Operations Research Letters 50 (2022) 667–673

Contents lists available at ScienceDirect

Operations Research Letters

journal homepage: www.elsevier.com/locate/orl

On small-depth tree augmentations

Ojas Parekh [a], R. Ravi [b], Michael Zlatin [b,*]

[a] Center for Computing Research, Sandia National Laboratories, Albuquerque, NM, USA
[b] Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA, USA

ARTICLE INFO

Article history:
Received 27 July 2021
Received in revised form 5 October 2022
Accepted 7 October 2022
Available online 14 October 2022

Keywords:
Approximation algorithm
Network design
Integrality gap

ABSTRACT

We study the Weighted Tree Augmentation Problem for general link costs. We show that the integrality gap of the odd-LP relaxation for the (weighted) Tree Augmentation Problem for a k-level tree instance is at most $2 - \frac{1}{2^{k-1}}$. For 2- and 3-level trees, these ratios are $\frac{3}{2}$ and $\frac{7}{4}$ respectively. Our proofs are constructive and yield polynomial-time approximation algorithms with matching guarantees.

1. Introduction

We consider the *weighted tree augmentation problem (TAP)*: Given an undirected graph $G = (V, E)$ with non-negative weights c on the edges, and a spanning tree T, find a minimum cost subset of edges $A \subseteq E(G) \setminus E(T)$ such that $(V, E(T) \cup A)$ is two-edge-connected. We will call the elements of $E(T)$ as (tree) edges and those of $E(G) \setminus E(T)$ as *links* for convenience. A graph is *two-edge-connected* if the removal of any edge does not disconnect the graph, i.e., it does not have any cut edges. Since cut edges are also sometimes called bridges, this problem has also been called *bridge connectivity augmentation* in prior work [10].

While TAP is well studied in both the weighted and unweighted case [10,14,17,8,5,16,1,9,12], it is NP-hard even when the tree has diameter 4 [10] or when the set of available links form a single cycle on the leaves of the tree T [6], and is also APX-hard [15]. Weighted TAP was one of the simplest network design problems without a better than 2-approximation in the case of general (unbounded) link costs and arbitrary depth trees, until very recently [18,19]. For the case of n-node trees with height k, Cohen and Nutov [8] gave a $(1 + \ln 2) \simeq 1.69$-approximation algorithm that runs in time $n^{3^k} \cdot poly(n)$ using an idea of Zelikovsy for approximating Steiner trees. Very recently, this approach has been extended to provide an approximation to the general case of the problem with the same performance guarantee by Traub and Zenklusen [18]. A follow-up paper by the same authors [19] improved the approximation ratio to nearly 1.5. However, these papers do not provide any new results on the integrality gap of some natural LP relaxations for the problem that we discuss next.

1.1. EDGE-LP relaxation

TAP can also be viewed as a set covering problem. The edges of the tree T define a laminar collection of cuts that are the elements to be covered using sets represented by the links. A link ℓ is said to *cover* an edge e if the unique cycle of $\ell + T$ contains e. Here we use $\mathrm{cov}(e)$ for a tree edge e to denote the set of links which cover e. The natural covering linear programming relaxation for the problem, EDGE-LP, is a special instance of a set covering problem with one requirement (element) corresponding to each cut edge in the tree. Since the tree edges define subtrees under them (after rooting it at an arbitrary node) that form a laminar family, this is also equivalent to a laminar cover problem [6].

$$\min \sum_{\ell \in E} c_\ell x_\ell$$

* Corresponding author.
E-mail addresses: odparek@sandia.gov (O. Parekh), ravi@cmu.edu (R. Ravi), mzlatin@andrew.cmu.edu (M. Zlatin).

https://doi.org/10.1016/j.orl.2022.10.002
0167-6377/

$$x(\text{cov}(e)) \geq 1 \qquad \forall e \in E(T) \tag{1}$$

$$x_\ell \geq 0 \qquad \forall \ell \in E \tag{2}$$

Fredrickson and Jájá showed that the integrality gap for EDGE-LP can not exceed 2 [10] and also studied the related problem of augmenting the tree to be two-node-connected (biconnectivity versus bridge-connectivity augmentation) [11]. Cheriyan, Jordán, and Ravi, who studied half-integral solutions to EDGE-LP and proved an integrality gap of $\frac{4}{3}$ for such solutions, also conjectured that the overall integrality gap of EDGE-LP was at most $\frac{4}{3}$ [6]. However, Cheriyan et al. [7] later demonstrated an instance for which the integrality gap of EDGE-LP is at least $\frac{3}{2}$.

1.2. ODD-LP relaxation

Fiorini et al. studied the relaxation consisting of all $\{0, \frac{1}{2}\}$-Chvátal-Gomory cuts of the EDGE-LP [9]. We call their extended linear program the ODD-LP.

We define $\delta(S)$ for $S \subset V$ as the set of all links and edges with exactly one endpoint in S, and recall that $\text{cov}(e)$ for a tree edge e is the set of links that cover e. We use $E(T)$ to refer to the set of tree edges, and L is the set of links, $E(G) \setminus E(T)$.

$$\min \sum_{\ell \in E} c_\ell x_\ell$$

$$x(\delta(S) \cap L) + \sum_{e \in \delta(S) \cap E(T)} x(\text{cov}(e)) \geq |\delta(S) \cap E(T)| + 1 \qquad \forall S \subseteq V, |\delta(S) \cap E(T)| \text{ is odd} \tag{3}$$

$$x_\ell \geq 0 \qquad \forall \ell \in E$$

We describe here the validity of the constraints in ODD-LP using a proof due to Robert Carr. Consider a set of vertices S such that $|\delta(S) \cap E(T)|$ is odd. By adding together the edge constraints for $\delta(S) \cap E(T)$ we get:

$$\sum_{e \in \delta(S) \cap E(T)} x(\text{cov}(e)) \geq |\delta(S) \cap E(T)|$$

Now we can add any non-negative terms to the left hand side and still remain feasible. Therefore

$$x(\delta(S) \cap L) + \sum_{e \in \delta(S) \cap E(T)} x(\text{cov}(e)) \geq |\delta(S) \cap E(T)|$$

is also feasible. Now consider any link ℓ. If x_ℓ appears an even number of times in $\sum_{e \in \delta(S) \cap E(T)} x(\text{cov}(e))$ then ℓ is not in $\delta(S)$. Similarly, if x_ℓ appears an odd number of times in $\sum_{e \in \delta(S) \cap E(T)} x(\text{cov}(e))$ then ℓ is in $\delta(S)$. So, the coefficient of every x_ℓ on the left hand side of this expression is even. In particular, for any integer solution the left hand side is even and the right hand side is odd. Therefore, we can strengthen the right hand side by increasing it by one, and the resulting constraint will still be feasible for any integer solution. The constraint,

$$x(\delta(S) \cap L) + \sum_{e \in \delta(S) \cap E(T)} x(\text{cov}(e)) \geq |\delta(S) \cap E(T)| + 1$$

is thus valid for any integer solution to TAP as desired.

We prove a lemma about how a transformation of a feasible solution to the ODD-LP that we call a link subdivision continues to preserve feasibility, that we use later in our integrality gap proofs.

Lemma 1.1. *Let (T, L) be a TAP instance and let $\bar{x}$ be a feasible solution to ODD-LP(T, L). Fix some link $\ell = (a, b) \in L$, and let $(a, v_1, \ldots, v_k, b)$ be an ordered sequence of distinct nodes on the path from a to b in T. Let $\Lambda = \{(a, v_1), (v_1, v_2), \ldots (v_{k-1}, v_k), (v_k, b))\}$ and let $L' = (L \setminus (a, b)) \cup \Lambda$ (call this a link subdivision). Then the solution $\bar{y}$ obtained from $\bar{x}$ by setting $\bar{y}_\ell = \bar{x}_{(a,b)}$ for all $\ell \in \Lambda$ is feasible for ODD-LP(T, L').*

Proof. Fix a set of vertices $S \subseteq V$ such that $|\delta(S) \cap E(T)|$ is odd. Since $\bar{x}$ is feasible, we have

$$\bar{x}(\delta(S) \cap L) + \sum_{e \in \delta(S) \cap E(T)} \bar{x}(\text{cov}(e)) \geq |\delta(S) \cap E(T)| + 1.$$

Since the links in Λ cover exactly the set of edges along the path between a and b in T, we have

$$\sum_{e \in \delta(S) \cap E(T)} \bar{x}(\text{cov}(e)) = \sum_{e \in \delta(S) \cap E(T)} \bar{y}(\text{cov}(e)).$$

We now consider two cases. If $a, b \in S$, then $(a, b) \notin \delta(S) \cap L$, and so clearly $\bar{y}(\delta(S) \cap L) \geq \bar{x}(\delta(S) \cap L))$. Otherwise, suppose without loss of generality that $a \in S$ and $b \notin S$. Then the path between a and b in T contains an odd number of edges in $\delta(S) \cap E(T)$. Each link in Λ which is not $\delta(S) \cap L$ covers an even number of edges in $\delta(S) \cap E(T)$. Therefore, there must be some link ℓ in Λ contained in $\delta(S) \cap L$. This link has value $\bar{y}_\ell = \bar{x}_{(a,b)}$, hence we have

$$\bar{y}(\delta(S) \cap L) \geq \bar{x}(\delta(S) \cap L),$$

and the claim follows. $\square$

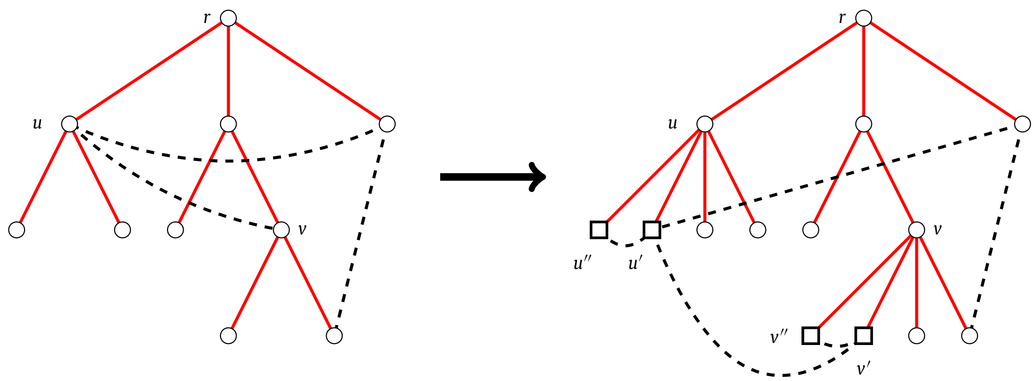

Fig. 1. Transformation to a leaf to leaf instance.

2. Preliminaries

We will use the following theorem about the ODD-LP [9]. For a choice of a root r, we call links which connect two different components of $T - r$ as cross-links, and those that go from a node of T to its ancestor as up-links.

Theorem 2.1 *([9], Theorem 1.1). The ODD-LP is integral for weighted TAP instances that contain only cross- and up-links.*

The integrality of the formulation is shown by demonstrating that the constraint matrix is an example of a binet matrix [2,3], a generalization of network matrices that are a well-known class of totally unimodular matrices. Moreover, while general Chvátal-Gomory closures are NP-hard to optimize over, these restricted versions over half-integral combinations can be optimized in polynomial time [4]. Such instances with only cross- and up-links are informally called "star-shaped" with the center of the star being the chosen root, so we will refer to the above result as saying that the ODD-LP for star-shaped instances centered at a root have integrality gap 1 and solutions to such instances can be obtained in polynomial time.

Without loss of generality, we may consider TAP instances where all links go between two leaves [13]. We reproduce the proof here for completeness.

Lemma 2.2. *Given an instance (T, L, c) of weighted TAP, there is a corresponding, polynomial-sized instance (T', L', c') with all links having both endpoints as leaves, such that there is a cost-preserving bijection between the solutions to the two instances.*

Proof. The proof proceeds by a simple graph reduction. Suppose we are given an instance defined by a graph G with associated tree T for the weighted TAP. We create a new instance of the leaf-to-leaf version as follows: For every internal node u in the original tree T, we add two new leaf nodes u' and u'' both adjacent to u to get a new tree T'. For every link $f = (v, u)$ in the original instance, we reconnect the link to now end in the leaf u' rather than the internal node u in the tree T'. Thus, if both v and u are internal nodes, the new link is (v', u'); if only u is internal, the new link is (v, u') and if both are leaves, the new link is the same (u, v) as in G. Note that the new graph G' is a leaf-to-leaf instance. In addition, for every internal node u in the original tree T, we add a new link of zero cost between u' and u'' - this will serve to cover the newly added edges (u, u') and (u, u'') without changing the coverage of any of the edges in the original tree T. See Fig. 1. □

Remark 2.3. The cost-preserving bijection described above can be extended to map fractional solutions of odd-LP(T, L) to odd-LP(T', L'). In other words, every weighted TAP problem can be reduced to an instance where all links go between a pair of leaves without loss of generality for investigating approximation ratios for the problem and integrality gaps of the odd-LP.

Note that given a rooted tree of k levels (i.e., the maximum distance of any leaf from the root is k), the above transformation results in a leaf-to-leaf instance also with k levels.

3. Improved integrality gaps for trees of depth 2 and 3

Theorem 3.1. *The integrality gap of the ODD-LP for a two-level tree instance is at most $\frac{3}{2}$. A solution with this approximation ratio for these instances can be obtained in polynomial time.*

Proof. Given a two-level TAP instance on tree T and links L, we first show how to transform the instance into two instances that we can solve exactly. We then apply the same transformation to the components of any integral solution A to obtain feasible solutions to the two new instances, the better of which has value at most $\frac{3}{2} \cdot c(A)$. We will then use the same reductions on fractional feasible solutions to the ODD-LP, and use Lemma 1.1 to show that the resulting star-shaped instances are feasible for the ODD-LP and then use Theorem 2.1 to arrive at the stated integrality gap and resulting algorithm.

We say that the root r is at level 1 and its children $\{c_1, c_2, \ldots, c_d\}$ are internal nodes at level 2, where d is the number of non-leaf children of the root. First using Lemma 2.2, we assume that all links go between a pair of leaves. We partition L into $L_1 \dot\cup L_2$, where L_i is the set of links whose least common ancestor (henceforth lca) is a node at level i in the tree.

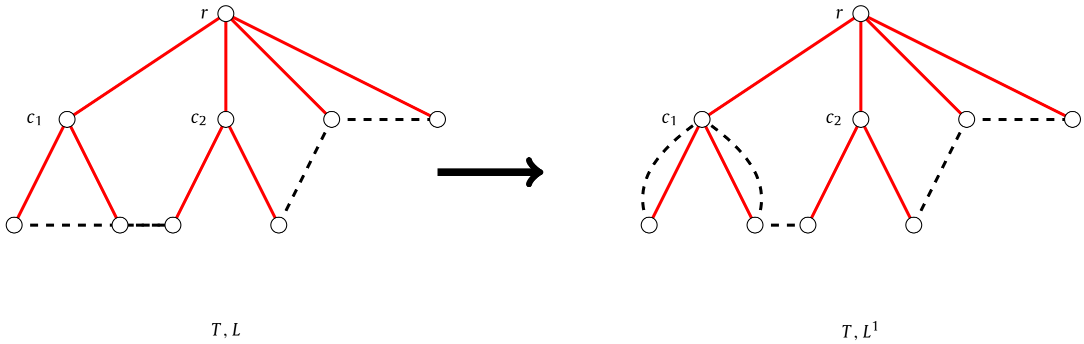

Fig. 2. Transformation to a star-shaped instance for the root.

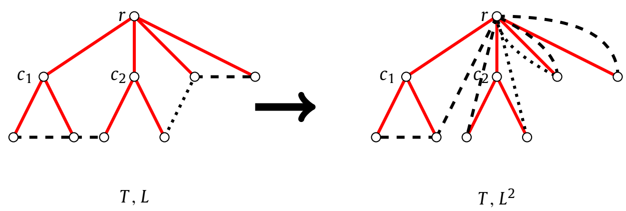

Fig. 3. Transformation to three star-shaped instances around the root and its two internal children.

We will create two new instances (T, L^1) and (T, L^2). The first instance (T, L^1) is obtained by replacing every link (u, v) in L_2 with lca c say, with two up-links (u, c) and (v, c) of the same cost. Notice that (T, L^1) is a star-shaped instance with root r. The second instance (T, L^2) is obtained by replacing every link (u, v) in L_1 with lca the root r, with two up-links (u, r) and (r, v) of the same cost. (T, L^2) can be decomposed into $d+1$ disjoint star shaped instances: d of these are defined by the star around each non-leaf child of the root, and the last is the star defined by the root and its leaf-children. Thus, (T, L^2) is a disjoint collection of star-shaped instances which can be solved by solving each of these $d+1$ induced star-shaped instances separately and taking the union of their solutions. See Figs. 2 and 3.

Given an optimal solution A to (T, L), we now construct two solutions A^1 and A^2, feasible to (T, L^1) and (T, L^2) respectively, by applying the same reduction as above to the corresponding components in A. Let $A = A_1 \dot\cup A_2$ where A_i the set of links whose lca is a node in level i of the tree. To create A^1, we replace each link (u, v) in A_2 with the two up-links (u, c) and (v, c) in L^1. Note that this set of links along with A_1 gives a feasible solution to (T, L^1) and has cost $c(A_1) + 2c(A_2)$. To create A^2, we replace each link (u, v) in A_1 with the two up-links (u, r) and (v, r) in L^2. Note that this set of links along with A_2 gives a feasible solution to (T, L^2) and has cost $2c(A_1) + c(A_2)$.

As described above, each of A^1 and A^2 can be transformed into a feasible solution to (T, L) with the same cost. Therefore, by taking the better of the two, there is a solution of cost at most $\min\{c(A_1) + 2c(A_2), 2c(A_1) + c(A_2)\} \le \frac{3}{2}c(A)$.

Note that the link replacements in both transformations are link subdivisions. Hence, by Lemma 1.1, we can apply the same transformation to any fractional solution feasible to ODD-LP(T, L) to obtain two fractional solutions feasible to ODD-LP(T, L^1) and ODD-LP(T, L^2) respectively. Since the resulting star shaped instances have integrality gap 1 by Theorem 2.1, the claims about the integrality gap and a polynomial-time approximation algorithm with this ratio also follow. □

Theorem 3.2. *The integrality gap of the odd-LP for a three-level tree instance is at most $\frac{7}{4}$. A solution with this approximation ratio for these instances can be obtained in polynomial time.*

Proof. As before, we first show how to transform the instance into new instances, and show that the same applied to any integral solution A gives a solution of value at most $\frac{7}{4} \cdot c(A)$. Again, the same reduction will also apply to fractional solutions that obey the ODD-LP constraints to prove the claim.

Using Lemma 2.2, we assume that all links go between a pair of leaves. We partition L into $L_1 \dot\cup L_2 \dot\cup L_3$, where L_i is the set of links whose lca is a node at level i in the tree. We say that the root r is at level 1 and its non-leaf children $\{c_1, c_2, \ldots, c_d\}$ are at level 2, and the children of these nodes that are internal nodes are in level 3 of the tree. We will create three new instances (T, L^1), (T, L^2) and (T, L^3) as follows.

- The first instance (T, L^1) is obtained by replacing every link (u, v) in $L_2 \cup L_3$ with lca c say, with two up-links (u, c) and (v, c) of the same cost. See Fig. 4. Notice that (T, L^1) is a star-shaped instance with root r.
- The second instance (T, L^2) is obtained by replacing every link (u, v) in $L_1 \cup L_3$ with lca c say, with two up-links (u, c) and (c, v) of the same cost. Notice, that (T, L^2) can be decomposed into $d+1$ different star shaped instances. One of these star-shaped instances is a one-level instance centered at the root and consisting of the root and all of its leaf children. Each of the remaining d instances corresponds to a non-leaf child of the root v_i, consisting of the whole subtree T_i rooted at v_i and its edge to the root (v_i, r). Notice that these instances are pairwise disjoint and are star-shaped with centers v_i. See Fig. 5.
- Finally, the third instance (see Fig. 6), is obtained as follows. For each link $(a, b) \in L_2$, with lca c say, we replace it with two up-links (a, c) and (b, c) of the same cost. The interesting transformation is for links in L_1, where we now make up to *three* copies. For every

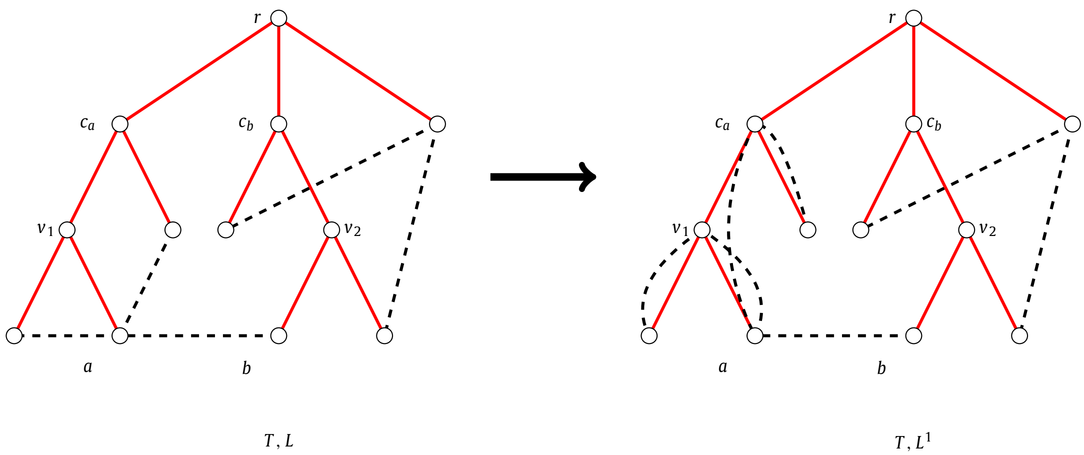

Fig. 4. Transformation to a star-shaped instance centered at the root.

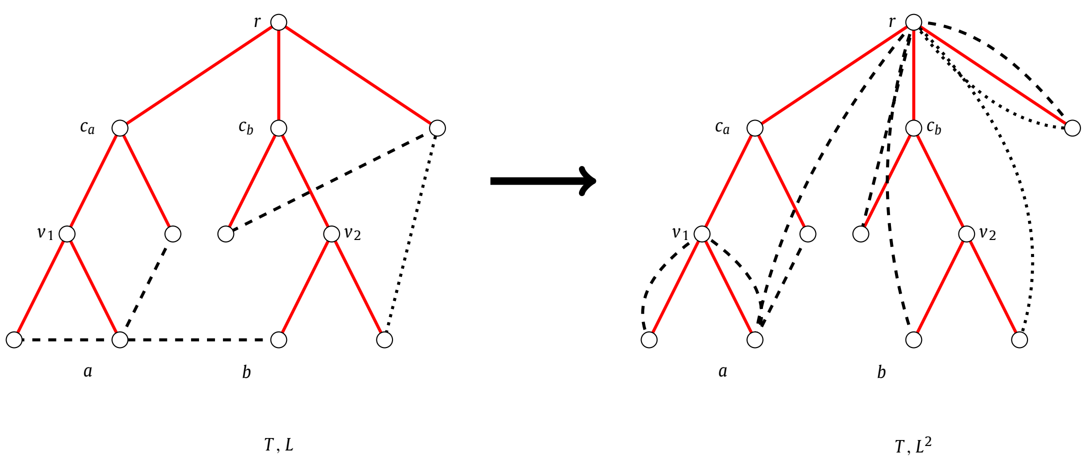

Fig. 5. Transformation to three star-shaped instances centered at the root and its two internal children.

link $(a, b) \in L_1$, let c_a and c_b denote the ancestor of a and b respectively in level 2. (if either a or b is in level 2 itself, then its ancestor in level 2 is itself). We now add three links (a, c_a), (c_a, c_b), (c_b, b) of the same cost as (a, b). Again, this instance can be solved exactly by decomposing it into pairwise disjoint star-shaped instances: one for each star around the internal nodes, say $v_1, \ldots, v_q$ in level 3, and one more tree around the root consisting of the set of all tree edges not in the stars around the v_i's.

Given an optimal solution A to (T, L), we now construct three solutions A^1, A^2 and A^3, feasible to (T, L^1), (T, L^2) and (T, L^3) respectively. Let $A = A_1 \dot\cup A_2 \dot\cup A_3$ where A_i the set of links whose lca is a node in level i of the tree. The solutions A^1, A^2 and A^3 are obtained from by transforming the links in A exactly as we transformed the corresponding type of links in L to obtain L^1, L^2 and L^3. Finally, each of A^1, A^2 and A^3 can be transformed into a feasible solution to (T, L) with the same cost. Therefore, by taking the best of the three, we can find a solution in polynomial time with cost at most $\min\{c(A_1)+2c(A_2)+2c(A_3), 2c(A_1)+c(A_2)+2c(A_3), 3c(A_1)+2c(A_2)+c(A_3)\} \leq \frac{7}{4}c(A)$.

Note that the link replacements in both transformations are link subdivisions. Hence, by Lemma 1.1, we can apply the same transformation to any fractional solution feasible to ODD-LP(T, L) to obtain three fractional solutions feasible to ODD-LP(T, L^1), ODD-LP(T, L^2), and ODD-LP(T, L^3) respectively. Since the resulting star shaped instances have integrality gap 1 by Theorem 2.1, the claims about the integrality gap and a polynomial-time approximation algorithm with this ratio also follow. □

4. Integrality gap for *k*-level trees

With the above cases, we can now calculate an upper bound on the value of the integrality gap for general k-level trees where the depth of any leaf from the root is k.

Theorem 4.1. *The integrality gap of the ODD-LP for a k-level tree instance is at most* $2 - \frac{1}{2^{k-1}}$. *A solution with this approximation ratio for these instances can be obtained in polynomial time for any fixed k.*

Proof. We will show how to transform a given k-level TAP instance (T, L) into k new instances, and show that the same applied to any integral solution A gives a solution of value at most $(2 - \frac{1}{2^{k-1}}) \cdot c(A)$. Again, the same reduction will also apply to fractional solutions that obey the ODD-LP constraints to prove the claim.

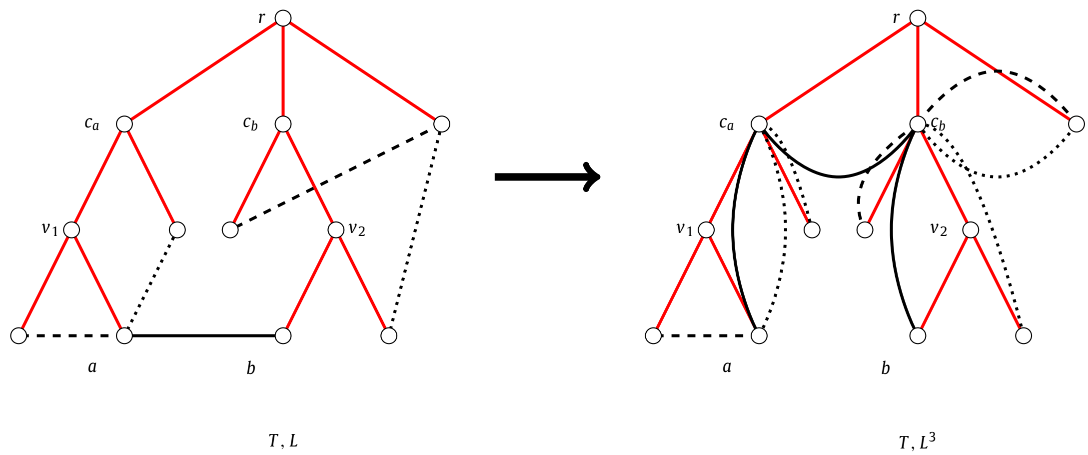

Fig. 6. Transformation to three star-shaped instances centered at the root and the stars around the two internal nodes in level 3.

Partition the links in L into subsets of links $L = L_1 \dot\cup L_2 \ldots \dot\cup L_k$ where L_ℓ is the subset whose lca is a node in level l of the tree for $\ell = 1, \ldots, k$. As before, we set up k new TAP instances $(T, L^1), \ldots, (T, L^k)$, where in (T, L^ℓ) we preserve the links in L_ℓ.

For $\ell = 1$, we replace each link (u, v) in $L_2, \ldots, L_k$ with two links (u, c) and (v, c), where c is the lca of u and v, yielding a star-shaped instance (T, L^1).

Suppose $1 < \ell \leq k$. For any link $(a, b) \in L_{\ell-1} \cup \bigcup_{p>\ell} L_p$, we replace it with the two links $(a, lca(a, b))$ and $(b, lca(a, b))$. For links $(a, b) \in L_q$ for $1 < q < \ell - 1$, let the ancestors of a and b in level $\ell - 1$ be u_a and u_b respectively, if they exist. We replace (a, b) with one of the following sets, with at most four links: $\{(a, u_a), (u_a, lca(u_a, u_b)), (lca(u_a, u_b), u_b), (u_b, b)\}$, or $\{(a, lca(a, u_b)), (lca(a, u_b), u_b), (u_b, b)\}$, or $\{(a, u_a), (u_a, lca(u_a, b)), (lca(u_a, b), b)\}$, or $\{(a, lca(a, b)), (lca(a, b), b)\}$, depending on which of u_a and u_b exist. Analogously, for $q = 1$, we instead use the following sets, with at most three links: $\{(a, u_a), (u_a, u_b), (u_b, b)\}$, $\{(a, u_b), (u_b, b)\}$, $\{(a, u_a), (u_a, b)\}$, $\{(a, b)\}$. Denote the obtained instance by (T, L^ℓ).

We now show that (T, L^ℓ) can be decomposed into pairwise disjoint star-shaped instances, thereby allowing us to solve these instances exactly. For every internal node v at level ℓ, we consider the instance on the subtree below it, along with the edge to its parent. These will be star-shaped instances with centers v. In addition, we consider a final instance, which will be a star-shaped instance around the root, whose tree edges are disjoint from the others.

We will show that each link is a cross-link or an up-link in one of the star-shaped instances. First consider the instances around the internal nodes v in level ℓ. Links in L_ℓ are already cross-links in these. Links in L_p for $p > \ell$ have been replaced with two links that become up links in these instances. Consider a link $(a, b) \in L_{l-1}$, such that v_a and v_b are the ancestors of a and b respectively that are in level l. We replaced this link with the two links $(a, lca(a, b))$ and $(b, lca(a, b))$. Now $lca(a, b)$ is a parent of v_a and v_b since $(a, b) \in L_{l-1}$ so these links form cross links for the star-shaped instances around v_a and v_b.

All the tree edges not in any of these star-shaped instances are considered in a final star-shaped instance rooted at r. For links $(a, b) \in L_q$ for $1 < q < \ell - 1$, let the ancestors of a and b in level $\ell - 1$ be u_a and u_b respectively, if they exist. For $q > 1$, all the links in these links become cross links for the star-shaped instances around the level ℓ internal nodes or up links for the instance rooted at r. The same holds for links in L_1, except that these sets also include cross links for the instance rooted at r.

Given an optimal solution A to (T, L), we can construct k solutions $A^1, A^2, \ldots, A^k$, feasible to $(T, L^1), (T, L^2), \ldots, (T, L^k)$ respectively. To get A^k, we simply apply the same transformations to the links of A which were applied to L to obtain L^k. Let $A = A_1 \dot\cup A_2, \cdots, \dot\cup A_k$, where A_i is the set of links in A with lca at level i, and let c_i denote the cost of the links in A_i. Each A^ℓ corresponds to a solution to (T, L) with at most the cost.

Based on the above construction, an upper bound on the cost of this set of candidate solutions is

$$C_1 = c_1 + 2c_2 + 2c_3 + 2c_4 + \ldots + 2c_k, \text{ if } \ell = 1$$

$$C_2 = 2c_1 + c_2 + 2c_3 + 2c_4 + \ldots + 2c_k, \text{ if } \ell = 2$$

$$C_3 = 3c_1 + 2c_2 + c_3 + 2c_4 + \ldots + 2c_k, \text{ if } \ell = 3$$

$$C_\ell = 3c_1 + 4c_2 + \ldots + 4c_{\ell-2} + 2c_{\ell-1} + c_\ell + 2c_{\ell+1} + \ldots + 2c_k, \ if \ \ 3 < \ell \leq k.$$

The best of these solutions has cost at most $\min\{C_1, \ldots, C_k\} \leq \sum_{i=1}^k \lambda_i C_i$ for any setting of weights such that $\lambda_i \geq 0$ for all i and $\sum_{i=1}^k \lambda_i = 1$.

Let $J(n) := \frac{2^n - (-1)^n}{3}$. Let $\lambda_i = \frac{1}{2^k - 1} J(k - i + 1)$ for $i \geq 2$. If k is even, let $\lambda_1 = \lambda_2$. If k is odd, let $\lambda_1 = \lambda_2 + 1$. With this choice of $\lambda = (\lambda_1, \ldots, \lambda_k)$, it is straightforward to verify that

$$\sum_{i=1}^k \lambda_i C_i = \left(\frac{2^k - 1}{2^{k-1}} \right) (c_1 + \ldots + c_2).$$

The key observation to verify this is that $J(n)$ satisfies the recurrence $J(n) = J(n-1) + 2J(n-2)$ for $n \geq 2$, which reflects the changes in the coefficients of $c_\ell, c_{\ell-1}$ and $c_{\ell-2}$ in going from $C_{\ell-1}$ to C_ℓ. With $J(1) = 1$ and $J(0) = 0$, this defines the Jacobsthal sequence, see (https://oeis.org/A001045).

Thus, we have $\min\{C_1, \ldots, C_k\} \le \left(2 - \frac{1}{2^{k-1}}\right)(c_1 + \ldots + c_k) = \left(2 - \frac{1}{2^{k-1}}\right) c(A)$.

As before, all link replacements correspond to link subdivisions. Thus, by Lemma 1.1, we can apply the same transformation to any fractional solution feasible to ODD-LP(T, L) to obtain k fractional solutions feasible to ODD-LP(T, L^{ℓ}) for each $\ell = 1, \ldots, k$. Since the resulting star shaped instances have integrality gap 1 by Theorem 2.1, the claims about the integrality gap and a polynomial-time approximation algorithm with this ratio also follow. □

While the above analysis shows integrality gaps of the ODD-LP converging to 2 as the depth of the tree grows, the main open question in our opinion is to show that the integrality gap of $\frac{3}{2}$ that we showed for 2-level trees is indeed the upper bound for all trees.

Data availability

No data was used for the research described in the article.

Acknowledgements

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525. OP was supported by the U.S. Department of Energy Office of Science, Office of Advanced Scientific Computing Research, Accelerated Research in Quantum Computing and Quantum Algorithms Teams programs. This material is based upon work supported by the U. S. Office of Naval Research under award number N00014-21-1-2243 to RR.

References

[1] David Adjiashvili, Beating approximation factor two for weighted tree augmentation with bounded costs, in: Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, 2017, pp. 2384–2399.
[2] Gautam Appa, Balázs Kotnyek, A bidirected generalization of network matrices, Networks 47 (4) (2006) 185–198.
[3] Gautam Appa, Balázs Kotnyek, Konstantinos Papalamprou, Leonidas Pitsoulis, Optimization with binet matrices, Oper. Res. Lett. 35 (3) (2007) 345–352.
[4] Alberto Caprara, Matteo Fischetti, {0, 1/2}-Chvátal-gomory cuts, Math. Program. 74 (3) (1996) 221–235.
[5] Joseph Cheriyan, Zhihan Gao, Approximating (unweighted) tree augmentation via lift-and-project, part I: stemless TAP, CoRR, arXiv:1508.07504 [abs], 2015.
[6] Joseph Cheriyan, Tibor Jordán, R. Ravi, On 2-coverings and 2-packings of laminar families, in: Algorithms-ESA'99, 1999, p. 72.
[7] Joseph Cheriyan, Howard Karloff, Rohit Khandekar, Jochen Könemann, On the integrality ratio for tree augmentation, Oper. Res. Lett. 36 (4) (2008) 399–401.
[8] Nachshon Cohen, Zeev Nutov, A (1+ ln2)-approximation algorithm for minimum-cost 2-edge-connectivity augmentation of trees with constant radius, Theor. Comput. Sci. 489 (2013) 67–74.
[9] Samuel Fiorini, Martin Groß, Jochen Könemann, Laura Sanità, Approximating weighted tree augmentation via chvátal-gomory cuts, in: Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, 2018, pp. 817–831.
[10] Greg N. Frederickson, Joseph Jájá, Approximation algorithms for several graph augmentation problems, SIAM J. Comput. 10 (2) (1981) 270–283.
[11] Greg N. Fredrickson, Joseph Jájá, On the relationship between the biconnectivity augmentation and traveling salesman problem, Theor. Comput. Sci. 19 (2) (1982) 189–201.
[12] Fabrizio Grandoni, Christos Kalaitzis, Rico Zenklusen, Improved approximation for tree augmentation: saving by rewiring, in: Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, ACM, 2018, pp. 632–645.
[13] Jennifer Iglesias, R. Ravi, Coloring down: 3/2-approximation for special cases of the weighted tree augmentation problem, 2017.
[14] Samir Khuller, Ramakrishna Thurimella, Approximation algorithms for graph augmentation, J. Algorithms 14 (2) (1993) 214–225.
[15] Guy Kortsarz, Robert Krauthgamer, James R. Lee, Hardness of approximation for vertex-connectivity network design problems, SIAM J. Comput. 33 (3) (2004) 704–720.
[16] Guy Kortsarz, Zeev Nutov, A simplified 3/2 ratio approximation algorithm for the tree augmentation problem, ACM Trans. Algorithms 12 (2) (2016) 23.
[17] R. Ravi, Steiner Trees and Beyond: Approximation Algorithms for Network Design, PhD thesis, Brown University, 1994.
[18] Vera Traub, Rico Zenklusen, A better-than-2 approximation for weighted tree augmentation, 2021.
[19] Vera Traub, Rico Zenklusen, Local search for weighted tree augmentation and Steiner tree, 2021.