedron:

$$x_e \geq 0 \quad \forall e \in E \tag{1}$$

$$x(\delta(T')) \geq 1 \quad \forall T' \in \mathcal{T} \tag{2}$$

Here $\delta(T')$ is the set of links with exactly one endnode in T', $x(F) = \sum_{e \in F} x_e$ is the sum of the variables indexed by the links in F, and $\mathcal{T}$ is the set of proper rootes subtrees of T w.r.t. the chosen root r.

Laminar set families play an important role in the design and analysis of exact and approximation algorithms for network design problems, both in the primal-dual method and the iterative rounding method, c.f. [17, 12]. Weighted TAP is the simplest network design problem for which a ratio better than 2 is not known. Breaking the "natural" ratio of 2 for Weighted TAP is a major open problem in network design, that may have implications on other problems.

As a starting point, Khuller [14] in his survey on high connectivity network design problems posed as a major open question achieving ratio better than 2 for TAP. Nagamochi [19] used a novel lower bound to achieve ratio $1.875 + \epsilon$ for TAP. The sequence of papers

[7, 8, 16] introduced additional new techniques to achieve ratio 1.8 by a much simpler algorithm and analysis, and also achieved the currently best known ratio 1.5.

Several algorithms for Weighted TAP with ratio better than 2 are known for special cases. In [6] is given an algorithm with ratio $(1 + \ln 2)$ and running time $n^{f(D)}$ where D is the diameter of T. In [3] it is shown how to round a half-integral solution to the cut-LP within ratio 4/3. However, as is pointed in [3], the cut-LP LP has extreme points which are not half integral.

Studying various LP-relaxations for TAP is motivated by the hope that these may lead to breaking the ratio of 2 for Weighted TAP. Thus several paper analyzed integrality gaps of LP/SDP relaxations for the problem. Cheriyan, Karloff, Khandekar, and Koenemann [4] gave an example of a TAP instance with integrality gap 1.5 w.r.t. a standard cut-LP. For the special case of TAP when every link connects two leaves, [18] obtained ratios 5/3 w.r.t. the cut LP, ratio 3/2 w.r.t. to a strengthened "leaf edge-cover" LP, and ratio 17/12 not related to any LP. However, the analysis of [18] does not extend directly to the general TAP. Cheriyan and Gao [1] showed that the 1.8-approximation algorithm of [8] achieves its ratio 1.8 w.r.t. an SDP relaxation obtained by Lasserre tightening of a standard LP supplemented by so called "non-overlapping" constraints. Recently in [2] they improved their analysis, showing that the 1.5-approximation ratio of [16] is achievable w.r.t. this SDP. However, the SDP and the analysis in [2] are quite involved, and hence might be very hard to extend to Weighted TAP.

Finally, we mention some work on the closely related 2-Edge-Connected Subgraph problem. This problem was also vastly studied. For general weights, the best known ratio is 2 by Fredrickson and Jájá [9], which can also be achieved by the algorithms in [15] and [13]. For particular cases, better ratios are known. Fredrickson and Jájá [10] showed that when the edge weights satisfy the triangle inequality, the Christofides heuristic has ratio 3/2. For the special case when all the edges of the input graph have unit weights (the "min-size" version of the problem), the currently best known ratio is 4/3 due to Sebo and Vygen [21].

1.3 Our results

In this paper, with the help of some ideas from [7, 18, 8, 16], we introduce two simple new LP-relaxations, and prove that they have integrality gap at most 7/4 for TAP. This is the first LP-relaxation for TAP for which integrality gap less than 2 is proved. We note that our algorithms use several ideas from [7, 8], but they are *not* identical to any previous algorithm.

The dual-fitting algorithm is our main result. This is essentially a primal-dual algorithm when we allow to violate the dual constraints by a factor of 7/4. Unlike *all* previous algorithms, we do not need to compute a maximum matching on the leaves, but pick some inclusionwise maximal matching. The algorithm uses some simple local steps and is faster than all previous algorithms – the running time is $\tilde{O}(mn)$. Since the algorithm and the LP are quite simple, it has a chance to be extended to Weighted Tap. We note that one type of constrainst we use, see constraints (3), can be naturally derived from the so called "subpartiton constrains" or "Gomory cuts", which say that k disjoint sets need at least $\lceil k/2 \rceil$ edges to cover them. The constraints (3) are derived from the case $k = 3$ when applied to the two lowest levels of the tree – leaves and stems.

2 New valid constraints

In this section we introduce new LP-relaxations for TAP and in subsequent sections prove that (for the unweighted case) they both have integrality gap 7/4. Our LP-relaxations combine some ideas from [18, 8, 16], but also use new crucial valid constraints. We need some definition to introduce these constraints.

▶ **Definition 1** (shadow, shadow-minimal cover)**.** Let $P(uv)$ denote the path between u and v in T. A link $u'v'$ is a **shadow** of a link uv if $P(u'v') \subseteq P(uv)$. A cover F of T is **shadow-minimal** if for every link $uv \in F$ replacing uv by any proper shadow of uv results in a set of links that does not cover T.

We refer to the addition of all shadows of existing links 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:

The Shadow-Completion Assumption. The set of links E is closed under shadows.

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. To **contract** a subtree T' of T is to combine the nodes in T' into a new 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.

▶ **Definition 2** (leaf, twin link, stem)**.** 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. A link $ab \in \delta(L, L)$ is a **twin link** and the least common ancestor s of a, b is a **stem** if the contraction of T_s results in a new leaf; such a, b are called **twins**. Let W denote the set of twin links, and for $e \in W$ let s_e denote the stem of e.

For $A \subseteq V$, we say that a rooted subtree T' of T is **A-closed** if there is no link in E from $A \cap T'$ to $T \setminus T'$, and T' is **A-open** otherwise.

▶ **Definition 3** (locked node, locking link, dangerous locking tree)**.** A node a (or a subtree T_a) is **locked** by a link $bb' \in \delta(L, L)$ and bb' is the **locking link of** a if (see Fig. 1(a)) the tree obtained from T by contracting T_a into the node a has a rooted proper subtree $T' = T_{r'}$ that is a-closed such that $L(T') = \{a, b, b'\}$; such minimal T' is called the **locking tree** of a (note that such locking tree is unique); a locking tree is a **dangerous locking tree** if it is as in Fig. 1(b) with the links depicted present in E; namely, a locking tree is dangerous if there exists an ordering b, b' of the locking link endnodes such that:

- The contraction of ab' does not create a new leaf.
- $ab' \in E$.
- T' is b-open.

Let $\mathcal{N}$ denote the set of non-dangerous locking trees.

Note that an ordering b, b' as in the above definition may not be unique; namely, it may be that also the contraction of ab does not create a new leaf, $ab \in E$, and T' is b'-open – see Fig. 1(c).

In what follows, let us use the following notation:

- For a stem s let $\sigma(s)$ denote the set of links in $\delta(s)$ that have an endnode not in T_s.
- For $T' \in \mathcal{N}$ let $\zeta(T')$ denote the set of links incident to some non-leaf node of T'.
- Let $\mathcal{O}_L = \{A \subseteq V : |A \cap L| \text{ is odd}\}$.
- For $x \in \mathbb{R}^E$ and $F \subseteq E$ let $x(F) = \sum_{e \in F} x_e$.

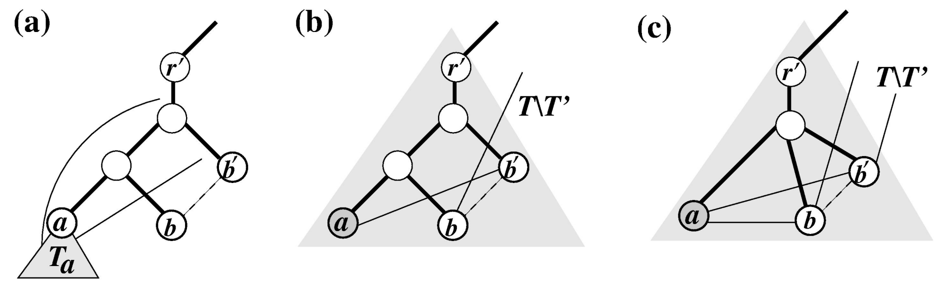

Figure 1 (a) A locking tree; no link with an endnode in T_a has its other endnode in $T \setminus T_{r'}$. (b,c) Dangerous trees; solid thin lines show links that must exist in E. The endnodes b, b' of the locking link are original leaves; in (a), a is an original leaf, and in (b),(c) the subtree T_a is contracted into a, so a may be a compound node or an original leaf. Some of the edges of T can be paths.

The proof of the following statement can be found in [8, 16]; we provide a proof sketch for completeness of exposition.

▶ **Lemma 4.** *Let F be a shadow-minimal cover of T. Then the following holds:*
(i) $\delta_F(L, V)$ *is an exact edge-cover of L, namely $|\delta_F(v)| = 1$ for every $v \in L$.*
(ii) *If $e \in F \cap W$ then $|\sigma(s_e) \cap F| = 1$.*
(iii) $\zeta(T') \cap F \neq \emptyset$ *for any $T' \in \mathcal{T}$.*

Proof. Let us say that two links **overlap** if their paths share an edge and one contains an end of the other. It is easy to see that F is not shadow minimal if and only if two links in F overlap. As any two links incident to the same leaf overlap, (i) follows.

Now let $e \in F \cap W$ and consider a link f that covers the parent edge of the stem s_e of e. It is easy to see that the only case that e and f do not overlap is if $f \in \sigma(s_e)$, and that any two links in $\sigma(s_e)$ overlap. This implies (ii).

Let T' be a locking tree as in Definition 3 (after T_a is contracted into a). We will show that if $\zeta(T) \cap F = \emptyset$ then T' is dangerous. Consider a link $e = au$ that covers the parent edge of a and a link $e' = u'v$ that covers the parent edge of T', where $v \notin T'$. Note that $e \neq e'$, since T' is a-closed. If $\zeta(T) \cap F = \emptyset$ then $\{u, u'\} \subseteq \{b, b'\}$, and since by (i) $|\delta_F(b)| = |\delta_F(b')| = 1$, we must have $\{u, u'\} = \{b, b'\}$. If $u = b$ and contraction of ab creates a new leaf, then the link in F that covers the parent edge of this new leaf belongs to $\zeta(T)$. Otherwise, T' must be dangerous, as claimed. ◀

Now we present our new valid inequalities for TAP.

▶ **Lemma 5.** *Suppose that the Shadow-Completion Assumption holds, and let x be the characteristic vector of a shadow minimal cover F of T. Then x satisfies the following constraints*

$$x(\sigma(s_e)) - x_e \geq 0 \qquad \forall e \in W \tag{3}$$

$$x(\zeta(T')) \geq 1 \qquad \forall T' \in \mathcal{T} \tag{4}$$

$$x(\delta(v)) = 1 \qquad \forall v \in L \tag{5}$$

$$x(\delta(A, V)) \geq \left\lceil \frac{|A \cap L|}{2} \right\rceil \qquad \forall A \in \mathcal{O}_L \tag{6}$$

Proof. Consider the polyhedron Π_L defined by (1), (5), and (6). Then Π_L is the convex hull of the exact edge-covers of L, see [20, Theorem 34.2]; thus by Lemma 4(i), these constraints are valid. The validity of the constraints (3) follows from Lemma 4(ii) (in fact, $x(\sigma(s_e)) = x_e$ holds), and the validity of the constraints (4) follows from Lemma 4(iii). ◀

In subsequent sections we will consider two LPs, where both have the constraints (1), (2), and (3). One LP has an additional constraint (4), while the other LP has additional constraints (5) and (6) instead.

3 The algorithms

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. We often refer to the contraction of every 2-edge-connected component of $T \cup I$ into a single node 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**, and denote by C the set of compound nodes of T/I. Non-compound nodes are referred to as **original nodes** (of T). For technical reasons, the root r is also considered as a compound node, hence initially $C = \{r\}$.

Our algorithms start with a partial solution $I = \emptyset$ and with a certain matching $M \subseteq \delta(L, L) \setminus W$. We denote by U the set of leaves of T/I unmatched by M. The algorithm iteratively finds a subtree T' of T/I and a cover I' of T', and **contracts** T' **with** I', which means adding I' to I and contracting T' into a new compound node. To use the notation T/I properly, we will assume that I' is an exact cover of T', namely, that the set of edges of T/I that is covered by I' equals the set of edges of T' (this is possible due to shadow completion).

Another property of a contracted tree T' is given in the following definition.

▶ **Definition 6** (M-compatible subtree)**.** Let M be a matching on the leaves of T/I. A 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'. We say that a contraction of T' with I' is M-compatible if T' is M-compatible.

Assuming all compound nodes were created by M-compatible contractions, then the following type of contractions is also M-compatible.

▶ **Definition 7** (greedy contraction)**.** Adding to the partial solution I a link with both endnodes in U 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 now describe a more complicated type of M-compatible contractions.

▶ **Definition 8** (semi-closed tree)**.** Let M be a matching on the leaves of T/I. A rooted subtree T' of T/I is **semi-closed** (w.r.t. M) 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.

For a semi-closed subtree T' of T/I let us use the following notation:

- M' is the set of links in M with both endnodes in T'.
- U' is the set of leaves of T' unmatched by M.

Our algorithms maintain the following invariant:

Algorithm 1: DUAL-FITTING($T = (V, \mathcal{E}), E$) (ratio: $\rho = 7/4$)

1 **initialize:** $I \leftarrow \emptyset$, $C \leftarrow \{r\}$.
2 $M \leftarrow$ maximal matching in $\delta(L, L) \setminus W$, $U \leftarrow$ leaves unmatched by M.
3 Contract every link $ab \in W$ with $a, b \in U$.
4 Exhaust greedy contractions and update I, C accordingly.
5 **while do**
6 $\quad$ T/I has more than one node
7 Find T', I' as in Lemma 10.
8 Contract T' with I'.
9 Exhaust greedy contractions and update I, C accordingly.
10 **return** I

Partial Solution Invariant. The partial solution I is obtained by sequentially applying a greedy contraction or a legal semi-closed tree contraction with an exact cover.

▶ **Definition 9** (dangerous semi-closed tree)**.** A semi-closed subtree of T/I is dangerous (w.r.t. a matching M) if it is as in Definition 3 with $bb' \in M$.

In [8, 16] the following is proved:

▶ **Lemma 10** ([8, 16])**.** *Suppose that the Partial Solution Invariant hold for T, M, and I, and that T/I has no greedy contraction. Then there exists a polynomial time algorithm that finds a non-dangerous semi-closed tree T' of T/I and an exact cover $I' \subseteq E$ of T' of size $|I'| = |M'| + |U'|$.*

A formal description of the algorithms is given in Algorithms 1 and 2. Algorithm 1 and its dual-fitting analysis are our main results, since they are relatively simple and new. Our algorithms differ from previous algorithms in the matching M computed at step 2. In Algorithm 1 the matching M is only required to be *inclusion maximal*, while all previous algorithms computed a *maximum size* matching in $\delta(L, L) \setminus W$. This is a substantioal difference, since otherwise, to perform an LP-based analysis, one needs to add the constraints (6), as we will do in the analysis of Algorithm 2.

Our algorithms are supplemented by an *LP-based analysis* to achieve ratios better than 2 w.r.t. to the following two linear programs (LP1) and (LP2), where

- (LP1) is defined by the constraints (1), (2), (3), and (4).
- (LP2) is defined by the constraints (1), (2), (3), (5), and (6).

$$
\textbf{(LP1)} \quad
\begin{array}{lllr}
\min & x(E) & & \\
\text{s.t.} & x_e \geq 0 & \forall e \in E & (1) \\
& x(\delta(T')) \geq 1 & \forall T' \in \mathcal{T} & (2) \\
& x(\sigma(s_e)) - x_e \geq 0 & \forall e \in W & (3) \\
& x(\zeta(T')) \geq 1 & \forall T' \in \mathcal{N} & (4)
\end{array}
$$

$$
\textbf{(LP2)} \quad
\begin{array}{lllr}
\min & x(E) & & \\
\text{s.t.} & x_e \geq 0 & \forall e \in E & (1) \\
& x(\delta(T')) \geq 1 & \forall T' \in \mathcal{T} & (2) \\
& x(\sigma(s_e)) - x_e \geq 0 & \forall e \in W & (3) \\
& x(\delta(v)) = 1 & \forall v \in L & (5) \\
& x(\delta(A, V)) \geq \lceil |A \cap L|/2 \rceil & \forall A \in \mathcal{O}_L & (6)
\end{array}
$$

Algorithm 2: PRIMAL-FITTING($T = (V, \mathcal{E}), E$) (ratio: $\rho = 7/4$)

initialize: $C \leftarrow \{r\}$.

$F_L \leftarrow$ min-w-weight exact edge-cover of L, $w_e = \begin{cases} \rho & e \in \delta(L, L) \setminus W \\ \rho - \frac{1}{2} & e \in \delta(L, V \setminus L) \\ \rho + \frac{1}{2} & e \in W \end{cases}$

$M \leftarrow \delta_{F_L}(L, L)$, $U \leftarrow$ the set leaves of T unmatched by M

$I \leftarrow M \cap W$, $M \leftarrow M \setminus W$.

Exhaust greedy contractions and update I, C accordingly.

while do

T/I has more than one node

Find T', I' as in Lemma 10.

Contract T' with I'.

Exhaust greedy contractions and update I, C accordingly.

return I

▶ **Theorem 11.** *Algorithm 1 computes a solution I of size at most $7/4$ times the optimal value of (LP1).*

▶ **Theorem 12.** *Algorithm 2 computes a solution I of size at most $7/4$ times the optimal value of (LP2).*

4 Dual-fitting analysis of Algorithm 1 (Theorem 11)

For a link $e \in E$ let us use the following notation:

- $\delta_{\mathcal{T}}^{-1}(e) = \{T' \in \mathcal{T} : e \in \delta(T')\}$; recall that $\mathcal{T}$ is the family of proper rooted subtrees of T.
- $\sigma_S^{-1}(e) = \{s \in S : e \in \sigma(s)\}$; recall that S is the set of stems of T.
- $\zeta_{\mathcal{N}}^{-1}(e) = \{T' \in \mathcal{N} : e \in \zeta(T')\}$; recall that $\mathcal{N}$ is the family of non-dangerous locking trees.

With this notation, the dual LP of (LP1) is:

$$\textbf{(D)} \quad \begin{array}{lll} \max & y(\mathcal{E}) + q(\mathcal{T}) & \\ \text{s.t.} & y(\delta_{\mathcal{T}}^{-1}(e)) + z(\sigma_S^{-1}(e)) - |\{e\} \cap W| z_e + q(\zeta_{\mathcal{N}}^{-1}(e)) \leq 1 & \forall e \in E \\ & y_{T'} \geq 0 & \forall T' \in \mathcal{T} \\ & z_w \geq 0 & \forall w \in W \\ & q_{T'} \geq 0 & \forall T' \in \mathcal{N} \end{array}$$

We rewrite Algorithm 1 with the updates of the dual variables as Algorithm 3.

Note that every compound node v of T/I is obtained by contracting some (not necesarly rooted) subtree T' of T, and that every compound leaf v of T/I is obtained by contracting a rooted subtree of T. Thus in the algorithm, assigning value y_v to a compound leaf v of T/I means that we assign value y_v to the subtree that was contracted into v. During the algorithm, every non-zero dual variable corresponds to some node v of T/I; if v is an original leaf then this variable is y_v, and if v is a compound node then these are the variables of subtrees contracted into v. This is so since in the "while loop" of the algorithm, immediately after some dual variable is raised, the entire subtree corresponding to this variable is contracted into a compound node.

▶ **Definition 13.** During the algorithm, the **dual load** $\mu(e)$ of a link e is defined as the sum of the dual variables in the constraint of e in the dual program, namely

$$\mu(e) = y(\delta_{\mathcal{T}}^{-1}(e)) + z(\sigma_S^{-1}(e)) - |\{e\} \cap W| z_e + q(\zeta_{\mathcal{N}}^{-1}(e)) \ .$$

Algorithm 3: DUAL-UPDATE$(T = (V, \mathcal{E}), E)$ (ratio: $\rho = 7/4$)

1 **initialize:** $C \leftarrow \{r\}$;
$y \leftarrow 0$, $z \leftarrow 0$, $q \leftarrow 0$.
2 $M \leftarrow$ maximal matching in $E(L, L) \setminus W$, $U \leftarrow$ leaves unmatched by M.
$y_v \leftarrow 1$ if $v \in U$, $y_v \leftarrow \rho - 1$ if $v \in L \setminus U$.
$z_e \leftarrow \rho - 1$ for every link $e = ab \in W$ with $a, b \in U$.
3 $I \leftarrow M \cap W$, $M \leftarrow M \setminus W$.
4 Exhaust greedy contractions and update I, C accordingly.
5 **while do**
6 T/I has more than one node
7 Find T', I' as in Lemma 10.
Case 1: $|C'| = 0$ and either: $|M'| = 0$ or $|M'| = 1, |U'| \geq 2$
$y_{T'} \leftarrow \rho - 1$
$y_v \leftarrow \rho - 1$ if $v \in U'$ and $y_v \leftarrow 0$ if $v \in L' \setminus U'$.
Case 2: $|C'| = 0$ and $|M'| = |U'| = 1$ (so $T' \in \mathcal{N}$)
$q_{T'} \leftarrow \rho - 1$
8 Contract T' with I'.
9 Exhaust greedy contractions and update I, M, C accordingly.
10 **return** I

The **dual credit** $\pi(v)$ of a node v is defined as follows. Let $\pi'(v)$ be the sum of the dual variables y and q that correspond to v minus the number of links used by the algorithm to contract the corresponding tree into v. Then $\pi(v) = \pi'(v)$ if v does not contain r, and $\pi(v) = \pi'(v) + 1$ otherwise.

Note that the dual load of a link $e = uv$ can be written as a sum of two parts $\mu(e) = \mu_v(e) + \mu_u(e)$ where: $\mu_v(e)$ is the sum of the dual variables associated with v that contribute to $\mu(e)$, and $\mu_u(e)$ is the sum of the dual variables associated with u that contribute to $\mu(e)$.

▶ **Lemma 14.** *At the end of step 3 of the algorithm, and then at the end of every iteration in the "while" loop, the following holds.*

(i) *If a link e has exactly one endnode in a node v of T/I, then $\mu_v(e) \leq 1$, and $\mu_v(e) \leq \rho - 1$ unless the original endnode of e contained in v is an original unmatched leaf. If e has both endnodes in a compound node v then $\mu(e) \leq \rho$.*

(ii) *If $\rho \geq 1.75$ then $\pi(v) \geq 1$ for any $v \in C \cup U$.*

Proof. It is easy to see that the statement holds at the end of step 3 of the algorithm. We will prove by induction on the number of contraction steps that the statement continues to hold during the algorithm. For that, let us consider various operations performed by the algorithm.

Let us consider the greedy contraction operation. Then (i) continues to hold since greedy contractions do not change the dual variables. Suppose that a link uv was contracted into a compound node c, where u, v are leaves of T/I. By the induction hypothesis, $\pi(u), \pi(v) \geq 1$. Thus $\pi(c) \geq \pi(u) + \pi(v) - 1 \geq 1$, and hence (ii) continues to hold as well.

The other operation is contracting a semi-closed tree T' with I' into a new compound node c. The easy case is when $|C'| \geq 1$ or $|M'| \geq 2$. Then (i) continues to hold since in this case we do not change the dual variables. Also (ii) holds for c, since in this case

$$\pi(c) \geq (\pi(C') + 2(\rho - 1)|M'| + |U'|) - (|M'| + |U'|) \geq \pi(C') + |M'|/2 \geq 1 \ .$$

Now we consider the more complicated cases 1 and 2 in the "while" loop.

Case 1: $|C'| = 0$ and either $|M'| = 0$ or $|M'| = 1, |U'| \geq 2$. Recall that in this case we zero the dual variables of the endnodes of the link in M', if any, raise the dual variables of the unmatched leaves by $\rho - 1$, and raise the dual variable $y_{T'}$ of T' from 0 to $\rho - 1$. Let $e = uv$ be a link and consider two cases.

If e has exactly one endnode in T', say v, then v is not an unmatched leaf of T', since T' is semi-closed. Thus since $C' = \emptyset$, $\mu_v(e) = 0$ after changing the dual variables, and $\mu_c(e) = \rho - 1$ after we contract T' into c. Hence (i) holds for e.

Suppose that e has both endnodes in T'. Then the dual load of e can only decrease, unless one endnode of e, say v, is an unmatched leaf of T'. Note that then the other endnode of e is not an unmatched leaf, since T' has no greedy contraction. Before we change the dual variables, we have $\mu_v(e) \leq 1$, by the induction hypothesis. After we change the dual variables, $\mu_u(e) = 0$ (since T' is semi-closed, and since we zero the dual variables of the endnodes of the link in M'). Hence at the end of the operation we have $\mu(e) \leq \rho$, and e enters the new compound node c, so (i) continues to hold.

Now we show that (ii) holds for the new compound node c. Note that

$$\begin{aligned}
\pi(c) &\geq (y_{T'} + \pi(U') + (\rho-1)|U'|) - (|M'| + |U'|) \\
&\geq (\rho - 1) + |U'| + (\rho - 1)|U'| - |M'| - |U'| \\
&= (\rho - 1)(|U'| + 1) - |M'|
\end{aligned}$$

If $|M'| = 0$ then we get $\pi(c) \geq 2(\rho - 1) = 1.5$. If $|M'| = 1$ and $|U'| \geq 2$ then we get $\pi(c) \geq 3(\rho - 1) - 1 = 1.25$. In both cases, (ii) continues to hold for c.

Case 2: $|C'| = 0$ and $|M'| = |U'| = 1$ (T' is locking non-dangerous). In this case we only raise the dual variable $q_{T'}$ to $\rho - 1$, and it is easy to verify that (i) continues to hold in this case. To see that (ii) continues to hold for the new compound node c note that

$$\begin{aligned}
\pi(c) &\geq (q_{T'} + \pi(U') + 2(\rho-1)|M'|) - (|M'| + |U'|) \\
&\geq (\rho - 1) + 1 + 2(\rho - 1) - 2 \\
&= 3(\rho - 1) - 1 = 1.25
\end{aligned}$$

This concludes the proof of the lemma. ◀

The above lemma implies that at the end of the algorithm, the dual solution (y, z, q) violates the dual constraints by a factor of ρ, and thus $(y, z, q)/\rho$ is a feasible solution to the dual program. Hence by the Weak Duality Theorem, $y(\mathcal{E}) + q(\mathcal{T}) \leq \rho\tau$, where τ is the optimal LP value. If $\rho \geq 1.75$, then the unique compound node (that contains r) has dual credit at least 1, and thus our dual solution fully pays for the links added, namely, $y(\mathcal{E}) + q(\mathcal{T}) \geq |I|$. Consequently, for $\rho = 1.75$ we get $|I| \leq y(\mathcal{E}) + q(\mathcal{T}) \leq \rho\tau$, as required.

5 Primal-fitting analysis of Algorithm 2 (Theorem 12)

5.1 Reduction to the minimum weight leaf edge-cover problem

Let Π be the polyhedron defined by the constraints of (LP1), namely:

$$\begin{array}{rcllr} x_e & \geq & 0 & \forall e \in E & (1) \\ x(\delta(T')) & \geq & 1 & \forall T' \in \mathcal{T} & (2) \\ x(\sigma(s_e)) - x_e & \geq & 0 & \forall e \in W & (3) \\ x(\delta(v)) & = & 1 & \forall v \in L & (5) \\ x(\delta(A, V)) & \geq & \lceil |A \cap L|/2 \rceil & \forall A \in \mathcal{O}_L & (6) \end{array}$$

Let $\tau = \min\{x(E) : x \in \Pi\}$ be the optimal value of (LP2). Let $R = V \setminus (L \cup S)$. Let $\rho \geq 1.5$ be a paramter set later to $\rho = 7/4$. Recall the weight function w on $E(L, V)$ defined at step 2 of Algorithm 2:

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

▶ **Lemma 15.** *Let F_L be a minimum w-weight exact edge-cover of L and $x \in \Pi$ such that $x(E) = \tau$. Then:*

$$\rho\tau \geq w(F_L) + \frac{1}{2}\sum_{v \in R} x(\delta(v)) \ . \tag{7}$$

Proof. Let Π_L be the polyhedron defined by the constraints (1), (5), and (6). Then Π_L is the convex hull of the exact edge-covers of L, see [20, Theorem 34.2]. 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 (1), (5), and (6). Since F_L is an optimal (integral) exact cover of L with respect to the weights w_e and $x' \in \Pi_L$, we have:

$$x' \cdot w \geq w(F_L) \ .$$

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 $\frac{1}{2}x_e$ token for each $e \in \delta(v)$.
(iv) Every $e \in W$ gets additional $\frac{1}{2}x_e$ token, 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), (ii), and (iii) hold. In the second step, every $e \in W$ gets $\frac{1}{2}x(\sigma(s_e))$ tokens moved at the first step to its stem s_e by the links in $\sigma(s_e)$. The amount of such tokens is at least $\frac{1}{2}x_e$, by (3). This gives an assignment of tokens as claimed. ◀

To prove Theorem 12 we prove the following.

▶ **Theorem 16.** *For $\rho = 7/4$, Algorithm 2 computes a solution I of size at most the right-hand size of (7). Thus $|I| \leq \rho\tau = \frac{7}{4}\tau$.*

5.2 Analysis of the algorithm (Proof of Theorem 16)

Let $M=\delta_{F_L}(L,L)$ be the set of leaf-to-leaf links in F_L and U the set of leaves unmatched by M. Then for $\rho=7/4$ we have:

$$w(F_L)=\rho|M\setminus W|+\left(\rho-\frac{1}{2}\right)|U|+\left(\rho+\frac{1}{2}\right)|M\cap W|=\frac{7}{4}|M\setminus W|+\frac{5}{4}|U|+\frac{9}{4}|M\cap W| \ .$$

Thus (7) implies:

$$\rho\tau\geq\frac{7}{4}|M\setminus W|+\frac{5}{4}|U|+\frac{9}{4}|M\cap W|+\frac{1}{2}\sum_{v\in R}x(\delta(v)) \ . \tag{8}$$

For the anlysis, we will assign tokens to nodes and edges of T according to the r.h.s. of (8), plus 1 extra token to (the compound node) r. Each time a contraction is performed (lines 3,4,7,8 in Algorithm 2), we assign 1 token to the compound node that results from the contraction. For example, every link $e\in M\cap W$ own $9/4$ tokens, and when it is added to the partial solution I at step 3 of Algorithm 2, these $9/4$ tokens pay both for the link addition and for the token assiged to the resulting compound node of T/I (and a spare of $1/4$ token remains). After all links in $M\cap W$ are moved from M to I, we maintain the following invariant for the tree T/I and for links in M and nodes in R that are not yet contracted into compound nodes.

Tokens Invariant

(i) Every $e\in M\setminus W$ owns $\rho=\frac{7}{4}$ tokens.
(ii) Every non-compound leaf unmatched by M owns $\rho-\frac{1}{2}=\frac{5}{4}$ tokens.
(iii) Every compound node owns 1 token.
(iv) Every $v\in R$ owns $\frac{1}{2}x(\delta(v))$ tokens.

For a subtree T' of T/I let us use the following notation:

- M' is the set of (not yet contracted) links in M with both endnodes in T'.
- U' is the set of leaves of T' unmatched by M.
- U'_0 is the set of original (non-compound) leaves of T' unmatched by M.
- C' is the set of non-leaf compound nodes of T' (this includes r, if $r\in T'$).
- R' is the set of (not yet contracted) nodes in R that belong to T'.
- $\Sigma'=\sum_{v\in R'}x(\delta(v))$

Let $tokens(T')$ denote the amount of tokens in T'; this includes the tokens on nodes of T' and tokens of links in M with both endnodes in T', namely:

$$\begin{aligned} tokens(T') &= \frac{7}{4}|M'|+(|C'|+|U'|-|U'_0|)+\frac{5}{4}|U'_0|+\frac{1}{2}\Sigma' \\ &= \frac{7}{4}|M'|+|U'|+\frac{1}{4}|U'_0|+\frac{1}{2}\Sigma'+|C'| \end{aligned}$$

If we require not to overspend the credit provided by (8), then each time we contract T' with I' we need the following property.

▶ **Definition 17.** A contraction of T' with I' is *legal* if $tokens(T')\geq|I'|+1$.

This means that the set I' of the links added to I and the 1 token assigned to the new compound node are paid by the total amount of tokens in T'. We do only legal contractions, which implies that at any step of the algorithm

$$|I|+tokens(T/I)\leq tokens(T) \ .$$

Thus at the last iteration, when T/I becomes a single compound node, $|I|$ is at most the right-hand side of (8).

Recall that after step 3, we have only two types of contractions of T' with I': a greedy contraction of a path by a single link between two unmatched leaves, and a contraction of a semi-closed tree with a link set of size $|I'| = |M'| + |U'|$. In the case of a greedy contraction, $tokens(T') \geq |U'| = 2$ while $|I'| = 1$; thus this contraction is legal. For a semi-closed subtree T' of T/I, we prove the following.

▶ **Lemma 18.** *Suppose that the Partial Solution Invariant and the Tokens Invariant hold for T, M, and I, and that T/I has no greedy contraction. Then $tokens(T') \geq |M'| + |U'| + 1$ holds for any non-dangerous semi-closed subtree T' of T/I.*

5.3 Proof of Lemma 18

Let T' be a semi-closed subtree of T/I w.r.t. M with root r' and node set V'. Assume that $tokens(T') - (|M'| + |U'|) < 1$. We will show that T' is dangerous. Note that by the Tokens Invariant:

$$tokens(T') - (|M'| + |U'|) = \frac{3}{4}|M'| + \frac{1}{4}|U'_0| + \frac{1}{2}\Sigma' + |C'| = \frac{1}{4}(3|M'| + |U'_0| + 2\Sigma') + |C'|$$

Since we assume that $tokens(T') - (|M'| + |U'|) < 1$, this immediately implies:

▶ **Lemma 19.** $|C'| = 0$ *and* $3|M'| + |U'_0| + 2\Sigma' < 4$*; thus* $|M'| \leq 1$*, and if* $|M'| = 1$ *then* $|U'_0| = 0$ *and* $\Sigma' < 1/2$*.*

Let us use the following additional notation:

- L' is the set of leaves of T'.
- S' is the set of (the original) stems of T'.

▶ **Lemma 20.** $|S'| = 0$*.*

Proof. Note that the Partial Solution Invariant implies that every stem s in T/I has exactly two leaf descendant, and they are both original leaves. Let a, b be the two leaf descendants of s, so a, b are original leaves and ab is a twin link. Since $ab \in W$, $ab \notin M'$. From the assumption that that T/I has no link greedy contraction we get that one of a, b is matched by M, as otherwise ab gives a greedy contraction. Moreover, $|M' \cap W| = 0$ and $|M'| \leq 1$ implies that $|M'| = 1$ and exactly one of a, b is matched by M. Consequently, $|M'| = |U'_0| = 1$, contradicting Lemma 19. ◀

▶ **Lemma 21.** $\Sigma' \geq |U'| + 1 - 2|M'|$*.*

Proof. Note that no link has both endnodes in U' (since T/I has no greedy contraction), and that $\delta(U') \cap \delta(T') = \emptyset$ (since T' is U'-closed). Thus

$$x(\delta(U') \cup \delta(T')) = \sum_{v \in U'} x(\delta(v)) + x(\delta(T')) \geq |U'| + 1 \ .$$

Let $e \in \delta(U')$. 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 exactly $2|M'|$. Hence $\Sigma' \geq |U'| + 1 - 2|M'|$, as claimed. ◀

▶ **Lemma 22.** *If* $|M'| = 1$ *then* $|U'| = 1$*.*

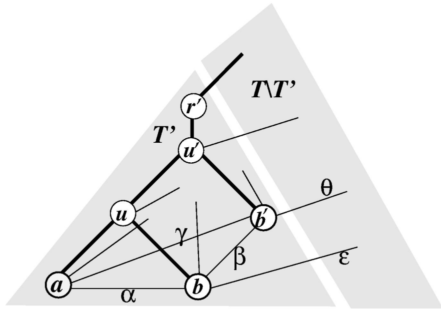

Figure 2 Illustration to the proof of Lemma 24.

Proof. If $|U'| \geq 2$ then Lemma 21 gives the contradiction $\Sigma' \geq 1$. Suppose that $|U'| = 0$. Then $|L'| = 2$, say $L' = \{b, b'\}$, and so $M' = \{bb'\}$, since $|M'| = 1$. Consequently, the contraction of bb' creates a new leaf. We obtain a contradiction by showing that then the path between b and b' in T/I has an internal compound node. By the Partial Solution Invariant b, b' are original leaves. Note that in the original tree T the contraction of bb' does not create a new leaf, since $bb' \notin W$. This implies that in T, there is a subtree $\hat{T}$ of T hanging out of a node z on the path between b and b' in T. This subtree $\hat{T}$ is not present in T/I, hence it was contracted into a compound node during the construction of our partial solution I. Thus T/I has a compound node $\hat{z}$ that contains $\hat{T}$, and since $\hat{z}$ contains a node z that belongs to the path between b and b' in T, the compound node of T/I that contains z belongs to the path between b and b'in T/I. ◀

▶ **Corollary 23.** $|C'| = |S'| = |U_0'| = 0$, $|M'| = |U'| = 1$ *(thus T' has 3 leaves), and* $\Sigma' < 1/2$.

Proof. We have $|C'| = 0$ and $|M'| \leq 1$ by Lemma 19 and $|S'| = 0$ by Lemma 20. If $|M'| = 0$ then from Lemma 21 we get that $\Sigma' \geq 2$, contradicting Lemma 19. Thus $|M'| = 1$ and by Lemmas 19 and 22 we have $\Sigma' < 1/2$, $|U'| = 1$, and $|U_0'| = 0$. ◀

We now use the properties of T' summarized in Corollary 23 to show that T' must be dangerous. Let bb' be the matched pair and a the unmatched (compound) 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. 2, and note that $u = u'$ or/and $u' = r'$ may hold). Let $x_{ab} = \alpha$, $x_{bb'} = \beta$, $x_{ab'} = \gamma$, $x(\delta(b, T \setminus T') = \epsilon$, and $x(\delta(b', T \setminus T') = \theta$.

▶ **Lemma 24.** *$\alpha, \theta > 0$ or $\gamma, \epsilon > 0$; if $u \neq u'$ then $\gamma, \epsilon > 0$.*

Proof. Consider the contribution to Σ' of links in cuts $\delta(a)$ and $\delta(T_{r'})$:

(i) Cut $\delta(a)$: $\frac{1}{2} > \Sigma' \geq x(\delta(a)) - (\alpha + \gamma) \geq 1 - (\alpha + \gamma)$; hence $\alpha + \gamma > \frac{1}{2}$.

(ii) Cut $\delta(T_{r'})$: $\frac{1}{2} > \Sigma' \geq x(\delta(T_{r'})) - (\theta + \epsilon) \geq 1 - (\theta + \epsilon)$; hence $\theta + \epsilon > \frac{1}{2}$.

In particular, we cannot have $\alpha, \gamma = 0$ or $\theta, \epsilon = 0$. We show that each one of the cases $\alpha, \epsilon = 0$ or $\gamma, \theta = 0$ is also not possible.

If $\alpha, \epsilon = 0$ then $\gamma, \theta > \frac{1}{2}$, giving the contradiction $1 = x(\delta(b')) \geq \gamma + \theta > 1$.

If $\gamma, \theta = 0$ then $\alpha, \epsilon > \frac{1}{2}$, giving the contradiction $1 = x(\delta(b)) \geq \alpha + \epsilon > 1$.

Now let us consider the case $u \neq u'$. Then by considering the cut $\delta(T_u)$ we get: $1/2 > \Sigma' \geq x(\delta(T_u)) - (\beta + \gamma + \epsilon) \geq 1 - (\beta + \gamma + \epsilon)$; hence $\beta + \gamma + \epsilon > 1/2$.

If $\gamma = 0$ then $\beta + \epsilon > 1/2$, and $\alpha > 1/2$ by (i); by considering the cut $\delta(b)$ we get the contradiction $1 = x(\delta(b)) \geq \alpha + \beta + \epsilon > 1/2 + 1/2 = 1$.

If $\epsilon = 0$ then $\beta + \gamma > 1/2$, and $\theta > 1/2$ by (ii); by considering the cut $\delta(b')$ we get the contradiction $1 = x(\delta(b')) \geq \beta + \gamma + \theta > 1/2 + 1/2 = 1$. ◀

Lemma 24 implies that T' is dangerous. 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.

This concludes the proof of Lemma 18.

References

1 J. Cheriyan and Z. Gao. Private communication. 2014.

2 J. Cheriyan and Z. Gao. Approximating (unweighted) tree augmentation via lift-and-project, part II. Manuscript, 2015.

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

4 J. Cheriyan, H. Karloff, R. Khandekar, and J. Koenemann. On the integrality ratio for tree augmentation. *Operation Research Letters*, 36(4):399–401, 2008.

5 Y. Chu and T. Liu. On the shortest arborescence of a directed graph. *Science Sinica*, 14:1396–1400, 1965.

6 N. Cohen and Z. Nutov. A $(1 + \ln 2)$-approximation algorithm for minimum-cost 2-edge-connectivity augmentation of trees with constant radius. *Theoretical Computer Science*, 489-490:67–74, 2013.

7 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.

8 G. Even, J. Feldman, G. Kortsarz, and Z. Nutov. A 1.8-approximation algorithm for augmenting edge-connectivity of a graph from 1 to 2. *ACM Transactions on Algorithms*, 5(2), 2009.

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

10 G. N. Frederickson and J. Jájá. On the relationship between the biconnectivity augmentation and traveling salesman problem. *Theoretical Computer Science*, 19(2):189–201, 1982.

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

12 M. Goemans and D. Williamson. A general approximation technique for constrained forest problems. *SIAM J. Computing*, 24(2):296–317, 1995.

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

14 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.

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

16 G. Kortsarz and Z. Nutov. A simplified 1.5-approximation algorithm for augmenting edge-connectivity of a graph from 1 to 2. To appear in *Transactions on Algorithms*, 2014.

17 L. C. Lau, R. Ravi, and M. Singh. *Iterative Methods in Combinatorial Optimization*. Cambridge University Press, 2011.

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

19 H. Nagamochi. An approximation for finding a smallest 2-edge connected subgraph containing a specified spanning tree. *Discrete Applied Math.*, 126:83–113, 2003.

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

21 A. Sebo and J. Vygen. Shorter tours by nicer ears: 7/5-approximation for the graph-TSP, 3/2 for the path version, and 4/3 for two-edge-connected subgraphs. *Combinatorica*, 34(5):597–629, 2014.