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*

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.

Keywords: Approximation Algorithm, Network Design, Integrality Gap
PACS: 0000, 1111
2000 MSC: 0000, 1111

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 remains 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 $\text{cov}(e)$ 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].

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

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

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

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 adding 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.

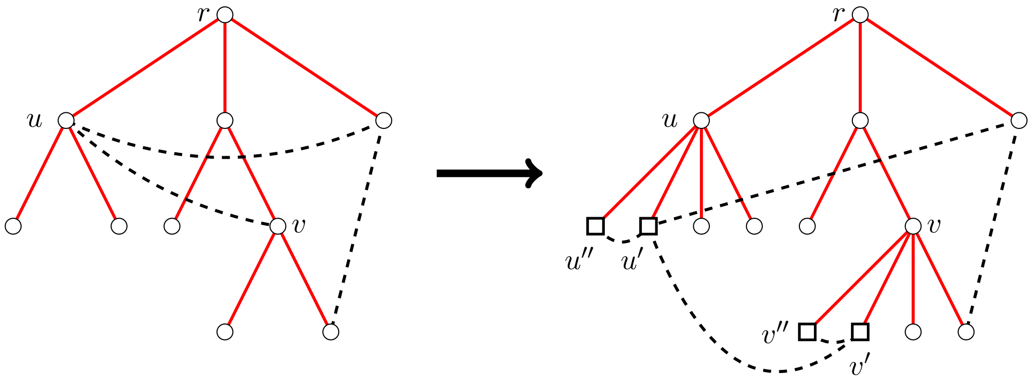

Figure 1: Transformation to a leaf to leaf instances

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. *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 Figure 1.

Given an solution $A \subseteq E(G) \setminus E(T)$ of minimum cost in the original instance on G, if we add the new zero cost edges for every internal node to A we get a solution of the same cost in the new instance. Conversely, the edges in any solution A' to the problem in G', when restricted to the original instance is a

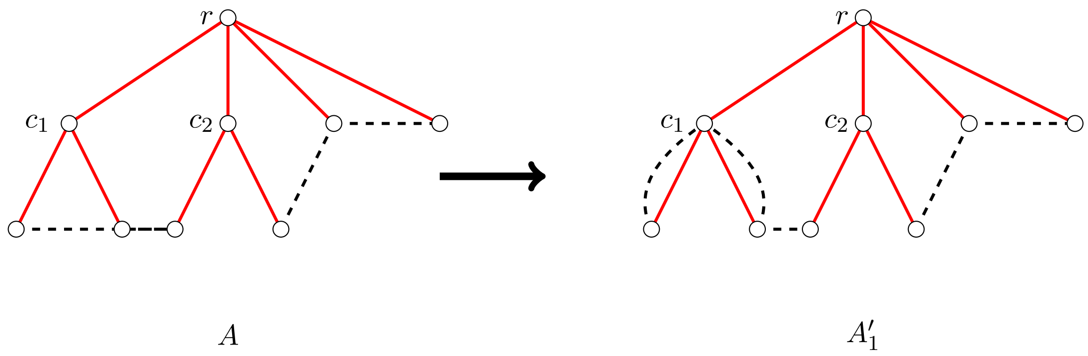

Figure 2: Transformation to a star-shaped instance for the root

solution of the same cost in G: this is because no edge of $A' \cap \cup_{\text{internal nodes } u} \{(u', u'')\}$ is useful in covering the edges of $E(T)$ in the original instance. □

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 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* $\frac{3}{2}$.

Proof. First we show how to transform any integral solution A into a feasible solution to two star-shaped instances, the better of which has value at most $\frac{3}{2} \cdot c(A)$. The same reduction will also apply to fractional solutions that obey the ODD-LP constraints.

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. Given an optimal solution A, partition the links in it into $A = A_1 \dot\cup A_2$ where A_i the set of links whose least common ancestor (henceforth lca) is a node in level i of the tree. (Note that the lca of any link will always be an internal node in any leaf-to-leaf instance like those that we consider).

Consider now two alternate instances with feasible solutions A'_1 and A'_2 as follows.

For an illustration of the first solution A'_1, see Figure 2. For every link (u, v) in A_2 with lca c say, we replace it with two up-links (u, c) and (v, c) of the same cost. Note that this set of links along with A_1 gives a solution to a star-shaped instance centered at the root r. This solution has cost $c(A_1) + 2c(A_2)$. Motivated by the existence of this solution, we can partition all the links $L = E(G) \setminus E(T)$ into $L = L_1 \dot\cup L_2$ where link (u, v) is in L_i if the $lca(u, v)$ is a node in level i of the tree. We then define a star-shaped instance centered at r 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. The minimum cost solution A'_1 we can find to this instance in polynomial time will have cost at most $c(A_1) + 2c(A_2)$.

For an illustration of the second solution A'_2, see Figure 3. In this case, we decompose the problem into $d + 1$ different star shaped instances of which d are from the subtrees defined by the star around each non-leaf child of the root, and the last is from the star defined by the root and its leaf-children. For this case, given a solution A we replace every link (u, v) in A_1 with lca the root r, with two up-links (u, r) and (r, v)

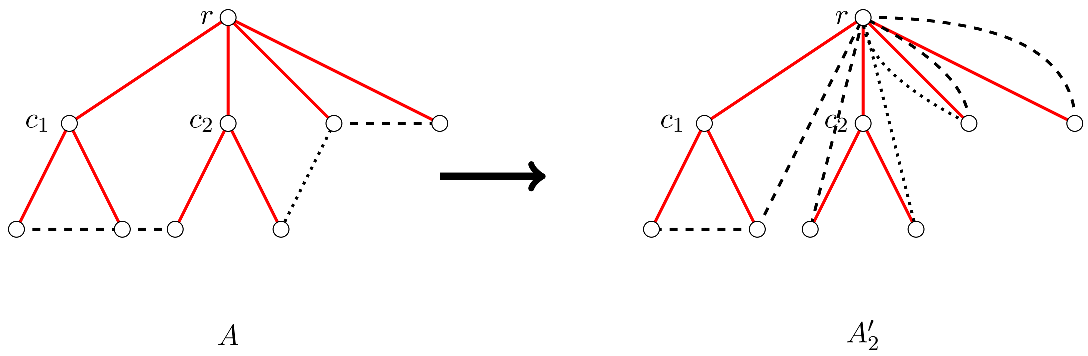

Figure 3: Transformation to three star-shaped instances around the root and its two internal children

of the same cost. Now consider the subtree T_i defined by the star around child c_i in T for $i = 1, \ldots, d$. For every link in A_1 that has an endpoint in this subtree, one of the two copies made above goes from this endpoint to the root r which is one of the leaves of this (star) tree. Similarly, the star around the root made of its leaf-children also has the copies of links in A_1 covering it. It is easy to verify that the subset of A_2 consisting of links with lca c_i along with the copies of the A_1 links defined above give a feasible solution to the weighted TAP on this star T_i. This solution can be found in each such subtree as well as the star around the root, and the sum of the costs of these solutions is $2c(A_1) + c(A_2)$. By defining the appropriate star-shaped subproblems as above, we can find in polynomial time, a solution A'_2 to the overall problem of cost at most $2c(A_1) + c(A_2)$.

Applying the above to the optimal solution A^*, we see that the best of the two solutions found above has cost at most $\min(c(A^*_1) + 2c(A^*_2), 2c(A^*_1) + c(A^*_2)) \leq \frac{3}{2}(c(A^*_1) + c(A^*_2)) = \frac{3}{2}c(A^*)$.

It is not hard to see that the values in any fractional solution on the links for the ODD-LP can be transformed into a feasible fractional solution to these two sets of star shaped instances of value as claimed above. Since the resulting star shaped instances have integrality gap 1 by Theorem 2.1, the claim about the integrality gap also follows. □

Theorem 3.2. *The integrality gap of the ODD-LP for a three-level tree instance is at most* $\frac{7}{4}$.

Proof. As before we will transform any integral solution A into a feasible solution to one of three sets of star-shaped instances 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.

Using Lemma 2.2, we assume that all links go between a pair of leaves. Given an optimal solution A, partition the links in it into $A = A_1 \dot\cup A_2 \dot\cup A_3$ where A_i the set of links whose lca is a node in level i of the tree. 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.

Consider now three alternate solutions A'_1, A'_2 and A'_3 as follows.

First we construct the solution A'_1 (See Figure 4) that uses links in A_1 once. For every link (u, v) in $A_2 \cup A_3$ with lca c say, we replace it with two up-links (u, c) and (v, c) of the same cost. Note that this set of links along with A_1 gives a solution to a star-shaped instance centered at the root r. This solution has cost $c(A_1) + 2c(A_2) + 2c(A_3)$. As before, to find such a solution, we can partition all the links $L = E(G) \setminus E(T)$ into $L = L_1 \dot\cup L_2 \dot\cup L_3$ where link (u, v) is in L_i if the $lca(u, v)$ is a node in level i of the tree. We then define a star-shaped instance centered at r 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. The minimum cost solution A'_1 we can find to this instance in polynomial time will have cost at most $c(A_1) + 2c(A_2) + 2c(A_3)$.

For the second solution (Figure 5), we proceed as before to decompose the problem into one per non-leaf neighbor v_i of the root by considering the whole subtree T_i under it along with its tree edge to the root,

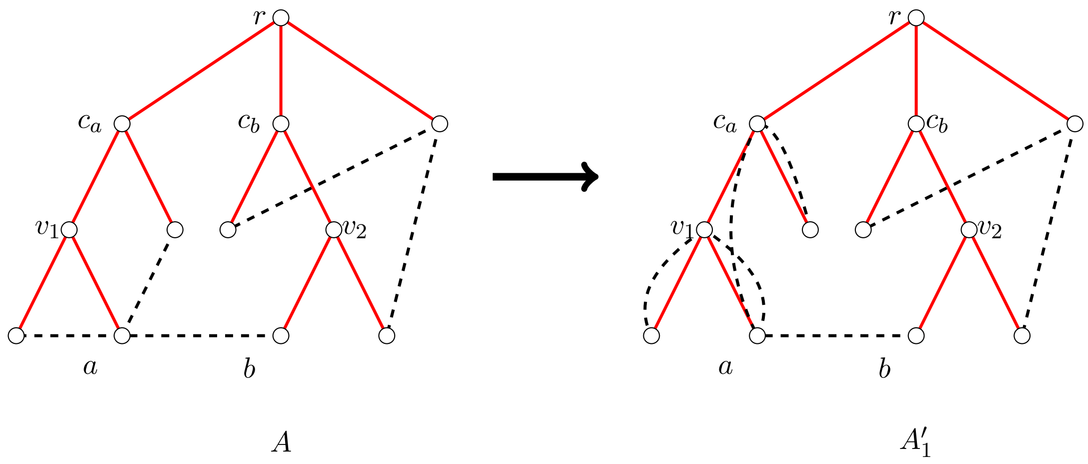

Figure 4: Transformation to a star-shaped instance centered at the root

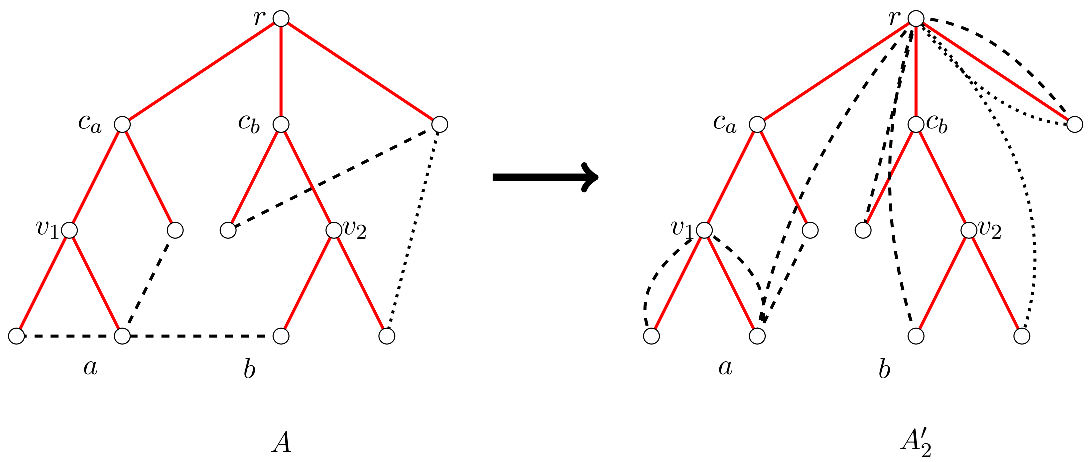

Figure 5: Transformation to three star-shaped instances centered at the root and its two internal children

and one more for the root with its leaf children. For this case, given a solution A we replace every link (u, v) in A_1 with lca the root r, with two up-links (u, r) and (r, v) of the same cost. For every link (u, v) in A_3 with lca v' say, we replace it with two up-links (u, v') and (v, v') of the same cost. Now consider the subtree T_i defined by the non-leaf child c_i in T along with its tree edge to the root for $i = 1, \ldots, d$. For every link in A_1 that has an endpoint in this subtree, one of the two copies made above goes from this endpoint to the root r which is one of the leaves of this tree. As before, the star around the root made of its leaf-children also has the copies of links in A_1 with an endpoint incident to each leaf covering the corresponding leaf child. It is easy to verify that the solution A_2 consisting of links with lca c_i along with the copies of the A_1 links defined above, and the doubled copies of links in A_3 give a feasible solution to the set of $d + 1$ star-shaped instances of the weighted TAP on the T_i's and the root. As before, a solution of at most this cost can be found in suitably defined modified instances and the sum of the costs of these solutions is at most $2c(A_1) + c(A_2) + 2c(A_3)$.

Finally, for the third solution (See Figure 6), we consider the stars around the internal nodes, say $v_1, \ldots, v_q$ in level 3, and one more tree around the root consisting all the set of all tree edges not in the stars around the v_i's. To obtain a set of star-shaped solutions from A for these instances we proceed as follows. For every link (a, b) in A_2 with lca c say, we replace it with two up-links (a, c) and (b, c) of

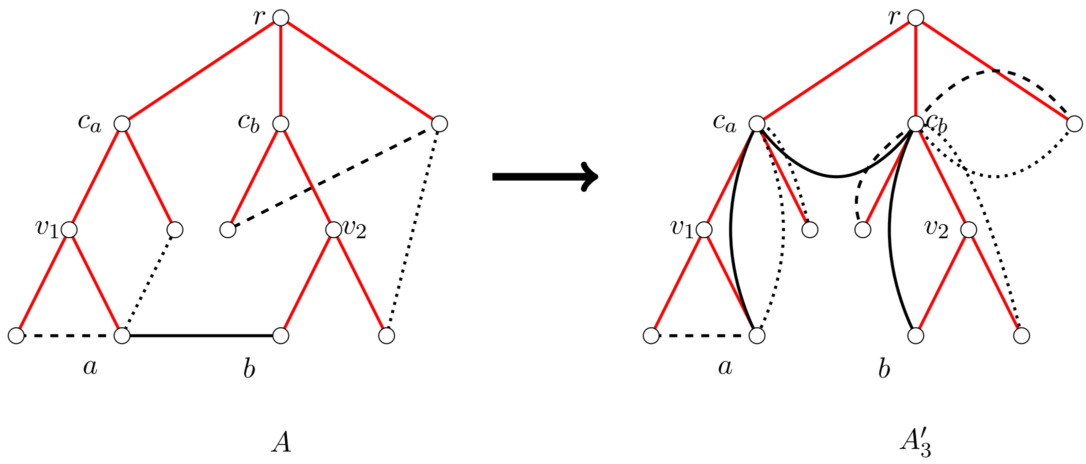

Figure 6: Transformation to three star-shaped instances centered at the root and the stars around the two internal nodes in level 3

the same cost. Note that the lca c is a leaf in one of the third level stars and so all these copies become star-shaped links for those corresponding instances. The interesting transformation is for links in A_1 where we now make up to *three* copies. For every link $(a, b) \in A_1$, let c_a and c_b denote the ancestor of a and b respectively in level 1. (if either a or b is in level 1 itself, then its ancestor in level 1 is itself). We now add three links $(a, c_a), (c_a, c_b), (c_b, b)$ of the same cost as (a, b). Note that the first and third link are leaf to leaf cross links in the stars corresponding to centers v_a and v_b (the ancestors of a and b in level 2 if they exist), and that the middle link (c_a, c_b) is a cross link in the star-shaped instance centered at the root. It is now easy to verify that the copies that we have produced form a set of feasible solutions to these star-shaped instances of total cost at most $3c(A_1) + 2c(A_2) + c(A_3)$.

The best of the above three solutions corresponding to the optimal solution A^* has cost at most $\min(c(A_1^*) + 2c(A_2^*) + 2c(A_3^*), 2c(A_1^*) + c(A_2^*) + 2c(A_3^*), 3c(A_1^*) + 2c(A_2^*) + c(A_3^*)) \leq \frac{7}{4}(c(A_1^*) + c(A_2^*) + c(A_3^*)) = \frac{7}{4}c(A^*)$.

As before, it is not hard to see that the values in any fractional solution on the links for the ODD-LP can be transformed into a feasible fractional solution to these three sets of star shaped instances of value as claimed above. Since the resulting star shaped instances have integrality gap 1 by Theorem 2.1, the claim about the integrality gap also follows.

□

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 $2 - \frac{1}{2^{k-1}}$.*

Proof. We show how to transform any integral solution A into a feasible solution to one of k star-shaped instances. Partition the links in A into subsets of links $A = A_1 \dot\cup A_2 \ldots \dot\cup A_k$ where A_l is the subset whose lca is a node in level l of the tree for $l = 1, \ldots, k$. Denote the cost of these subsets of links by $c_1, \ldots, c_k$ so that the total cost of A is $c = \sum_{l=1}^{k} c_k$. As before we set up k sets of solutions, with the l^{th} solution attempting to use edges in A_l only once.

Note that for $l = 1$, we replace all links in $A_2, \ldots, A_k$ with two links going to the lca and decompose the resulting solution into one for a star-shaped instance around the root. The cost of this candidate solution is

$$C_1 = c_1 + 2c_2 + \ldots + 2c_k.$$

For $1 < l \leq k$, for every internal node v at level l, we consider the subtree below it, along with the edge to its parent and create the solution for this star-shaped instance from the solution A. In addition we create one star-shaped instance around the root, whose tree edges are disjoint from the others, to create a final candidate solution. First consider the star-shaped instances around the internal nodes v in level l. Links in A_l are already cross links in these. For any link $(a, b) \in A_{l-1} \cup \bigcup_{p>l} A_p$, we replace it with the two links $(a, lca(a, b))$ and $(b, lca(a, b))$. Links in A_p for $p > l$ are replaced with two links that become up links in these instances. Consider a link $(a, b) \in A_{l-1}$, 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 $(a, b) \in A_{l-1}$ 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 $(a, b) \in A_q$ for $1 < q < l - 1$, let the ancestors of a and b in level $l - 1$ 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. For $q > 1$, all the links in these sets are cross links for the star-shaped instances around the level l internal nodes or up links for the instance rooted at r. Analogously, for $q = 1$, we can instead use the following sets, with at most three links: $\{(a, u_a), (u_a, u_b), (u_b, b)\}$, $\{(a, u_b), (u_b, b)\}$, $\{(a, u_a), (u_a, b)\}$, $\{(a, b)\}$. In contrast to the $q > 1$ case, these sets also include cross links for the instance rooted at r.

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

$$\begin{aligned} C_2 &= 2c_1 + c_2 + 2c_3 + 2c_4 + \ldots + 2c_k, \text{ if } l = 2 \\ C_3 &= 3c_1 + 2c_2 + c_3 + 2c_4 + \ldots + 2c_k, \text{ if } l = 3 \\ C_l &= 3c_1 + 4c_2 + \ldots + 4c_{l-2} + 2c_{l-1} + c_l + 2c_{l+1} + \ldots + 2c_1, \text{ if } l > 3. \end{aligned}$$

To find the worst case ratio of $\min(C_1, \ldots, C_k)$ and $c_1 + \ldots + c_k$, we show we can set the costs so that all the terms in the numerator are equal.

Setting $C_1 = C_2$ gives $c_1 + 2c_2 + \ldots + 2c_k = 2c_1 + c_2 + 2c_3 + \ldots + 2c_k$ which simplifies to

$$c_1 = c_2.$$

Setting $C_2 = C_3$ gives $2c_1 + c_2 + 2c_3 + \ldots + 2c_k = 3c_1 + 2c_2 + c_3 + 2c_4 + \ldots + 2c_k$ which simplifies to

$$c_3 = 2c_1 = c_1 + c_2.$$

Setting $C_3 = C_4$ gives $3c_1 + 2c_2 + c_3 + 2c_4 + \ldots + 2c_k = 3c_1 + 4c_2 + 2c_3 + c_4 + 2c_5 + \ldots + 2c_k$ which simplifies to

$$c_4 = 2c_2 + c_3.$$

In general, setting $C_l = C_{l+1}$ gives

$$c_{l+1} = 2c_{l-1} + c_l.$$

The worst case ratio is then

$$\frac{C_1}{c_1 + \ldots + c_k} = \frac{c_1 + 2c_2 + \ldots + 2c_k}{c_1 + \ldots + c_k} = 2 - \frac{1}{1 + \sum_{2 \leq l \leq k} 2^{l-2}} = 2 - \frac{1}{2^{k-1}}.$$

□

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.

5. Tight example and a lower bound on the odd-LP

In Theorem 3.1, we showed that it is possible to obtain a feasible TAP solution of weight $c(A_1) + 2c(A_2)$, where $A = A_1 \dot{\cup} A_2$ is an optimal TAP solution. To improve upon the bound in Theorem 3.1, a natural idea is to try to obtain a solution of cost $c(A_1) + \alpha c(A_2)$, where $\alpha < 2$. Note that any strengthening of this form immediately yields an upper bound less than $\frac{3}{2}$ on the integrality gap of the odd-LP for 2-level TAP. However, we show that a direct improvement in this way is impossible.

By Lemma 2.2, without loss of generality we consider a leaf-to-leaf instance (T, L). If $u \in$ odd-LP(T, L), we will write $u = (x, y)$ where x is the projection of u onto the cross-links and y is its projection onto the in-links.

Theorem 5.1. *Let (T, L) be the TAP instance given in Figure 7, where $(x, y) = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 1)$ is an extreme point of* $\text{odd-LP}(T, L)$*. Then $(x, \alpha y) \notin \text{TAP}(T, L)$ for any $\alpha < 2$.*

Proof. Suppose $u = (x, \alpha y) \geq \frac{1}{k} \sum_{i=1}^{k} B_i$ where B_i is the incidence vector of a integral TAP solution. Since ℓ_6 has a value of 1, we can assume without loss of generality that link ℓ_6 appears in B_i for all $i \in [k]$. All cross-links have value $\frac{1}{2}$, so they appear in at most $\frac{k}{2}$ of the integral TAP solutions. We will show that the in-link ℓ_5 must be used in all k integral TAPs.

Note that edge e is covered by exactly two cross-links ℓ_1 and ℓ_2. Thus, each B_i must include at least one of ℓ_1 or ℓ_2 to be feasible. Since each of ℓ_1 and ℓ_2 is used in at most $\frac{k}{2}$ integral solutions, we conclude that every B_i includes exactly one of ℓ_1 and ℓ_2.

The minimal feasible TAP solutions which include ℓ_1 but not ℓ_2 are:

$$\{\ell_1, \ell_6, \ell_3, \ell_4\}, \{\ell_1, \ell_6, \ell_4, \ell_5\}, \text{ and } \{\ell_1, \ell_6, \ell_3, \ell_5\}.$$

The minimal feasible TAP solutions which include ℓ_2 but not ℓ_1 are:

$$\{\ell_2, \ell_6, \ell_3, \ell_4\}, \{\ell_2, \ell_6, \ell_4, \ell_5\}, \text{ and } \{\ell_2, \ell_6, \ell_3, \ell_5\}.$$

Thus, we may assume that each B_i is one of the aforementioned feasible integral TAP solutions. Note that in all six such solutions, two links out of $\{\ell_3, \ell_4, \ell_5\}$ are used. Hence, in total, links from $\{\ell_3, \ell_4, \ell_5\}$ are used $2k$ times over all B_i. By assumption, the cross-links ℓ_3 and ℓ_4 are used at most $\frac{k}{2}$ times. Hence ℓ_5 is used at least k times.

In particular, the in-link ℓ_5 must be used in all k integral TAP solutions in the convex combination. Since its value was only $\frac{1}{2}$, we see that $(x, \alpha y) \notin \text{TAP}(T, L)$ for any $\alpha < 2$. □

5.1. Lower Bound on the odd-LP

Here we demonstrate a lower bound on the integrality gap of the odd-LP, even for 2-level TAP. In the TAP instance in Figure 7, let all links have cost 1 except link ℓ_6 which has cost 0. The optimal integral solution has cost 3. The optimal fractional solution to the odd-LP has cost at most $\frac{5}{2}$, since $(x, y) = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 1)$ is feasible. Hence the integrality gap is at least $\frac{6}{5}$.

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.

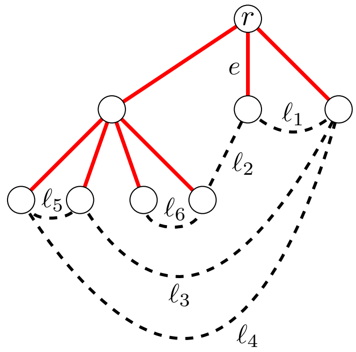

Figure 7: In the above 2-level TAP instance (T, L), the point $(x, y) = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 1)$ is an extreme point for odd-LP(T, L). However, the point $(x, \alpha y)$ is not a convex combination of integral TAP solutions for any $\alpha < 2$.

References

[1] David Adjiashvili. Beating approximation factor two for weighted tree augmentation with bounded costs. *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2384–2399, 2017.

[2] Gautam Appa and Balázs Kotnyek. A bidirected generalization of network matrices. *Networks: An International Journal*, 47(4):185–198, 2006.

[3] Gautam Appa, Balázs Kotnyek, Konstantinos Papalamprou, and Leonidas Pitsoulis. Optimization with binet matrices. *Operations research letters*, 35(3):345–352, 2007.

[4] Alberto Caprara and Matteo Fischetti. {0, 1/2}-chvátal-gomory cuts. *Mathematical Programming*, 74(3):221–235, 1996.

[5] Joseph Cheriyan and Zhihan Gao. Approximating (unweighted) tree augmentation via lift-and-project, part I: stemless TAP. *CoRR*, abs/1508.07504, 2015.

[6] Joseph Cheriyan, Tibor Jordán, and R Ravi. On 2-coverings and 2-packings of laminar families. *Algorithms-ESA'99*, pages 72–72, 1999.

[7] Joseph Cheriyan, Howard Karloff, Rohit Khandekar, and Jochen Könemann. On the integrality ratio for tree augmentation. *Operations Research Letters*, 36(4):399–401, 2008.

[8] Nachshon Cohen and Zeev Nutov. A (1+ ln2)-approximation algorithm for minimum-cost 2-edge-connectivity augmentation of trees with constant radius. *Theoretical Computer Science*, 489:67–74, 2013.

[9] Samuel Fiorini, Martin Groß, Jochen Könemann, and Laura Sanità. Approximating weighted tree augmentation via chvátal-gomory cuts. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 817–831. Society for Industrial and Applied Mathematics, 2018.

[10] Greg N Frederickson and Joseph Jájá. Approximation algorithms for several graph augmentation problems. *SIAM Journal on Computing*, 10(2):270–283, 1981.

[11] Greg N Fredrickson and Joseph Jájá. On the relationship between the biconnectivity augmentation and traveling salesman problem. *Theoretical Computer Science*, 19(2):189–201, 1982.

[12] Fabrizio Grandoni, Christos Kalaitzis, and Rico Zenklusen. Improved approximation for tree augmentation: saving by rewiring. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 632–645. ACM, 2018.

[13] Jennifer Iglesias and R. Ravi. Coloring down: $3/2$-approximation for special cases of the weighted tree augmentation problem, 2017.

[14] Samir Khuller and Ramakrishna Thurimella. Approximation algorithms for graph augmentation. *Journal of Algorithms*, 14(2):214–225, 1993.

[15] Guy Kortsarz, Robert Krauthgamer, and James R Lee. Hardness of approximation for vertex-connectivity network design problems. *SIAM Journal on Computing*, 33(3):704–720, 2004.

[16] Guy Kortsarz and Zeev Nutov. A simplified 3/2 ratio approximation algorithm for the tree augmentation problem. *Transaction on Algorithm*, 12(2):23, 2016.

[17] R. Ravi. *Steiner Trees and Beyond: Approximation Algorithms for Network Design*. PhD thesis, Brown University, 1994.

[18] Vera Traub and Rico Zenklusen. A better-than-2 approximation for weighted tree augmentation, 2021.

[19] Vera Traub and Rico Zenklusen. Local search for weighted tree augmentation and steiner tree, 2021.