A Better-Than-2 Approximation for the Directed Tree Augmentation Problem

Meike Neuwohner* Olha Silina† Michael Zlatin‡

Abstract. We introduce and study a directed analogue of the weighted Tree Augmentation Problem (WTAP). In the weighted Directed Tree Augmentation Problem (WDTAP), we are given an oriented tree $T = (V, A)$ and a set of directed links $L \subseteq V \times V$ with positive costs. The goal is to select a minimum cost set of links which enters each fundamental dicut of T (cuts with one leaving and no entering tree arc). WDTAP captures the problem of covering a cross-free set family with directed links. It can also be used to solve weighted multi 2-TAP, in which we must cover the edges of an undirected tree at least twice. WDTAP can be approximated to within a factor of 2 using standard techniques. We provide an improved $(1.75 + \varepsilon)$-approximation algorithm for WDTAP in the case where the links have bounded costs, a setting that has received significant attention for WTAP. To obtain this result, we discover a class of instances, called "willows", for which the natural set covering LP is an integral formulation. We further introduce the notion of "visibly k-wide" instances which can be solved exactly using dynamic programming. Finally, we show how to leverage these tractable cases to obtain an improved approximation ratio via an elaborate structural analysis of the tree.

1 Introduction The design of networks that are resilient to connection failures constitutes a fundamental task in combinatorial optimization. An important class of network design problems are network *augmentation* problems, in which we are given a graph and a set of additional edges, called *links*, which we seek to add so that the resulting graph achieves the desired connectivity properties. One of the most well-studied network augmentation problems is the Tree Augmentation Problem (TAP). Given an undirected tree $T = (V, E)$ and a set of links $L \subseteq \binom{V}{2}$, TAP asks for a minimum cardinality subset of the links whose addition renders T 2-edge-connected. A solution to TAP must include a link crossing each fundamental cut induced by the edges of T. The fundamental cuts form a cross-free family[1] and can be represented by a laminar set family.

TAP can also be interpreted as a set covering problem on the edges of T, where a link $\ell = \{u, v\}$ covers every edge on the unique u-v-path in T. As TAP is known to NP-hard and APX-hard [18], there has been a long line of research on approximation algorithms for TAP [11, 20, 6, 7, 9, 4, 19], starting with a 2-approximation by Frederickson and JáJá [11] and culminating in the best known approximation ratio of 1.393 [4]. There has further been a lot of research on the *weighted* Tree Augmentation Problem (WTAP), where every link is equipped with a positive cost, and the task is to minimize the total cost of the selected link set. Until a few years ago, no better approximation ratio than 2 was known. Recently, this approximation barrier has been breached, resulting in the best known approximation ratio of $1.5 + \varepsilon$ [23, 24]. Prior to this, several works have considered the bounded cost ratio case, where the ratio between the maximum and the minimum link cost can be bounded by a constant [1, 10, 13]. In this setting, the best known approximation factor is 1.458 [13].

The Directed Tree Augmentation Problem In this paper, we introduce a directed variant of the tree augmentation problem in which both the links and the underlying tree consist of directed arcs. Given an oriented tree $T = (V, A)$ and an arc $a = (u, v)$, we define the *fundamental dicut associated with* a to be the vertex set U of the weakly connected component of $T - a$ containing u. We say that a link $\ell = (x, y)$ *covers* the dicut U if $y \in U$, but $x \notin U$. In the Directed Tree Augmentation Problem (DTAP), we are given an oriented tree $T = (V, A)$ and a set of directed links $L \subseteq V \times V$, and the goal is to cover all fundamental dicuts of T using a minimum cardinality subset of L. Note that the fundamental dicuts of T form a cross-free

*Department of Mathematics, London School of Economics and Political Science
†Department of Mathematics, Carnegie Mellon University
‡Department of Computer Science, Pomona College

[1] A set family $C \subseteq 2^V$ is *cross-free* if for every $A, B \in C$ with $A \cap B \neq \emptyset$ and $A \cup B \neq V$, we have $A \subseteq B$ or $B \subseteq A$.

set family. In fact, using a result of Edmonds and Giles on tree-representations of cross-free families [8], DTAP captures the problem of covering an arbitrary cross-free family with directed links. DTAP can also be seen as a covering problem on the arcs of an oriented tree: a tree arc is *covered* by a directed link (u, v) if the unique path from v to u in T contains the tree arc in the *forward* direction. See Figure 1.1 for an example.

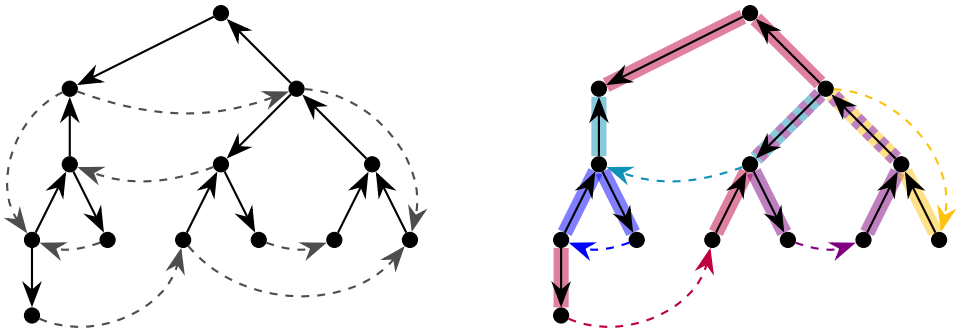


Figure 1.1: A DTAP instance is shown on the left, with links drawn as dashed lines. A feasible solution is shown on the right. Colors indicate the tree arcs that are covered by each link.

Interestingly, DTAP also captures some other natural set covering problems on the edges of a tree. Suppose $T = (V, E)$ is an undirected tree and a set of links is given. The (multi) 2-TAP problem is the problem of selecting a smallest multi-set of links (meaning that we are allowed to select the same link twice) so that each tree edge $e \in E$ is covered at least twice by the links we select. The multi 2-TAP problem reduces to DTAP, see Subsection A.1.

DTAP can be shown to be NP-hard as well as APX-hard, using similar reductions as in [11, 18] for Strong Connectivity Augmentation on oriented trees. For completeness, we give a hardness proof in Subsection A.2.

We further define the *weighted* Directed Tree Augmentation Problem (WDTAP), in which we are given a positive cost for each link, and the task is to minimize the total cost of the selected link set. Like many problems in network design, WDTAP admits a straightforward 2-approximation. To see this, note that WDTAP can be solved in polynomial time when the underlying tree T is an arborescence. Indeed, in this case the constraint matrix of the natural integer programming formulation is a network matrix and hence totally unimodular. This tractable case can be leveraged to obtain a 2-approximation in general. First, choose an arbitrary root vertex r. This partitions the arcs of the tree into *up-arcs* pointing towards the root, and *down-arcs* pointing away from the root. Contracting the up-arcs and down-arcs, respectively, yields two instances of WDTAP in which the oriented tree is a rooted arboresence. Thus, we can cover the up-arcs and down-arcs separately, paying at most the cost of an optimum solution each. The union of these two solutions yields a 2-approximation.

The main result of this paper is a better-than-2 approximation for WDTAP in the case where the *cost ratio* of the instance, the ratio between the maximum and the minimum cost of a link, is bounded.

THEOREM 1.1. *Let $\Delta \geq 1$ and let $\varepsilon > 0$. There exists a polynomial-time $(1.75+\varepsilon)$-approximation algorithm for WDTAP, restricted to instances with cost ratio at most Δ.*

Further related work Many network design problems exhibit a natural approximation barrier of 2. This is in part due to a fundamental result of Jain [16] who gave a unified iterative rounding 2-approximation algorithm for the Survivable Network Design Problem, which captures, e.g., the weighted Tree and Connectivity Augmentation Problem and their Steiner variants [21, 14], the Steiner Forest Problem, and the (weighted) k-Edge-Connected Spanning Subgraph Problem. Jain's algorithm represented the best known approximation ratio for these problems for many decades, and only in recent years we have seen several breakthroughs. As mentioned, the best ratio for WTAP, and in fact, also the weighted Connectivity Augmentation Problem (WCAP) is now $1.5 + \varepsilon$ [24], and recent exciting progress on the Steiner forest has shown that a better-than-2 approximation is possible there as well [2]. If the weights are uniform, both TAP and CAP can be approximated to within a factor of 1.393 [4] while the 2-Edge-Connected Spanning Subgraph Problem admits a slightly better than 1.25-approximation [3, 15]. The weighted 2-Edge-Connected Spanning Subgraph Problem remains at a factor of 2, even in the bounded cost setting.

Directed network design problems are often more challenging to approximate. The Directed Steiner Tree Problem is already set cover hard, see e.g. [5]. In terms of augmentation problems, the Strong Connectivity Augmentation Problem (SCAP) is natural: we are given a weakly connected digraph which we seek to make strongly connected by adding directed

links of cheapest cost. SCAP has seen little progress since Frederickson and Jájá proved that it admits a 2-approximation in the 1980s [11]. It has been shown to be fixed parameter tractable with respect to the solution size [17]. Polyhedral results due to Schrijver [22] show that the linear program of finding a minimum cost strong augmentation of a given digraph D is integral when D is source-sink connected, or when the available arcs are exactly the reverse arcs of those in D. There are no known better-than-2 approximations for strongly connecting an oriented tree, even in the unweighted case.

Organization of the paper The remainder of this paper is organized as follows. In Section 2, we formally define WDTAP and introduce basic notation which will be used throughout the paper. In Section 3, we briefly describe past works for WTAP and how these approaches fail in our setting. In Section 4, we give an overview of our techniques to prove Theorem 1.1. In Section 5, we define the class of "willows" for which the standard LP relaxation turns out to be integral. In Section 6, we characterize instances that can be solved in polynomial time via a standard dynamic programming approach. Sections 7 to 10 present our $(1.75+\varepsilon)$-approximation for WDTAP with bounded cost ratio and its analysis.

2 Preliminaries An instance of WDTAP consists of a directed tree $T=(V,A)$, and directed links $L \subseteq V \times V$ with positive costs $c : L \to \mathbb{R}_{>0}$. For $\ell = (u,v) \in L$, we denote the unique u-v-path in T by P_ℓ. The *(directed) coverage* $\overrightarrow{\mathrm{cov}}(\ell)$ consists of all backward arcs on P_ℓ. To be consistent with the literature on (undirected) tree augmentation, we use $\mathrm{cov}(\ell)$ to denote the set of all arcs on P_ℓ. However, we point out that in the context of WDTAP, a link ℓ only covers the arcs in $\overrightarrow{\mathrm{cov}}(\ell)$, as opposed to all arcs in $\mathrm{cov}(\ell)$. Finally, we write $\overleftarrow{\mathrm{cov}}(\ell) := \mathrm{cov}(\ell) \setminus \overrightarrow{\mathrm{cov}}(\ell)$ to denote the set of arcs on P_ℓ that ℓ "covers in the wrong direction". A set of links $F \subseteq L$ is *feasible* if every tree arc in A is covered by some link in F, i.e., $A \subseteq \bigcup_{\ell \in F} \overrightarrow{\mathrm{cov}}(\ell)$. WDTAP asks for a feasible link set of minimum cost $c(F) := \sum_{\ell \in F} c(\ell)$. For convenience, given an instance of $(T=(V,A),L,c)$ of WDTAP, we will fix a vertex $r \in V$ and call it the *root*. We will call the tuple (T,L,c,r) a *rooted instance* of WDTAP. For $v \in V$, we write $T_v = (U_v, A_v)$ to denote the *subtree rooted at* v. We call arcs that are pointing towards/away from the root *up-arcs* and *down-arcs*, respectively, and we write $A = A_{up} \dot\cup A_{down}$ to denote the partition into up- and down-arcs. For a link $\ell = (u,v)$, we define the *apex* of ℓ to be $\mathrm{apex}(\ell) := \mathrm{lca}(u,v)$ (where $\mathrm{lca}(u,v)$ denotes the least common ancestor of u and v, the vertex on P_ℓ closest to the root). Note that a link $\ell=(u,v)$ *covers* the up-arcs along the v-$\mathrm{apex}(\ell)$-path in T, and the down-arcs along the u-$\mathrm{apex}(\ell)$-path in T. We call a link of the form $\ell=(u,v)$ with $v=\mathrm{apex}(\ell)$ ($u=\mathrm{apex}(\ell)$) an up-link (down-link). A *shadow* of a link $\ell=(u,v)$ is a link of the form $\ell'=(u',v')$, where u' and v' appear in this order on P_ℓ. We may assume without loss of generality that the WDTAP instances we are working with are *shadow-complete*: this means that for every $\ell \in L$, L contains every possible shadow ℓ' of ℓ and moreover, $c(\ell') \le c(\ell)$. Note that unlike the undirected setting where a link ℓ covers all edges on the tree path connecting its endpoints, and, in particular, every shadow has a strictly smaller coverage, this is no longer true in the directed case. For this reason, we define the *generic shadow* $s(\ell)$ of a link ℓ as the minimal shadow of ℓ with $\overrightarrow{\mathrm{cov}}(s(\ell)) = \overrightarrow{\mathrm{cov}}(\ell)$ and let $\overline{P}_\ell := P_{s(\ell)}$. Given a feasible solution to a WDTAP instance, we may always replace each link by its generic shadow without increasing costs or destroying feasibility.

The *cost ratio* of a WDTAP instance is defined as $\Delta = \frac{\max_{\ell \in L} c(\ell)}{\min_{\ell \in L} c(\ell)}$.

The *arc-link-coverage matrix* of an instance of WDTAP is the matrix $M \in \{0,1\}^{A \times L}$ with $M_{a,\ell} = 1$ if and only if $a \in \overrightarrow{\mathrm{cov}}(\ell)$. For subsets $B \subseteq A$ and $L' \subseteq L$, we denote by $M[B,L']$ the submatrix of M with rows indexed by B and columns indexed by L'. Obtaining an optimum solution to WDTAP is equivalent to finding an optimum *integral* solution to the linear program (2.1).

$$\min \left\{ \sum_{\ell \in L} c(\ell) \cdot x_\ell : M \cdot x \ge \mathbf{1}, x \ge 0 \right\}. \tag{2.1}$$

3 Comparison with previous work

3.1 A decomposition-based approach for WTAP with bounded cost ratio ... In order to provide intuition for our algorithm, it is instructive to first describe some of the ideas that are used in [1, 10, 13] to obtain better-than-2-approximations for WTAP with bounded cost ratio[2]. The overall approach pursued in these works consists of two main steps:

1. Design an α-approximation algorithm for a certain class of well-structured instances.
2. Decompose a general instance into subinstances from this class such that α-approximate solutions to the subinstances can be combined to an $(\alpha+\varepsilon)$-approximate solution to the original instance.

[2] In doing so, we will mostly follow the description in [13], but provide a slightly modified perspective on certain arguments to make them align better with the remainder of this paper.

[13] consider the class of so-called *k-wide instances*, where $k \in \mathbb{N}$ is a constant. A (rooted) instance of WTAP is called k-wide if every subtree of a child of the root contains at most k leaves. A 1.5-approximation for k-wide instances can be obtained by trading off two different algorithms: after splitting all cross-links[3] at their apex, a k-wide instance decomposes into a union of independent instances with at most k leaves each. These can be solved exactly using dynamic programming [13]. On the other hand, after splitting every in-link into two up-links, the natural LP relaxation becomes integral after adding so-called *odd cut constraints* [10].

The decomposition into k-wide instances is guided by a solution x to a linear programming relaxation, which is used to estimate the cost of *splitting links* to cut off subtrees as independent subinstances. More precisely, when saying that we obtain the LP solution x' from the LP solution x by splitting a set of links L' at a vertex v, we mean that x' arises from x by, for every $\ell = \{u, w\} \in L'$ such that v is an inner vertex of P_ℓ, setting $x'(\ell) = 0$ and increasing the x'-value on each of the two shadows $\{u, v\}$ and $\{v, w\}$ by $x(\ell)$. The constant cost ratio ensures that up to a constant factor, costs are proportional to x-values; if the total x-value only increases by an ε-fraction, then the cost will also only increase by an $O(\varepsilon)$-fraction. The first step of the decomposition procedure is to contract edges $e \in E(T)$ that are *heavily covered* [1], i.e., for which $x(\{\ell \colon e \in \mathrm{cov}(\ell)\}) \geq \zeta_\varepsilon \coloneqq \frac{2}{\varepsilon}$. These edges can be covered at cost $\varepsilon \cdot c(x)$ [1]. In the second step of the decomposition procedure, the tree is traversed from bottom to top and subtrees hanging off certain inner vertices are split off. More precisely, an edge $e = \{v, w\}$, where w is closer to the root, is called *ε-light*[4] if $x(\{\ell \colon e \in \mathrm{cov}(\ell)\}) \leq \varepsilon \cdot x(\{\ell \colon \mathrm{cov}(\ell) \cap E(T_v) \neq \emptyset\})$, i.e., if the total coverage of e amounts to at most an ε-fraction of the total coverage of T_v. When disattaching the subtree T_v, every link $\ell = \{u, w\}$ that leaves T_v is split at v into two shadows, one whose coverage is contained in T_v and one whose coverage is contained in $T - T_v$. As every split link covers e, the splitting increases the total x-value by at most $x(\{\ell \colon e \in \mathrm{cov}(\ell)\})$. These costs can be charged to the total x-value within the subtree T_v that is subsequently cut off, ensuring that the iterated splitting only increases the total cost by an $O(\varepsilon)$-fraction. At the end of the splitting, every subinstance is $k_\varepsilon \coloneqq 2 \cdot \varepsilon^{-1} \cdot \zeta_\varepsilon$-wide. To see this, let w be the root of a subinstance and let T_v be a subtree hanging off the root. If there are more than k_ε leaves in the subtree, then $x(\{\ell \colon \mathrm{cov}(\ell) \cap E(T_v) \neq \emptyset\}) > \frac{k_\varepsilon}{2}$ because every link can cover the edges incident to at most two leaves. On the other hand, $x(\{\ell \colon \{v, w\} \in \mathrm{cov}(\ell)\}) < \zeta_\varepsilon$ because all heavily covered edges were contracted, implying that $\{v, w\}$ is ε-light. But this means that T_v would have been cut off, a contradiction.

3.2 ... and why it doesn't work for DTAP Given the similarities between WTAP and WDTAP, it appears tempting to transfer the ideas described in Subsection 3.1 from the undirected to the directed setting. In this section, we point out the issues with this approach, before discussing how to resolve them in the following sections. Recall that in order to decompose a WTAP instance (with bounded cost ratio) into k-wide subinstances, while only increasing the total cost by an $O(\varepsilon)$-fraction, we had to ensure two properties: first of all, we argued that subtrees with many leaves attract a large LP value. More precisely, we observed that whenever a subtree T_v hanging off the edge $e = \{v, w\}$ contains more than k_ε leaves, then $x(\{\ell \colon \mathrm{cov}(\ell) \cap E(T_v) \neq \emptyset\})$, the total LP value on links covering edges within the subtree, is large. Second of all, we had to make sure that $x(\{\ell \colon e \in \mathrm{cov}(\ell)\})$, the total LP value of links covering the edge e, is comparably small. Note that the links covering e are precisely the links one has to duplicate to split off the subtree T_v as an independent instance. Hence, combining both properties ensures that the total cost increase incurred by splitting links only amounts to an $O(\varepsilon)$-fraction of the optimum cost.

The first property can also be easily ensured in the directed setting. Let $a = (v, w)$ be an up-arc (down-arcs can be handled analogously) and assume that T_v contains more than k leaves. Again, each link ℓ can cover the arcs incident to at most two leaves; more precisely, $\overrightarrow{\mathrm{cov}}(\ell)$ contains at most one up- and at most one down-arc incident to a leaf. Hence, any solution x to the natural LP relaxation (2.1) will satisfy $x(\{\ell \colon \overrightarrow{\mathrm{cov}}(\ell) \cap A(T_v) \neq \emptyset\}) \geq \frac{k}{2}$. A problem arises, however, with the second condition. In the undirected setting, every link ℓ with $e \in E(P_\ell)$ covers the edge e. This property can be used to ensure that $x(\{\ell \colon e \in \mathrm{cov}(\ell)\})$ is not too large by covering all heavily covered edges at an $O(\varepsilon)$-fraction of the LP cost and then contracting them. In contrast, in the directed setting, this reasoning can only be used to guarantee that no arc is *heavily covered in the right direction*, i.e., that $x(\{\ell \colon a \in \overrightarrow{\mathrm{cov}}(\ell)\})$ is not too large. However, we cannot control whether an arc is *heavily covered in the wrong direction*, i.e., whether $x(\{\ell \colon a \in \overleftarrow{\mathrm{cov}}(\ell)\})$ is large. But in order to fully cut off T_v, we need to be able to bound $x(\{\ell \colon a \in \mathrm{cov}(\ell)\}) = x(\{\ell \colon a \in \overrightarrow{\mathrm{cov}}(\ell)\}) + x(\{\ell \colon a \in \overleftarrow{\mathrm{cov}}(\ell)\})$. Figure 3.1 illustrates an example where we cannot cut off a subtree, even though it contains more than k leaves.

The reader might wonder why we are focusing on the techniques used in [1, 10, 13] that yield better-than-2-approximations for WTAP with bounded cost ratio, and do not consider the more recent works [23, 24] that give better-than-2-approximations

[3] In the undirected setting, a link is called an *up-link* if its apex coincides with one of its endpoints. A *cross-link* is a link whose apex is the root that is not an up-link. All links that are neither up- nor cross-links are called *in-links*.

[4] This term was introduced in [13]. Their definition (slightly) differs from the one presented here because they do not work with a rooted tree.

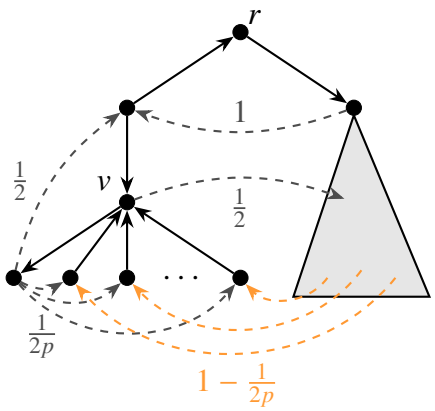


Figure 3.1: An instance of WDTAP, with arcs indicated by solid arrows and links shown as dashed arrows. A solution to (2.1) is indicated next to the links. All links have cost 1. The subtree hanging off the vertex v contains a large number p of leaves that are connected to v via up-arcs. In the given LP solution, these up-arcs are (partially) covered by the orange links, each of which covers the down-arc entering v in the wrong direction. Splitting all of the orange links at v is too expensive.

for (general) WTAP. The reason for this is that the techniques in [23, 24] crucially rely on certain structural properties of undirected solutions, such as the fact that the coverage of a link is connected. As such, it is unclear how to apply them in the directed setting.

4 Our contribution The main result of this paper is a polynomial-time better-than-2-approximation for WDTAP with bounded cost ratio.

THEOREM 1.1. *Let $\Delta \geq 1$ and let $\varepsilon > 0$. There exists a polynomial-time $(1.75+\varepsilon)$-approximation algorithm for WDTAP, restricted to instances with cost ratio at most Δ.*

Our approach towards Theorem 1.1 is inspired by the decomposition strategy pursued in [1, 10, 13] to obtain better-than-2-approximations for WTAP with bounded cost ratio. However, as we discussed in the previous section, WDTAP exhibits fundamental differences to its undirected analogue, which renders a simple adaptation of prior techniques infeasible. Instead of completely decomposing the instance, our approach relies on *partially decomposing* the instance by splitting certain links in such a way that the total cost only increases by an $O(\varepsilon)$-fraction. Our main technical contributions can be summarized as follows:

- We introduce the concept of *bounded visible width*, which allows us to characterize when an instance of WDTAP can be solved in polynomial time using a standard dynamic programming approach.
- We define a new class of instances called *willows* and show that for these instances, the natural LP relaxation (2.1) is integral.
- We discuss how to carefully split certain links to achieve strong structural properties, while only incurring an arbitrary small increase in the total solution cost. Then, we explain how splitting certain subsets of the links results in instances of bounded visible width and willows, respectively, both of which we can solve exactly. Trading off three different solutions results in the final approximation guarantee.

4.1 New notions of partial decomposition: visible width and willows In this section, we introduce the concepts of visible width and up- and down-independence, which allow us to characterize the structural properties that we gain by splitting links, and to leverage these to solve certain types of instances exactly.

Visible width The notion of constant visible width characterizes WDTAP instances that can be solved exactly using a natural dynamic programming approach, similar to the one used for k-wide instances [13]. The basic idea is to traverse the tree from the leaves to the root and to iteratively construct a cheapest solution covering the subtree T_v, given a fixed "interface" to the remaining instance. More precisely, for $v \in V$, let L_v be the set of links ℓ such that v is an inner vertex of $\overline{P}_\ell$. We further partition L_v into the set L_v^{cross} of *v-cross-links* having v as their apex, the set $L_v^{\downarrow} := \{\ell = (u,w) \in L_v : u \notin U_v \wedge w \in U_v \setminus \{v\}\}$ of links *pointing into* and the set $L_v^{\uparrow} := \{\ell = (u,w) \in L_v : u \in U_v \setminus \{v\} \wedge w \notin U_v\}$ of links *pointing out of* T_v. Note

that the links in L_v^{cross} are precisely those that connect the solutions in different subtrees of T_v hanging off v, while the links in $L_v^{\uparrow} \cup L_v^{\downarrow}$ are precisely those creating interactions between T_v and $T - T_v$. In order to obtain an optimum WDTAP solution, it suffices to, for $v \in V$ and $F \subseteq L_v^{\uparrow} \cup L_v^{\downarrow}$, compute a cheapest solution $S(v, F) \subseteq L$ for (T_v, L, c) with the property that $S(v, F) \cap (L_v^{\uparrow} \cup L_v^{\downarrow}) = F$. Then $S(r, \emptyset)$ constitutes an optimum solution to (T, L, c). Of course, the problem with this approach is that in general, there are exponentially many choices for the set F and moreover, we need to be able to control the number of v-cross-links used in a solution in order to merge solutions for the children of v into a solution for v efficiently. Hence, our goal is to characterize instances for which we can guarantee the existence of an optimum solution S such that $|S \cap L_v|$ can be bounded by a constant for every $v \in V$; such a solution can be found in polynomial time using the above-mentioned DP approach. To this end, consider a *shadow-minimal* optimum solution S^*, i.e., an optimum solution in which no link can be replaced by a proper shadow whilst maintaining feasibility. It is not hard to see that every $\ell \in S^*$ covers the first and the last arc of P_ℓ and moreover, it is the unique link in S^* that does so. This implies that the lowest up-arcs in T_v covered by links in $(L_v^{\downarrow} \cup L_v^{cross}) \cap S^*$ are pairwise distinct and form an ancestor-free set of up-arcs, i.e., for two arcs $a \neq a'$ in this set, a' does not appear on the path connecting the top vertex of a to the root. Similarly, the lowest down-arcs in T_v covered by links $(L_v^{\uparrow} \cup L_v^{cross}) \cap S^*$ form an ancestor-free set of down-arcs. Motivated by these observations, we say that a vertex v can *see* an up-arc (a down-arc) $a \in A_v$ if there exists a link $\ell \in L_v^{\downarrow} \cup L_v^{cross}$ ($\ell \in L_v^{\uparrow} \cup L_v^{cross}$) with $a \in \overrightarrow{\text{cov}}(\ell)$. Equivalently, v can see an arc $a \in A_v$ if there is a link ℓ with $a \in \overrightarrow{\text{cov}}(\ell)$ (i.e., ℓ covers a) such that v is an *inner vertex* of $\overline{P}_\ell$. We define the *visible up-width (visible down-width)* at v to be the maximum size of an ancestor-free set of up-arcs (down-arcs) that v can see. The *visible width* of an instance is the maximum over the visible up- and down-widths at the vertices. The previous considerations imply that WDTAP instances with constant visible width can be solved exactly in polynomial time via dynamic programming.

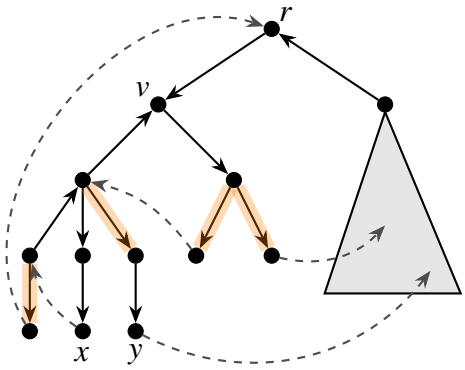


Figure 4.1: The visible width at v is at least 4, as certified by the four orange-shaded arcs in its subtree. This is an ancestor-free set of down-arcs which are all visible to v. However, the black down-arc incident to x, as well as the down-arc above, are not visible to v since they are not covered by a link ℓ for which v is an inner vertex of $\overline{P}_\ell$. The down-arc incident to y is visible to v, but does not form an ancestor-free set with the shaded down-arc above.

Note that the visible width of an instance depends both on the rooted tree as well as the set of links that we are allowed to select. During the course of our algorithm, we will maintain a solution x to (2.1) and we will always compute the visible width with respect to the *support of* x (and its shadows). In particular, modifying x by splitting links can block a vertex from seeing certain arcs in its subtree and decrease the visible width.

Willows For WTAP, [10] have shown that if the instance only contains cross-links and up-links, then the constraint matrix of the natural LP relaxation is a binet matrix, which they use to argue that adding odd cut constraints suffices to guarantee integrality. For WDTAP, it is not hard to see that if the instance only contains cross-links and up- and down-links, then the constraint matrix of (2.1) is totally unimodular. We generalize this result by introducing *willows*, a class of instances that may contain cross-links with respect to multiple "local roots".

Let (T, L, c, r) be a rooted WDTAP instance. We call a vertex $v \in V(T)$ *up-independent (down-independent)* if $L_v^{\downarrow} = \emptyset$ ($L_v^{\uparrow} = \emptyset$). If v is up-independent (down-independent), then the problem of covering the up-arcs (down-arcs) in T_v is "independent from" the problem of covering arcs outside T_v in the sense that no link can cover both. We call (T, L, c, r) a *willow* if there exists a set $W \subseteq V(T)$ such that every vertex in W is up- or down-independent, and every link in L is either

an up-link, a down-link, or a *W-cross-link*, meaning that its apex is contained in W. Note that the root r is always both up- and down-independent.

THEOREM 4.1. *Let (T, L, c, r) be a willow. Then* (2.1) *is integral.*

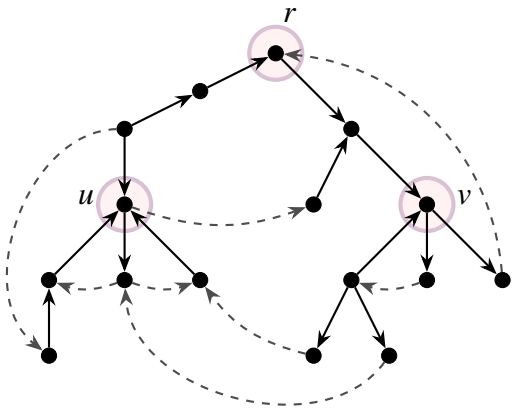


Figure 4.2: A willow (choosing $W = \{r, u, v\}$). Notice that u is down-independent, v is up-independent, and the root r is both. All links are either up-links, down-links, or have their apex in W.

4.2 Our approach

Blue-sky version To provide some intuition how the notions of visible width and willows can be leveraged towards a better-than-2-approximation for WDTAP with bounded cost ratio, assume for a moment that we could prove the following "dream theorem".

Dream Theorem. Let $\varepsilon > 0$ and x a solution to (2.1). We can, in polynomial time, compute a solution x^* to (2.1) of cost $c(x^*) \leq (1+\varepsilon) \cdot c(x)$ that arises from x by splitting links, and a set $W \subseteq V$ such that:

(i) W consists of up- and down-independent vertices with respect to $\operatorname{supp}(x^*) = \{\ell \in L \colon x^*(\ell) > 0\}$.

(ii) Let L' arise from $\operatorname{supp}(x^*)$ by splitting every W-cross-link at its apex. Then (T, L') has visible width at most $k(\varepsilon, \Delta)$ (where $k(\varepsilon, \Delta)$ is some constant depending on ε and the cost ratio Δ of the instance).

Using the dream theorem, we could obtain a $(1.5 + O(\varepsilon))$-approximation for WDTAP with bounded cost ratio as follows: first apply the dream theorem to compute x^* and W subject to (i)-(ii). Let L^*_{cross} denote the set of W-cross-links in $\operatorname{supp}(x^*)$. By splitting all links in $\operatorname{supp}(x^*) \setminus L^*_{cross}$ that are neither up- nor down-links at their apex, we obtain a willow. Hence, we can compute a solution of cost at most $c(x^*) + \sum_{\ell \in L^*_{cross}} c(\ell) \cdot x^*(\ell)$ by Theorem 4.1. By splitting all links in L^*_{cross} at their apex, we obtain an instance of visible width at most $k(\varepsilon, \Delta)$ by (ii), which we can solve optimally. If we could argue that this solution costs at most $c(x^*) + \sum_{\ell \in L \setminus L^*_{cross}} c(\ell) \cdot x^*(\ell)$, then taking the best of the two solutions gives a solution of cost at most $1.5 \cdot c(x^*) \leq 1.5 \cdot (1+\varepsilon) \cdot c(x)$. There is the slight issue that the optimum solution found by the DP might be more expensive than the LP "suggests". To remedy this, we can embed our algorithm into the partial separation framework from [1] that has also been used in [13] to obtain a solution with the appropriate cost relative to the LP (see Section 7 for the details). In each step, our *partial separation oracle* will either find a $(1.75 + O(\varepsilon))$-approximate solution, or a *violated visibly k-wide modification inequality*. Visibly k-wide modification inequalities are valid constraints for the integer hull of (2.1) that, loosely speaking, enforce that our LP solution is not "too cheap" on "subinstances" of visible width at most k. See Definition 7.11 for a formal definition.

Coming down to earth The problem with the blue-sky approach is that the dream theorem is not true, essentially due to the issue with heavy coverage in the "wrong direction" outlined in Subsection 3.2. However, we can prove a *weaker version* of the dream theorem that tells us that heavy coverage in the wrong direction is, in fact, *the only issue* that we have to handle.

Given $\varepsilon > 0$, we fix constants $\zeta_1 \ll \zeta_2 \ll k$ (see Section 8 for the precise values). ζ_1 and ζ_2 will be thresholds for considering an arc to be heavily covered in the "right" and "wrong direction", respectively. k will be the bound on the visible width of instances that we aim for. In the following, we describe our partial separation oracle, that, given a solution x to (2.1),

either finds a violated visibly k-wide modification inequality, or a solution of cost at most $(1+\varepsilon)^2 \cdot 1.75 \cdot c(x)$. As a first step, similar to the outline in Subsection 3.1, we will contract every arc a that is ζ_1-*covered*, i.e., satisfies $x(\{\ell\colon a \in \overrightarrow{\mathrm{cov}}(\ell)\}) \geq \zeta_1$. For ζ_1 chosen large enough, these can be covered at a total cost of $\varepsilon \cdot c(x)$. Hence, we may assume in the following that there are no ζ_1-covered arcs. We further call an arc a ζ_2-*heavy* if $x(\{\ell\colon a \in \overleftarrow{\mathrm{cov}}(\ell)\}) \geq \zeta_2$. Finally, for $v \in V \setminus \{r\}$, let a_v be the first arc on the v-r-path in T. We are now ready to state our weaker version of the dream theorem:

THEOREM 4.2. *Let $\varepsilon > 0$ and x a solution to* (2.1)*. We can, in polynomial time, compute a solution x^* to* (2.1) *of cost $c(x^*) \leq (1+\varepsilon) \cdot c(x)$ that arises from x by splitting links, and a set $W \subseteq V$ such that:*

(i) W consists of up- and down-independent vertices with respect to $\mathrm{supp}(x^*) = \{\ell \in L\colon x^*(\ell) > 0\}$.

(ii) Let L' arise from $\mathrm{supp}(x^*)$ *by splitting every W-cross-link at its apex.*

  (a) The visible width at r, as well as at every $v \in V \setminus \{r\}$ for which a_v is not ζ_2-heavy, is at most k.

  (b) For every $v \in V \setminus \{r\}$ such that a_v is a ζ_2-heavy up-arc (down-arc), the visible up-width (down-width) at v is at most k.

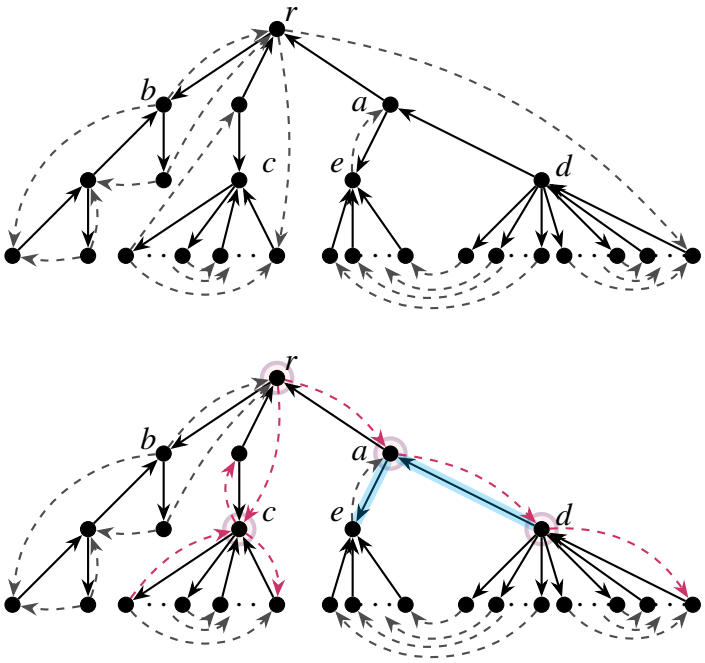


Figure 4.3: Illustration of Theorem 4.2. (Top) A DTAP instance and a solution x to (2.1) whose support is shown as dashed links. (Bottom) The resulting solution x^* with $W = \{a, c, d, r\}$, where arcs a_e and a_d are ζ_2-heavy (light blue). The links contained in the support of x^*, but not in the support of x, are shown in purple. All vertices except a, c, d, e have visible width at most $k = 3$. After splitting W-cross-links, a and c will have visible width 0, d will have visible up-width 0 and e will have visible down-width 0.

In the following, we will sketch how to prove the weakened dream theorem using ideas similar to the decomposition approach used for WTAP. The heavy lifting happens in Subsection 4.3, where we explain how to handle heavy arcs and leverage the weakened dream theorem to obtain the desired approximation guarantee.

Proving Theorem 4.2 Our strategy to prove Theorem 4.2 is similar to the decomposition approach for WTAP with bounded cost ratio described in Subsection 3.1. However, instead of completely cutting off certain subtrees, we will only split certain links covering light arcs. This requires a more intricate scheme to bound the cost of the splitting and avoid "overcharging". We fix a constant $\gamma \sim \varepsilon$. For $v \in V \setminus \{r\}$, we say that the arc a_v is γ-*up-light* if $x(L_v^{\downarrow}) \leq \gamma \cdot x(\{\ell \in L\colon \ell \text{ covers an up-arc in } T_v \text{ that } v \text{ can see}\} \setminus L_v^{\downarrow})$, where visibility is defined with respect to the support of x. Analogously, we define the notion of γ-*down-light* arcs by replacing $L_v^{\downarrow}$ by $L_v^{\uparrow}$ and "up-arc" by "down-arc". We obtain

an LP solution x^* and a vertex set W as stated in Theorem 4.2 as follows. We traverse $V \setminus \{r\}$ in order of decreasing distance to the root (i.e., from bottom to top). For $v \in V \setminus \{r\}$, if a_v is γ-up-light, we split every link in $L_v^{\downarrow}$ at v and add v to W. Similarly, if a_v is γ-down-light, we split every link in $L_v^{\uparrow}$ at v and add v to W. Finally, add r to W. By construction, every vertex in W is up- or down-independent. To bound the total cost increase by $O(\varepsilon) \cdot c(x)$, we observe that if $\ell \in L \setminus L_v^{\downarrow}$ covers an up-arc in T_v, then ℓ only covers arcs within T_v; otherwise, we had $\ell \in L_v^{\downarrow}$. When splitting all links in $L_v^{\downarrow}$ at v, every up-arc in T_v becomes invisible to every vertex outside T_v. Hence, $x(\ell)$ will be "charged against" for splitting at a γ-up-light arc at most once, and the same reasoning applies for γ-down-light arcs. Hence, $\gamma \sim \varepsilon$ yields the desired cost bound. Finally, we sketch how to derive (ii). We will only explain how to bound the visible up-width at every vertex; the visible-down width can be handled analogously. As $r \in W$, the visible (up)-width at r is 0 after splitting all W-cross-links at their apex. Next, let $v \in V \setminus \{r\}$ and assume that the visible up-width at v is greater than k. This means that there is an ancestor-free set A' of at least $k+1$ up-arcs in T_v that are all visible from v. Given that no link can cover two up-arcs from an ancestor-free set simultaneously, this implies $x(\{\ell \in L\colon \ell \text{ covers an up-arc in } T_v \text{ that } v \text{ can see}\}) > k$ holds at the end of the splitting procedure and by the order in which vertices are considered, it also holds when we look at v. If $x(L_v^{\downarrow}) < \zeta_2$, then by our choice of constants, a_v is γ-up-light. Hence, every link in $L_v^{\downarrow}$ is split at v, $v \in W$ and after splitting all W-cross-links at their apex, the visible up-width at v is $0 < k$, a contradiction. So we must have $x(L_v^{\downarrow}) \geq \zeta_2 > \zeta_1$. Note that if a_v is an up-arc, then $a_v \in \overrightarrow{\mathrm{cov}}(\ell)$ for every $\ell \in L_v^{\downarrow}$; otherwise $a_v \in \overleftarrow{\mathrm{cov}}(\ell)$ for every $\ell \in L_v^{\downarrow}$. As we contracted all ζ_1-covered arcs, a_v must be a down-arc that is ζ_2-heavy.

4.3 Components and cores: handling coverage in the wrong direction

An instructive special case To explain how we handle ζ_2-heavy arcs, it is helpful to first consider the slightly artificial, but instructive special case in which for every ζ_2-heavy arc a_v, the parent arc (if exists) is oppositely oriented. More precisely, we assume that if $a_v = (v, w)$ is a ζ_2-heavy up-arc and $w \neq r$, then a_w is a down-arc, and if $a_v = (w, v)$ is a ζ_2-heavy down-arc and $w \neq r$,then a_w is an up-arc. In this situation, we can again obtain a $(1.5 + O(\varepsilon))$-approximation using the following result:

Theorem 4.3. *We can, in polynomial time, compute a solution x^{**} to (2.1) of cost $c(x^{**}) \leq (1+\varepsilon) \cdot c(x^*)$ that arises from x^* by splitting links, and $X \subseteq V$ such that:*

(i) X consists of up- and down-independent vertices with respect to $\mathrm{supp}(x^{**}) = \{\ell \in L\colon x^{**}(\ell) > 0\}$.

(ii) Let L' arise from $\mathrm{supp}(x^{**})$ by splitting every X-cross-link at its apex. For every $v \in V \setminus \{r\}$ such that a_v is a ζ_2-heavy up-arc (down-arc), the visible down-width (up-width) of v w.r.t. L' is 0.

Before proving Theorem 4.3, let us first discuss how to leverage it to obtain the desired approximation guarantee. As splitting links can only reduce the visible width, Theorem 4.2 (ii) and Theorem 4.3 (ii) tell us that after splitting every $(W \cup X)$-cross-link in $\mathrm{supp}(x^{**})$, we obtain an instance of visible width at most k. On the other hand, as splitting links cannot destroy up- or down-independence, Theorem 4.2 (i) and Theorem 4.3 (i) tell us that after splitting every link in $\mathrm{supp}(x^{**})$ that is not a $(W \cup X)$-cross-link at its apex, we obtain a willow. Hence, we may proceed as in Subsection 4.2 to obtain a solution of cost $(1.5 + O(\varepsilon)) \cdot c(x)$.

Proving Theorem 4.3 We obtain x^{**} as follows: for $v \in V \setminus \{r\}$ such that $a_v = (v, w)$ is a ζ_2-heavy up-arc, we split every link in $L_v^{\downarrow}$ at v and every link in $L_w^{\uparrow}$ at w. We add v and w to X. Similarly, for $v \in V \setminus \{r\}$ such that $a_v = (w, v)$ is a ζ_2-heavy down-arc, we split every link in $L_v^{\uparrow}$ at v and every link in $L_w^{\downarrow}$ at w. Again, we add v and w to X. Property (i) is clear by construction. For property (ii), let w.l.o.g. $v \in V \setminus \{r\}$ such that $a_v = (v, w)$ is a ζ_2-heavy up-arc. Then L' doesn't contain any v-cross-link. Moreover, every link in $\ell \in L_v^{\uparrow} \cap L'$ has to end at w because all w-cross-links and all links in $L_w^{\uparrow}$ were split at w. In particular, as the up-link a_v is not covered by ℓ, v is not an inner vertex of $\overline{P}_\ell$. But this implies that v cannot see any down-arc in T_v. It remains to bound the cost of the splitting. Again, let $v \in V \setminus \{r\}$ such that $a_v = (v, w)$ is a ζ_2-heavy up-arc. We know that $x(L_v^{\downarrow}) \leq x(\{\ell\colon a_v \in \overrightarrow{\mathrm{cov}}(\ell)\} < \zeta_1$ because there are no ζ_1-covered arcs. Similarly, $x(L_w^{\uparrow}) < \zeta_1$ because if $w = r$, then $L_w^{\uparrow} = \emptyset$, and otherwise, a_w is a down-arc. On the other hand, $x(\{\ell\colon a_v \in \overleftarrow{\mathrm{cov}}(\ell)\}) \geq \zeta_2$, and $\{\ell\colon a_v \in \overleftarrow{\mathrm{cov}}(\ell)\} \subseteq \{\ell\colon a_w \in \overrightarrow{\mathrm{cov}}(\ell)\} \cup \{\ell\colon \mathrm{apex}(\ell) = w\}$, if $w \neq r$, and $\{\ell\colon a_v \in \overleftarrow{\mathrm{cov}}(\ell)\} \subseteq \{\ell\colon \mathrm{apex}(\ell) = w\}$ otherwise. This implies that $x(\{\ell\colon \mathrm{apex}(\ell) = w\}) \geq \zeta_2 - \zeta_1$ because a_w, if exists, is not ζ_1-covered. Using $\zeta_1 \ll \zeta_2$, we can charge the splitting of the links in $L_v^{\downarrow}$ and $L_w^{\uparrow}$ against the total costs of the links with apex w.

The general case To handle the general case, we consider connected *components* of the (oriented) forests (V, A_{up}) and (V, A_{down}), and define the *core of a component* to consist of all of the paths connecting ζ_2-heavy arcs in the component to its root. We denote the set of vertices and arcs that are contained in a core C by V_C and A_C, respectively. Note that the special

case we considered corresponds to the situation in which every core has depth 1. While the proof of Theorem 4.3 extends to the case where every core has constant depth, this approach is too costly in general. Instead, we perform a more involved trade-off between three different solutions, obtaining a solution of cost at most $(1.75 + O(\varepsilon)) \cdot c(x)$, or a violated visibly k-wide modification inequality. To this end, we define L_{cross} to be the collection of all $W \cup V_C$-cross-links, where W is the vertex set from Theorem 8.3. We define $\overrightarrow{L} := \{\ell \colon \overrightarrow{\text{cov}}(\ell) \cap A_C \neq \emptyset\}$ and $\overleftarrow{L} := \{\ell \colon \overleftarrow{\text{cov}}(\ell) \cap A_C \neq \emptyset\}$ to be the sets of links covering a core arc in the right or in the wrong direction, respectively. Via carefully designed splitting operations, that only increase the total costs by an $O(\varepsilon)$-fraction, we can ensure useful structural properties, including $\overrightarrow{L} \cap \overleftarrow{L} = \emptyset$. Moreover, we can establish the following three statements.

1. Splitting all links in $\overleftarrow{L} \cup L_{cross}$ yields an instance of constant visible width.
2. Splitting all links in $\overleftarrow{L}$ and all links in $L \setminus L_{cross}$ at their apex yields a willow.
3. Splitting all links in $L \setminus \overleftarrow{L}$ at their apex and every link in $\overrightarrow{L}$ once more yields an instance corresponding to the disjoint union of willows.

Taking an appropriate weighted average of these three solutions yields an approximation guarantee of $1.75 + O(\varepsilon)$.

5 Total unimodularity for willows Let $(T = (V, A), L, c, r)$ be a rooted WDTAP instance, and recall the linear programming relaxation given in (2.1). This is not an integral formulation in general; see Subsection A.3. In this section, we derive sufficient conditions for the incidence matrix M to be totally unimodular, yielding an integral formulation and allowing us to solve the corresponding WDTAP instance in polynomial time.

We begin by formally defining the notions of up- and down-independence and willows introduced in Section 4.

Definition 5.1. *We say that $v \in V$ is up-independent with respect to $L' \subseteq L$ if for every $\ell \in L'$, we have $\overrightarrow{\text{cov}}(\ell) \cap A_v \cap A_{up} = \emptyset$ or $\overrightarrow{\text{cov}}(\ell) \subseteq A_v$. We say that v is down-independent with respect to L' if for every $\ell \in L'$, we have $\overrightarrow{\text{cov}}(\ell) \cap A_v \cap A_{down} = \emptyset$ or $\overrightarrow{\text{cov}}(\ell) \subseteq A_v$.*

We recap the following definitions from Section 2 and Section 4. We call a link $\ell = (u, v)$ an *up-link* if $v = \text{apex}(\ell)$ and a *down-link* if $u = \text{apex}(\ell)$. For a set of vertices W, we call ℓ a *W-cross-link* if $\text{apex}(\ell) \in W$ and ℓ is neither an up- nor a down-link.

Definition 5.2. *We call a rooted WDTAP instance (T, L, c, r) a willow if there is a vertex set $W \subseteq V(T)$ such that*

- *every vertex in W is up- or down-independent with respect to L and*
- *every link in L is an up-link, a down-link or a W-cross-link.*

Theorem 5.3 (unimodularity theorem). *Let (T, L, c, r) be a willow and let M be its arc-link-coverage matrix. Then M is totally unimodular. In particular, an optimum integral solution to* (2.1) *can be found in polynomial time.*

For the proof, it is convenient to introduce the following additional notation: For an arc a, we call the endpoint of a that is closer to the root the *apex* of a and denote it by $\text{apex}(a)$. Given two vertices u and v of T, we write P_{uv} to denote the u-v-path in T.

Proof of Theorem 5.3. Let $T = (V, A)$ and let W be as in Definition 5.2. We may assume $r \in W$ because r is both up- and down-independent. To establish total unimodularity of M, we use the criterion by Ghouila-Houri [12]. It states that a matrix $A \in \{-1, 0, 1\}^{m \times n}$ is totally unimodular if and only if for every subset $R \subseteq \{1, \dots, m\}$ of the rows, there exists a signing $\sigma \colon R \to \{-1, +1\}$ such that for every column $j \in \{1, \dots, n\}$, $\sum_{i \in R} \sigma(i) \cdot A_{ij} \in \{-1, 0, 1\}$, where A_{ij} denotes the entry of A in row i and column j.

Applying this to our setting where rows correspond to arcs and columns correspond to links, we need to prove that for every $B \subseteq A$, there exists a signing $\sigma \colon B \to \{-1, +1\}$ such that

$$\text{for every } \ell \in L \quad \sum_{a \in \overrightarrow{\text{cov}}(\ell) \cap B} \sigma(a) \in \{-1, 0, 1\}. \tag{5.1}$$

For two vertices v and w, we define $\text{dist}_{up}(v, w)$ and $\text{dist}_{down}(v, w)$ to be the number of up- and down-arcs from B on the v-w-path in T, respectively. To construct the signing, we define starting signs $\varphi_{up}, \varphi_{down} \colon W \to \{-1, +1\}$ in order of increasing distance (in all of T) to the root r.

- We set $\varphi_{up}(r) = +1$ and $\varphi_{down}(r) = -1$.
- Let $u \in W$ be up-independent and let $v \in W \setminus \{u\}$ be the next vertex after u on the u-r-path in T. We define $\varphi_{down}(u) \coloneqq \varphi_{down}(v) \cdot (-1)^{\mathrm{dist}_{down}(u,v)}$ and $\varphi_{up}(u) \coloneqq -\varphi_{down}(u)$.
- Let $u \in W$ be down-independent (but not up-independent) and let $v \in W \setminus \{u\}$ be the next vertex after u on the u-r-path in T. We define $\varphi_{up}(u) \coloneqq \varphi_{up}(v) \cdot (-1)^{\mathrm{dist}_{up}(u,v)}$ and $\varphi_{down}(u) \coloneqq -\varphi_{up}(u)$.

For an arc a, let $\mu(a)$ be the first vertex (i.e., then one closest to $\mathrm{apex}(a)$) from W on the $\mathrm{apex}(a)$-r-path in T.

- For an up-arc $a \in B$, we set $\sigma(a) = \varphi_{up}(\mu(a)) \cdot (-1)^{\mathrm{dist}_{up}(\mathrm{apex}(a),\mu(a))}$.
- For a down-arc $a \in B$, we set $\sigma(a) = \varphi_{down}(\mu(a)) \cdot (-1)^{\mathrm{dist}_{down}(\mathrm{apex}(a),\mu(a))}$.

Figure 5.1 shows an example of this signing for $B = A$.

Claim 5.4. *Let $a \in A_{up} \cap B$ and let $u \in W \cap V(P_{\mathrm{apex}(a)r})$. Assume that no vertex in $W \cap V(P_{\mathrm{apex}(a)u}) \setminus \{u\}$ is up-independent. Then $\sigma(a) = \varphi_{up}(u) \cdot (-1)^{\mathrm{dist}_{up}(\mathrm{apex}(a),u)}$.*

Proof of claim. Let $W \cap V(P_{\mathrm{apex}(a)u}) = (\mu(a) = u_s, \dots, u_0 = u)$ with $u_s, \dots, u_0$ appearing in this order when traversing $P_{\mathrm{apex}(a)u}$ from $\mathrm{apex}(a)$ to u. Using $\varphi_{up}(u_i) = \varphi_{up}(u_{i-1}) \cdot (-1)^{\mathrm{dist}_{up}(u_i,u_{i-1})}$ for $i = 1, \dots, s$, we obtain

$$\begin{aligned}\sigma(a) &= \varphi_{up}(u_s) \cdot (-1)^{\mathrm{dist}_{up}(\mathrm{apex}(a),u_s)} = \varphi_{up}(u) \cdot (-1)^{\mathrm{dist}_{up}(\mathrm{apex}(a),u_s)+\sum_{i=1}^{s} \mathrm{dist}_{up}(u_i,u_{i-1})} \\ &= \varphi_{up}(u) \cdot (-1)^{\mathrm{dist}_{up}(\mathrm{apex}(a),u)}.\end{aligned}$$

∎

Analogously, we obtain the following claim.

Claim 5.5. *Let $a \in A_{down} \cap B$ and let $u \in W \cap V(P_{\mathrm{apex}(a)r})$. Assume that no vertex in $W \cap V(P_{\mathrm{apex}(a)u}) \setminus \{u\}$ is down-independent. Then $\sigma(a) = \varphi_{down}(u) \cdot (-1)^{\mathrm{dist}_{down}(\mathrm{apex}(a),u)}$.*

Claim 5.6. *Let $\ell = (u, v) \in L$ and let a and a' be two up-arcs in B that appear consecutively on $P_{\mathrm{apex}(\ell)v}$. Then $\sigma(a') = -\sigma(a)$.*

Proof of claim. Let $a = (x, y)$ and $a' = (x', y')$ and assume w.l.o.g. that a appears before a' on $P_{\mathrm{apex}(\ell)v}$ (traversing it from $\mathrm{apex}(\ell)$ to v), i.e., a is above a'. No vertex $v \in V(P_{xy'})$ is up-independent because $a' \in \overrightarrow{\mathrm{cov}}(\ell) \cap A_v \cap A_{up}$ and $a \in \overrightarrow{\mathrm{cov}}(\ell) \setminus A_v$. As $\mu(a)$ is the first vertex from W on P_{yr}, we can apply Claim 5.4 to conclude that

$$\sigma(a') = \varphi_{up}(\mu(a)) \cdot (-1)^{\mathrm{dist}_{up}(y',\mu(a))} = (-1) \cdot \varphi_{up}(\mu(a)) \cdot (-1)^{\mathrm{dist}_{up}(y,\mu(a))} = -\sigma(a)$$

because a and a' are consecutive up-arcs from B on $P_{\mathrm{apex}(\ell)v}$, i.e., $\mathrm{dist}_{up}(y', \mu(a)) = \mathrm{dist}_{up}(y, \mu(a)) + 1$. ∎

Analogously, we obtain the following claim:

Claim 5.7. *Let $\ell = (u, v) \in L$ and let a and a' be two down-arcs in B that appear consecutively on $P_{u\mathrm{apex}(\ell)}$. Then $\sigma(a') = -\sigma(a)$.*

Now, we are ready to show that our signing satisfies (5.1). Let $\ell \in L$. If $\overrightarrow{\mathrm{cov}}(\ell) \cap B$ consists of only up- or only down-arcs, this follows from Claim 5.6 or Claim 5.7, respectively. Finally, assume that ℓ is a W-cross-link such that $\overrightarrow{\mathrm{cov}}(\ell) \cap B$ contains at least one up- and one down-arc. Let $\mathrm{apex}(\ell) = u$ and let $a = (x, y)$ and $a' = (x', y')$ be the up- and the down-arc in $\overrightarrow{\mathrm{cov}}(\ell) \cap B$ closest to u. No vertex $v \in V(P_{yu}) \setminus \{u\}$ is up-independent because $a \in \overrightarrow{\mathrm{cov}}(\ell) \cap A_{up} \cap A_v$ and $a' \in \overrightarrow{\mathrm{cov}}(\ell) \setminus A_v$. No vertex $v \in V(P_{x'u}) \setminus \{u\}$ is down-independent because $a' \in \overrightarrow{\mathrm{cov}}(\ell) \cap A_{down} \cap A_v$ and $a \in \overrightarrow{\mathrm{cov}}(\ell) \setminus A_v$. By Claim 5.4 and Claim 5.5, using that a and a' are the up-/down-arc in $\overrightarrow{\mathrm{cov}}(\ell) \cap B$ closest to u, we get $\sigma(a) = \varphi_{up}(u) = -\varphi_{down}(u) = -\sigma(a')$. Claim 5.6 and Claim 5.7 allow us to conclude that the signs of the arcs in $\overrightarrow{\mathrm{cov}}(\ell) \cap B$ alternate along P_ℓ, implying $\sum_{a \in \overrightarrow{\mathrm{cov}}(\ell) \cap B} \sigma(a) \in \{-1, 0, 1\}$ as desired.

The fact that M is totally unimodular implies that all vertex solutions to the linear program (2.1) are integral. We can find an optimum vertex solution in polynomial time, giving an optimum solution to (T, L, c, r). ∎

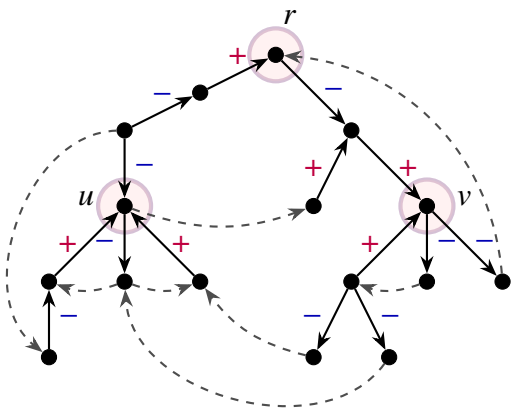


Figure 5.1: The signing constructed in the proof of Theorem 5.3 for the willow from Figure 4.2 and $B = A$ (the set of all arcs).

6 Dynamic program for instances of constant visible width In this section, we define the notion of the *visible width* of a WDTAP instance. We then show that WDTAP instances with constant visible width can be solved exactly using a dynamic program. Let $(T = (V, A), L, c)$ be an instance of WDTAP and let T be rooted at $r \in V$. We introduce some further common terminology that we will use in the following. Given $v \in V \setminus \{r\}$, we call the endpoint of a_v other than v the *parent* of v. (Recall that a_v is the first arc on the v-r-path in T.) We call a vertex w that has v as its parent a *child* of v. For $v \in V$, we say that a vertex w is an *ancestor* of v if w lies on the v-r-path in T, and we say that v is a *descendant* of w. If in addition, $w \neq v$, we call w a *strict ancestor* of v and v a *strict descendant* of w. Note that for $v \in V$, the set of descendants of v is U_v, the vertex set of T_v. Finally, we call an arc a' an *ancestor* of another arc a if a' appears on the $\mathrm{apex}(a)$-r path in T. In order to formally introduce the concept of visible width, we need the notion of an *ancestor-free* arc set.

Definition 6.1. *We call an arc set $F \subseteq A$ ancestor-free if there are no arcs $a, a' \in F$ such that a' appears in the $\mathrm{apex}(a)$-r path in T.*

We now define the notion of which arcs in the subtree of v are visible to v.

Definition 6.2. *We say that an arc $a \in A_v$ is visible to a vertex $v \in V$ (with respect to a set of links L') if there exists a link $\ell \in L'$ such that $a \in \overrightarrow{\mathrm{cov}}(\ell)$ and $v \in \mathrm{in}(\overline{P}_\ell)$, where $\mathrm{in}(\overline{P}_\ell)$ denotes the set of inner vertices of $\overline{P}_\ell$. We denote by $A_v^{vis}(L')$ the set of arcs that are visible from v with respect to L'.*

Definition 6.3. *For a vertex $v \in V$, we define the visible up-width, denoted by $\mathrm{viwidth}_{up}(v)$ (visible down-width, denoted by $\mathrm{viwidth}_{down}(v)$) at v to be the maximum size of an ancestor-free set of up-arcs (down-arcs) that are visible for v (with respect to L). We define the visible width at v as*

$$\mathrm{viwidth}(v) := \max\{\mathrm{viwidth}_{up}(v), \mathrm{viwidth}_{down}(v)\}.$$

We define the visible width of the instance to be $\max_{v \in V} \mathrm{viwidth}(v)$.

Definition 6.4. *We call a link set $L' \subseteq L$ k-thin if for every $v \in V$, $|\{\ell \in L' : v \in \mathrm{in}(P_\ell)\}| \leq k$.*

We remark that our definition of thinness slightly differs from the one introduced in [23] (in the context of WTAP) in that we do not count links ending in a vertex v.

Lemma 6.5. *Assume that $(T = (V, A), L, c)$ has visible width at most k. Let $F \subseteq L$ be a shadow-minimal (meaning that no link can be replaced by a strict shadow without destroying feasibility) solution to the instance. Then F is $2k$-thin.*

Proof. As F is shadow-minimal, we have $\ell = s(\ell)$ and $P_\ell = \overline{P}_\ell$ for every $\ell \in F$. Let $v \in V$ and let $F' := \{\ell \in F : v \in \mathrm{in}(P_\ell)\}$. We need to show that $|F'| \leq 2k$. For each $\ell \in F'$, let $w_\ell \in U_v \setminus \{v\}$ be an endpoint of ℓ (this endpoint is unique unless ℓ is a v-cross-link, in which case we may select either endpoint). Let $a_\ell \in A(P_\ell)$ be the arc incident to w_ℓ. Note that $a_\ell \in A_v$. Let $F'_{up} := \{\ell \in F' : a_\ell \in A_{up}\}$ and let $F'_{down} := \{\ell \in F' : a_\ell \in A_{down}\}$. We show that $|F'_{up}| \leq k$ and $|F'_{down}| \leq k$, which implies the desired statement. We only show $|F'_{up}| \leq k$, $|F'_{down}| \leq k$ can be

derived analogously. We observe that by shadow-minimality of F, we must have $a_\ell \in \overrightarrow{\mathrm{cov}}(\ell)$ for every $\ell \in F'$. In particular, ℓ witnesses that a_ℓ is visible for v. In fact, shadow-minimality allows us to derive an even stronger statement: we must have $a_\ell \in \overrightarrow{\mathrm{cov}}(\ell) \setminus \bigcup_{\ell' \in F \setminus \{\ell\}} \overrightarrow{\mathrm{cov}}(\ell')$. In particular, the arcs $(a_\ell)_{\ell \in F'_{up}}$ are pairwise distinct. We further claim that they form an ancestor-free arc set. As viwidth$(v) \leq k$, this implies $|F'_{up}| \leq k$. Assume towards a contradiction that there were two links $\ell = (u, x), \ell' = (u', x') \in F'_{up}$ such that a_ℓ appears on the path $P_{y'r}$ in T from the head y' of $a_{\ell'} =: (x', y')$ to the root r. As $a_\ell \in A_v$, a_ℓ appears on the y'-v-subpath $P_{y'v}$ of $P_{y'r}$. As y' is the parent of the head x' of ℓ' and $v \in \mathrm{in}(P_{\ell'})$, $P_{y'v}$ is a subpath of $P_{\ell'}$ and as v is an ancestor of y', ℓ' covers every up-arc on that path, including a_ℓ. But this contradicts the fact that $a_\ell \in \overrightarrow{\mathrm{cov}}(\ell) \setminus \bigcup_{\ell'' \in F \setminus \{\ell\}} \overrightarrow{\mathrm{cov}}(\ell'')$. ∎

We remark that there always exists a shadow-minimal optimum solution because we can iteratively replace links in an optimum solution by strict shadows without increasing the cost until the solution is shadow-minimal.

Lemma 6.6. *Let $N \in \mathbb{N}$ be a constant. Given a rooted WDTAP instance (T, L, c, r), we can, in polynomial time, find a cheapest N-thin solution, or decide that the instance is infeasible.*

Proof. Let (T, L, c, r) be a rooted instance of WDTAP. Recall that for $v \in V$, $T_v = (U_v, A_v)$ is the subtree rooted at v. We define the following three links sets for $v \in V$:

- L_v is the set of links that have at least one endpoint in $U_v \setminus \{v\}$. Note that any link that covers an arc $a \in A_v$ must be contained in L_v.
- L_v^{out} consists of all links with one endpoint in $U_v \setminus \{v\}$ and the other endpoint in $V \setminus U_v$. Note that for every $\ell \in L_v^{out}$, $v \in \mathrm{in}(P_\ell)$.
- L_v^{cross} consists of all links ℓ with both endpoints in $U_v \setminus \{v\}$ and $\mathrm{apex}(\ell) = v$. Note that for every $\ell \in L_v^{cross}$, $v \in \mathrm{in}(P_\ell)$.

We further point out that if $\ell \in L$ and $v \in \mathrm{in}(P_\ell)$, then $\ell \in L_v^{out} \cup L_v^{cross}$.

Let $v \in V$ and $Y \subseteq L_v^{out}$. We call a link set $F \subseteq L_v$ *feasible* for (v, Y) if F is N-thin, $F \cap L_v^{out} = Y$ and F covers every arc in A_v. We define $c(v, Y)$ to be the minimum cost of a feasible link set for (v, Y), or ∞, if no such link set exists.

We will use dynamic programming to, for every $v \in V$ and $Y \subseteq L_v^{out}$ with $|Y| \leq N$, compute $c(v, Y)$, as well as a feasible link set $F^*(v, Y)$ for (v, Y) with $c(F^*(v, Y)) = c(v, Y)$, or $F^*(v, Y) = \emptyset$, if $c(v, Y) = \infty$. We remark that if the instance admits a feasible solution, then by shadow-completeness, we, for every arc $a = (u, w)$, have a link $\ell = (w, u)$ just covering a, and we can always use them to complete Y to an N-thin solution. We note that $L_r = L$ (assuming that we do not have links of the form (v, v) that do not cover any arc) and $L_r^{out} = \emptyset$, so $F^*(r, \emptyset)$ yields a cheapest N-thin solution to the instance, or $c(r, \emptyset) = \infty$ and $F^*(r, \emptyset) = \emptyset$ if no such solution exists.

As N is a constant, there is only a polynomial number of pairs (v, Y) that we consider. We traverse the pairs in order of non-increasing distance of v to the root, which ensures that when considering a pair (v, Y), all pairs (v', Y') with $v' \in U_v \setminus \{v\}$ have already been processed. Hence, it remains to show how to, in polynomial time, compute $c(v, Y)$ and $F^*(v, Y)$, assuming that we have already computed $c(v', Y')$ and $F^*(v', Y')$ for all pairs (v', Y') with $v' \in U_v \setminus \{v\}$.

If v is a leaf of T, then $L_v = L_v^{out} = \emptyset$ and $A_v = \emptyset$ and we have $c(v, \emptyset) = 0$ and $F^*(v, \emptyset) = \emptyset$. Next, assume that v is not a leaf of T. Let $Y \subseteq L_v^{out}$ such that $|Y| \leq N$. For $Z \subseteq L_v^{cross}$ with $|Y| + |Z| \leq N$, we say that a link set $F \subseteq L_v$ is *feasible* for (v, Y, Z) if it is feasible for (v, Y) and $F \cap L_v^{cross} = Z$. We denote the minimum cost of a link set that is feasible for (v, Y, Z) by $c(v, Y, Z)$ and let $c(v, Y, Z) = \infty$ if no such link set exists. In addition to the values $c(v, Y, Z)$, we will compute link sets $F^*(v, Y, Z)$ such that $F^*(v, Y, Z)$ is a feasible link set for (v, Y, Z) with $c(F^*(v, Y, Z)) = c(v, Y, Z)$, if exists, and $F^*(v, Y, Z) = \emptyset$ if $c(v, Y, Z) = \infty$. We have

$$c(v, Y) = \min\{c(v, Y, Z) \colon Z \subseteq L_v^{cross}, |Y| + |Z| \leq N\}$$

because if $F \subseteq L_v$ is feasible for (v, Y, Z), then it is also feasible for (v, Y), and conversely, if F is feasible for (v, Y), then F is N-thin, so $N \geq |F \cap L_v^{out}| + |F \cap L_v^{cross}| = |Y| + |F \cap L_v^{cross}|$, and hence, F is feasible for $(v, Y, F \cap L_v^{cross})$. Moreover, if Z attains the above minimum, then we can set $F^*(v, Y) = F^*(v, Y, Z)$. As there is only a polynomial number of sets $Z \subseteq L_v^{cross}$ with $|Y| + |Z| \leq N$, it suffices to show how to, for a fixed choice of Z, compute $c(v, Y, Z)$ and $F^*(v, Y, Z)$.

Let $v_1, \ldots, v_k$ be the children of v in T (recall that v is not a leaf). For $i \in \{0, \ldots, k\}$, let L_v^i be the set of links with at least one endpoint in $U_i := \bigcup_{j=1}^{i} U_{v_i}$, i.e., $L_v^0 = \emptyset$ and $L_v^k = L_v$. We call a link set $F \subseteq L_v^i$ *feasible* for (v, Y, Z, i) if F is N-thin, $F \cap L_v^{out} = Y \cap L_v^i$, $F \cap L_v^{cross} = Z \cap L_v^i$, and F covers every arc in $A_i := \bigcup_{j=1}^{i} A_{v_j} \cup \{a_{v_j}\}$. (Recall that a_{v_j} is the arc connecting v_j to its parent v.) We define $c(v, Y, Z, i)$ to be the minimum cost of a feasible link set for (v, Y, Z, i), or ∞,

if no such link set exists. We will compute the values $c(v, Y, Z, i)$ for $i = 0, \dots, k$, and, whenever $c(v, Y, Z, i) \neq \infty$, we will compute a link set $F^*(v, Y, Z, i)$ attaining $c(v, Y, Z, i)$; otherwise, we will set $F^*(v, Y, Z, i) = \emptyset$.

Note that $c(v, Y, Z) = c(v, Y, Z, k)$ and that $F^*(v, Y, Z, k)$ is a feasible choice for $F^*(v, Y, Z)$. Hence, it remains to explain how to determine the values $c(v, Y, Z, i)$ and $F^*(v, Y, Z, i)$ in polynomial time.

For $i \in \{1, \dots, k\}$, let ℓ_i^* be the link with endpoints v_i and v that covers a_{v_i}, i.e., $\ell_i^* = (v, v_i)$ if $a_{v_i} = (v_i, v)$ and vice versa. Note that $\ell_i^* \in L$ by shadow-completeness and because there exists a link in L covering a_{v_i}; otherwise, the instance is infeasible and we can return this information. Moreover, let $\mathcal{Y}_i$ be the collection of all sets $Y' \subseteq L_{v_i}^{out}$ such that $|Y'| \leq N$ and $Y' \cap (L_v^{out} \cup L_v^{cross}) = (Y \cup Z) \cap L_{v_i}^{out}$.

Claim 6.7. *We have $c(v, Y, Z, 0) = 0$. For $i \in \{1, \dots, k\}$,*

$$c(v, Y, Z, i) = c(v, Y, Z, i-1) + c(Y \cap (L_v^i \setminus L_v^{i-1})) + c(Z \cap (L_v^i \setminus L_v^{i-1})) \\ + \min\{c(v_i, Y') - c(Y' \cap (Y \cup Z)) + \chi[a_{v_i} \text{ not covered by } Y' \cup Y \cup Z] \cdot c(\ell_i^*) \colon Y' \in \mathcal{Y}_i\},$$

where $\chi[a_{v_i}$ is not covered by $Y' \cup Y \cup Z]$ is 1 if a_{v_i} is not covered by $Y' \cup Y \cup Z$, and 0 otherwise.

Proof of claim. As $L_v^0 = \emptyset$, $c(v, Y, Z, 0) = 0$. Next, let $i \in \{1, \dots, k\}$. We first prove that every set $Y' \in \mathcal{Y}_i$ for which the right hand side is finite yields a valid upper bound on $c(v, Y, Z, i)$. To this end, assume that $c(v, Y, Z, i-1) < \infty$ and let $F' := F^*(v, Y, Z, i-1)$ attain this value. Let further $Y' \in \mathcal{Y}_i$ such that $c(v_i, Y') < \infty$ and let $F_i := F^*(v_i, Y')$. Let $F := F' \cup F_i \cup (Y \cup Z) \cap L_v^i$, if a_{v_i} is covered by $Y' \cup Y \cup Z$, and let $F := F' \cup F_i \cup (Y \cup Z) \cap L_v^i \cup \{\ell_i^*\}$ otherwise. Then

$$F \cap L_v^{i-1} = F' \text{ and } F \cap L_{v_i} = F_i \text{ and } F \cap (L_v^{out} \cup L_v^{cross}) = (Y \cup Z) \cap L_v^i \tag{6.1}$$

by construction and by definition of $\mathcal{Y}_i$. Then F covers every arc in A_i because F' covers every arc in A_{i-1}, F_i covers every arc in A_{v_i}, and we also made sure that a_{v_i} is covered. We further have $F \subseteq L_v^i$ by construction. To see that F is N-thin, we note that for every vertex $v' \in U_{i-1}$, there are at most N links in F with $v' \in \text{in}(P_\ell)$ because F' is N-thin, by (6.1) and because if $v' \in \text{in}(P_\ell)$ for $\ell \in F$, then $\ell \in L_v^{i-1}$. Similarly, for every vertex $v' \in U_{v_i}$, there are at most N links in F with $v' \in \text{in}(P_\ell)$ by (6.1) and because F_i is N-thin. There are at most N links in F with $v \in \text{in}(P_\ell)$ by (6.1) and because $|Y| + |Z| \leq N$. For $w \in U_v \setminus (U_i \cup \{v\})$, there are at most N links in F with $w \in \text{in}(P_\ell)$ because $F \subseteq L_v^i$ and $|Z| \leq N$. Finally, for $w \in V \setminus U_v$, there are at most N links in F with $w \in \text{in}(P_\ell)$ because $|Y| \leq N$.

It remains to show that the expression on the right-hand side that we evaluate yields an upper bound on $c(F)$. If we include the link ℓ_i^*, then we add its cost. The cost of every link in $L_v^{i-1} \cap F = F'$ is added (via the term $c(v, Y, Z, i-1)$). The cost of every link in $F_i \setminus (Y \cup Z)$ is added via the term $c(v_i, Y') - c(Y' \cap (Y \cup Z))$ because

$$F_i \cap (Y \cup Z) = (F_i \cap L_{v_i}^{out}) \cap (Y \cup Z) = Y' \cap (Y \cup Z).$$

Finally, the cost of every link in $(Y \cup Z) \cap (L_v^i \setminus L_v^{i-1})$ is added.

Next, we show that if $c(v, Y, Z, i)$ is finite, there exists a set $Y' \in \mathcal{Y}_i$ for which the value of the right-hand side is at most $c(v, Y, Z, i)$. To this end, let $F \subseteq L_v^i$ be feasible for (v, Y, Z, i) with $c(F) = c(v, Y, Z, i)$. Define $F' := F \cap L_v^{i-1}$ and $F_i := F \cap L_{v_i}$, and let $Y' := F \cap L_{v_i}^{out}$. Then F' is feasible for $(v, Y, Z, i-1)$ and F_i is feasible for (v, Y'). Moreover, $|Y'| \leq N$ (as F is N-thin) and

$$Y' \cap (L_v^{out} \cup L_v^{cross}) = F \cap L_{v_i}^{out} \cap (L_v^{out} \cup L_v^{cross}) = L_{v_i}^{out} \cap (F \cap (L_v^{out} \cup L_v^{cross})) \\ = L_{v_i}^{out} \cap (Y \cup Z) \cap L_v^i = L_{v_i}^{out} \cap (Y \cup Z),$$

so $Y' \in \mathcal{Y}_i$. It remains to show that the cost term that we get on the right-hand side when choosing Y' is at most $c(F)$. To this end, we have

$$c(v, Y, Z, i-1) \leq c(F') \text{ and } c(v_i, Y') - c(Y' \cap (Y \cup Z)) \leq c(F_i) - c(F_i \cap (Y \cup Z)) = c(F_i \setminus (Y \cup Z))$$

because $Y' \cap (Y \cup Z) = (F \cap L_{v_i}^{out}) \cap (Y \cup Z) = F_i \cap (Y \cup Z)$ and because F' and F_i are feasible for $(v, Y, Z, i-1)$ and (v_i, Y'), respectively. We note that the subsets F', $F_i \setminus (Y \cup Z)$, $Y \cap (L_v^i \setminus L_v^{i-1})$ and $Z \cap (L_v^i \setminus L_v^{i-1})$ are pairwise distinct because $F' \cap F_i \subseteq Z$. Finally, we observe that none of the previous subsets can contain the link ℓ_i^* and that if a_{v_i} is not covered by $Y' \cup Y \cup Z$, then a_{v_i} can only be covered by ℓ_i^*, so $\ell_i^* \in F$. This is because every link covering a_{v_i} must have one endpoint in U_{v_i} and its other endpoint in $V \setminus U_{v_i}$ and unless the endpoints are v and v_i (i.e., $\ell = \ell_i^*$), we have $\ell \in Y' \cup Y \cup Z$. □

Using the claim, we can compute all of the values $c(v, Y, Z, i)$ in polynomial time. In the proof of the claim, we have further seen how to compute $F^*(v, Y, Z, i)$ attaining $c(v, Y, Z, i)$ in polynomial time. This concludes the proof. ∎

Combining the results of Lemmas 6.5 and 6.6, we conclude that WDTAP instances of constant visible width can be solved exactly in polynomial time.

COROLLARY 6.8. *If (T, L, c, r) is a rooted instance of WDTAP with visible width at most k, then we can, in polynomial time, find an optimal solution, or decide that the instance is infeasible.*

Proof. For a feasible visibly k-wide instance, there exists an optimal solution that is at most $2k$-thin. Hence, we can run the dynamic programming algorithm above with $N = 2k$ to find the optimal solution for this instance, or decide that it is infeasible. ∎

7 The partial separation framework This section describes the high level framework of our algorithm, which is to implement a partial separation oracle for a certain LP formulation we call the visibly k-wide modification LP. We then show how this partial separation oracle implies an algorithm for the WDTAP problem.

7.1 Splitting links First, we formalize the "link splitting" operation, which will be used throughout the paper and in particular will allow us to define the visibly k-wide modification LP.

Fix a (rooted) WDTAP instance (T, L, c, r).

DEFINITION 7.1. *A splitting of the link set L is a function $\sigma\colon L \to 2^L$ mapping $\ell \in L$ to a set of shadows $\ell_1, \dots, \ell_t$ of ℓ such that (the arc sets of) $P_{\ell_1}, \dots, P_{\ell_t}$ form a partition of (the arc set of) P_ℓ. The support of the splitting is* $\operatorname{supp}(\sigma) := \{\ell \in L\colon \exists \ell' \in L\colon \ell \in \sigma(\ell')\}$.

Next, we define how to apply a splitting to a solution to (2.1) to generate a new feasible solution of (2.1).

DEFINITION 7.2. *Let x be a feasible solution to* (2.1) *and let σ be a splitting of L. We let the solution $x' = \operatorname{split}(x, \sigma)$ to* (2.1) *that we obtain from x by applying σ be defined by $x'_{\ell'} := \sum_{\ell \in L\colon \ell' \in \sigma(\ell)} x_\ell$.*

The following proposition shows that splitting links can only reduce the visible up- or down-width of any vertex. To state it, we introduce the following notation.

DEFINITION 7.3. *Let x be a solution to* (2.1)*. The support* $\operatorname{supp}(x)$ *of x consists of all links ℓ with $x_\ell > 0$.*

PROPOSITION 7.4. *Let x be a solution to* (2.1)*, let σ be a splitting of L and let $x' := \operatorname{split}(x, \sigma)$. Then for every vertex v, the visible up-width (visible down-width) of v with respect to* $\operatorname{supp}(x')$ *is at most the visible up-width (visible down-width) of v with respect to* $\operatorname{supp}(x)$.

Proof. It suffices to show that every arc that is visible for a vertex v with respect to $\operatorname{supp}(x')$ is also visible for that vertex with respect to $\operatorname{supp}(x)$. Let a be an arc that is visible for v with respect to $\operatorname{supp}(x')$. Then there is $\ell' \in \operatorname{supp}(x')$ such that $a \in \overrightarrow{\operatorname{cov}}(\ell')$ and $v \in \operatorname{in}(\overline{P}_{\ell'})$. As $\ell' \in \operatorname{supp}(x')$, there is $\ell \in \operatorname{supp}(x)$ such that $\ell' \in \sigma(\ell)$. Then ℓ' is a shadow of ℓ, so $a \in \overrightarrow{\operatorname{cov}}(\ell)$ and $v \in \operatorname{in}(\overline{P}_\ell)$. Hence, a is also visible for v with respect to $\operatorname{supp}(x)$. ∎

The following proposition shows that the coverage of all tree arcs is preserved by the splitting operation.

PROPOSITION 7.5. *Let x be a feasible solution to* (2.1)*, let σ be a splitting of L and let $x' := \operatorname{split}(x, \sigma)$. For $a \in A$, we have*

- $x'(\{\ell \in L\colon a \in \overrightarrow{\operatorname{cov}}(\ell)\}) = x(\{\ell \in L\colon a \in \overrightarrow{\operatorname{cov}}(\ell)\})$,
- $x'(\{\ell \in L\colon a \in \overleftarrow{\operatorname{cov}}(\ell)\}) = x(\{\ell \in L\colon a \in \overleftarrow{\operatorname{cov}}(\ell)\})$ *and*
- $x'(\{\ell \in L\colon a \in \operatorname{cov}(\ell)\}) = x(\{\ell \in L\colon a \in \operatorname{cov}(\ell)\})$.

Proof. Let $a \in A$. We only show the first equality, the other ones can be derived analogously. For any link $\ell \in L$ with $a \in \overrightarrow{\operatorname{cov}}(\ell)$, there exists a unique link $\ell_a \in \sigma(\ell)$ with $a \in \overrightarrow{\operatorname{cov}}(\ell_a)$. Conversely, if $a \in \overrightarrow{\operatorname{cov}}(\ell')$ and $\ell' \in \sigma(\ell)$, then $a \in \overrightarrow{\operatorname{cov}}(\ell)$ (and $\ell' = \ell_a$). This implies

$$\begin{aligned} x'(\{\ell \in L\colon a \in \overrightarrow{\operatorname{cov}}(\ell)\}) &= \sum_{\ell' \in L\colon a \in \overrightarrow{\operatorname{cov}}(\ell')} \sum_{\ell \in L\colon \ell' \in \sigma(\ell)} x_\ell = \sum_{\ell \in L\colon a \in \overrightarrow{\operatorname{cov}}(\ell)} \underbrace{|\{\ell' \in \sigma(\ell)\colon a \in \overrightarrow{\operatorname{cov}}(\ell')\}|}_{=1} \cdot x_\ell \\ &= x(\{\ell \in L\colon a \in \overrightarrow{\operatorname{cov}}(\ell)\}). \end{aligned}$$

∎

The following proposition simply counts the additional cost incurred by splitting.

Proposition 7.6. *Let x be a feasible solution to* (2.1)*, let σ be a splitting of L and let $x' \coloneqq \operatorname{split}(x, \sigma)$. Then x' is a feasible solution to* (2.1) *of cost*

$$c(x') = \sum_{\ell \in L} \left(\sum_{\ell' \in \sigma(\ell)} c(\ell') \right) \cdot x_\ell.$$

Proof. Feasibility of x' follows from Proposition 7.5. For the cost, we calculate

$$\sum_{\ell \in L} c(\ell) \cdot x'_\ell = \sum_{\ell \in L} \sum_{\ell' \in \sigma(\ell)} c(\ell') \cdot x_\ell.$$

□

We will often apply splittings sequentially, which is captured by the following definition.

Definition 7.7. *Let σ and σ' be two splitting of L. Then concatenation $\sigma' \circ \sigma$ of the two splittings is defined via $(\sigma' \circ \sigma)(\ell) = \bigcup_{\ell' \in \sigma(\ell)} \sigma'(\ell')$.*

Note that the concatenation of two splittings of L is again a splitting of L. We further observe the following.

Proposition 7.8. *Let x be a feasible solution to* (2.1) *and let σ and σ' be two splitting of L. Then* $\operatorname{split}(x, \sigma' \circ \sigma) = \operatorname{split}(\operatorname{split}(x, \sigma), \sigma')$.

Proof. Let $x' \coloneqq \operatorname{split}(x, \sigma)$ and $x'' \coloneqq \operatorname{split}(\operatorname{split}(x, \sigma), \sigma')$. For $\ell'' \in L$, we have

$$x''_{\ell''} = \sum_{\ell' \in L \colon \ell'' \in \sigma'(\ell')} x'_{\ell'} = \sum_{\ell' \in L \colon \ell'' \in \sigma'(\ell')} \sum_{\ell \in L \colon \ell' \in \sigma(\ell)} x_\ell = \sum_{\ell \in L \colon \ell'' \in (\sigma' \circ \sigma)(\ell)} x_\ell.$$

For the last equality, we used that for $\ell \in L$, $\sigma(\ell)$ consists of shadows of ℓ with pairwise disjoint undirected coverages. In particular, we can have $\ell'' \in \sigma'(\ell')$ for at most one $\ell' \in \sigma(\ell)$ because ℓ'' has to be a shadow of ℓ'. □

The following proposition helps us to bound the cost increase incurred by splittings.

Proposition 7.9. *Let $\Delta > 1$ and assume that $c \colon L \to [1, \Delta]$. Let x be a solution to* (2.1) *and let σ be a splitting of L. Let $x' \coloneqq \operatorname{split}(x, \sigma)$. Then*

$$c(x') \leq c(x) + \sum_{\ell \in L} (|\sigma(\ell)| - 1) \cdot c(\ell) \cdot x_\ell \leq c(x) + \Delta \cdot \sum_{\ell \in L} (|\sigma(\ell)| - 1) \cdot x_\ell.$$

Proof. We have

$$\begin{aligned} c(x') &= \sum_{\ell \in L} \sum_{\ell' \in \sigma(\ell)} c(\ell') \cdot x_\ell \leq \sum_{\ell \in L} |\sigma(\ell)| \cdot c(\ell) \cdot x_\ell \\ &= c(x) + \sum_{\ell \in L} (|\sigma(\ell)| - 1) \cdot c(\ell) \cdot x_\ell \leq c(x) + \Delta \cdot \sum_{\ell \in L} (|\sigma(\ell)| - 1) \cdot x_\ell, \end{aligned}$$

where the first inequality follows from the fact that $c(\ell') \leq c(\ell)$ whenever ℓ' is a shadow of ℓ, and the second inequality follows from $|\sigma(\ell)| \geq 1$ and $c(\ell) \leq \Delta$. □

7.2 The visibly k-wide modification LP Using splittings, we will introduce a new type of valid inequality for the integer hull of (2.1). To define it, we need to consider subinstances that arise by contracting certain arcs. Given an arc set A^*, we denote by T/A^* the tree that arises from T by contracting the arcs in A^*. For a link set L^*, L^*/A^* denotes the link set arising from this contraction. For a function $f : A \to B$, and $C \subseteq A$, we use the notation $f \restriction_C$ to denote the restriction of f to the domain C.

Lemma 7.10. *Let σ be any splitting of the link set and let $A' \subseteq A$. Then*

$$\sum_{\ell \in L} \left(\sum_{\ell' \in \sigma(\ell)} c(\ell') \right) \cdot x_\ell \geq c(OPT(T/A', \operatorname{supp}(\sigma)/A', c \restriction_{\operatorname{supp}(\sigma)})) \tag{7.1}$$

is a valid constraint for the integer hull of (2.1), where $OPT(T/A', \operatorname{supp}(\sigma)/A', c \restriction_{\operatorname{supp}(\sigma)})$ denotes an optimum solution to the WDTAP instance $(T/A', \operatorname{supp}(\sigma)/A', c \restriction_{\operatorname{supp}(\sigma)})$.

Proof. Let x be an integral solution to (2.1) and let $x' \coloneqq \text{split}(x, \sigma)$. Then x' is an integral solution to (2.1) with $\text{supp}(x') \subseteq \text{supp}(\sigma)$. In particular, $\text{supp}(x')/A'$ is a feasible solution to $(T/A', \text{supp}(\sigma)/A', c \restriction_{\text{supp}(\sigma)})$ of cost at most $c(x') = \sum_{\ell \in L} \left(\sum_{\ell' \in \sigma(\ell)} c(\ell')\right) \cdot x_\ell$ by Proposition 7.6. ∎

Definition 7.11. *A visibly k-wide modification is a pair (σ, A'), where σ is a splitting of the link set and $A' \subseteq A$, such that $(T/A', \text{supp}(\sigma)/A', r)$ has visible width at most k. We call the corresponding contraint* (7.1) *a visibly k-wide modification inequality.*

Our approach will be to observe certain solutions to the linear program (2.1) and to obtain an integral solution of relatively low cost, or to find a visibly k-wide modification inequality violated by the current solution to add to the constraints of (2.1).

7.3 Proof of Theorem 1.1 The main technical theorem of this paper guarantees the existence of a partial separation oracle for the visibly k-wide-modification LP. This theorem is stated below, and in this subsection we will show how to use it to prove Theorem 1.1.

Theorem 7.12. *Let $\bar{\varepsilon}, \Delta > 0$. We can compute a constant $k(\bar{\varepsilon}, \Delta)$ with the following property: Given a rooted instance $(\bar{T}, \bar{L}, \bar{c}, \bar{r})$ of WDTAP with cost ratio at most Δ and a feasible solution $\bar{x}$ to* (2.1)*, we can, in polynomial time, either find a solution $S \subseteq \bar{L}$ with $\bar{c}(S) \leq (1.75 + \bar{\varepsilon}) \cdot \bar{c}(\bar{x})$, or find a visibly $k(\bar{\varepsilon}, \Delta)$-wide modification inequality that is violated by $\bar{x}$.*

Assuming Theorem 7.12, we are now ready to prove Theorem 1.1, which we restate for convenience.

Theorem 1.1. *Let $\Delta \geq 1$ and let $\varepsilon > 0$. There exists a polynomial-time $(1.75+\varepsilon)$-approximation algorithm for WDTAP, restricted to instances with cost ratio at most Δ.*

Proof. Let $\Delta \geq 1$ and let $\varepsilon > 0$. We may assume that the constants ε and Δ are rational numbers because we can replace them with rational constants ε' and Δ' with $0 < \varepsilon' < \varepsilon$ and $\Delta < \Delta'$ otherwise. Fix a rooted WDTAP instance (T, L, c, r) with cost ratio at most Δ. We can check in polynomial time if (T, L, c, r) is feasible by checking if each tree arc is covered by at least one link in L. Hence, we will assume that (T, L, c, r) is feasible in the following. By re-scaling the costs, we may assume $c \colon L \to [1, \Delta]$. Then the cost of an optimal solution OPT satisfies $1 \leq c(OPT) \leq \Delta|L| \leq \Delta n^2$, where n is the number of vertices of T. Let $\bar{\varepsilon} \coloneqq \min\{1, \frac{\varepsilon}{10}\}$, let $k \coloneqq k(\bar{\varepsilon}, \Delta)$ and let $M \coloneqq \lceil \log_{1+\bar{\varepsilon}} n^2\Delta \rceil$. We will use binary search on the interval $[1, (1+\bar{\varepsilon})^M]$. Note that the runtime of the algorithm will depend on ε and Δ.

In the following, we will describe a subroutine that, given a rational number $c^* \in [1, (1+\bar{\varepsilon})^M]$, in polynomial time (in the encoding lengths of (T, L, c, r), Δ, ε and c^*) either returns a solution F to (T, L, c, r) with $c(F) \leq (1.75+\bar{\varepsilon}) \cdot (1+\bar{\varepsilon}) \cdot c^*$, or decides that $c^* < c(OPT)$. The subroutine is defined as follows. Given c^*, we apply the ellipsoid method to (try to) find a feasible point x in the polyhedron P given by the constraints in (2.1), all visibly k-wide modification inequalities, and $c(x) \leq (1+\bar{\varepsilon}) \cdot c^*$. Note that the encoding length of every constraint is polynomially bounded in the encoding lengths of (T, L, c, r), ε and c^*. Moreover, $P \subseteq [0, (1+\bar{\varepsilon}) \cdot c^*]^L$ since $c(\ell) \geq 1$ for every $\ell \in L$. If $c^* \geq c(OPT)$, then we further have $OPT + [0, \frac{\bar{\varepsilon}}{\Delta \cdot |L|} \cdot c^*]^L \subseteq P$, where we interpret OPT as a vector in $\{0,1\}^L$. Finally, we can separate all constraints in (2.1), as well as the constraint $c(x) \leq (1+\bar{\varepsilon}) \cdot c^*$, in polynomial time. To separate the visibly k-wide modification inequalities, we will use Theorem 7.12.

More precisely, in each iteration of the ellipsoid method, given $y \in \mathbb{Q}^L$, we do the following: If y violates any of the constraints of (2.1) or $c(y) > (1+\bar{\varepsilon}) \cdot c^*$, we return the corrresponding violated constraint. Otherwise, we apply Theorem 7.12 to either find a WDTAP solution F with $c(F) \leq (1.75+\bar{\varepsilon}) \cdot c(y) \leq (1.75+\bar{\varepsilon}) \cdot (1+\bar{\varepsilon}) \cdot c^*$, or a violated visibly k-wide modification inequality. In the first case, we return F and stop; in the second case, we continue the ellipsoid method. After a polynomial number of iterations, we have either found a WDTAP solution F with $c(F) \leq (1.75+\bar{\varepsilon}) \cdot (1+\bar{\varepsilon}) \cdot c^*$, or the volume of the ellipsoid is small enough, allowing us to deduce that $c^* < c(OPT)$.

Throughout the binary search, we maintain an interval $[(1+\bar{\varepsilon})^a, (1+\bar{\varepsilon})^b]$ such that $c(OPT) \geq (1+\bar{\varepsilon})^a$ and we have a WDTAP solution F with $c(F) \leq (1.75+\bar{\varepsilon}) \cdot (1+\bar{\varepsilon})^{b+1}$. Continually checking the point $c^* = (1+\bar{\varepsilon})^{\lfloor \frac{a+b}{2} \rfloor}$, we obtain an interval of the form $[(1+\bar{\varepsilon})^t, (1+\bar{\varepsilon})^{t+1}]$. In this case, we are guaranteed an integral solution F of cost at most

$$\begin{aligned} c(F) &\leq (1.75+\bar{\varepsilon}) \cdot (1+\bar{\varepsilon})^{t+2} \leq (1.75+\bar{\varepsilon}) \cdot (1+\bar{\varepsilon})^2 \cdot c(OPT) \\ &\leq (1.75 + 4.5\bar{\varepsilon} + 3.75\bar{\varepsilon}^2 + \bar{\varepsilon}^3) \cdot c(OPT) \leq (1.75+\varepsilon) \cdot c(OPT). \end{aligned}$$

∎

We remark that the partial separation framework has already been used in [1, 13]. The remainder of the main part of this paper is dedicated to proving Theorem 7.12.

8 Proving the weakened dream theorem In this section, we prove the weakened dream theorem (Theorem 4.2). To this end, we will first introduce some notation that allows us to state Theorem 8.3, a slightly more formal version of Theorem 4.2.

Definition 8.1. *Let (T, L, c, r) be a rooted instance of WDTAP, let x be a solution to* (2.1) *and let $\alpha \geq 0$. We call an arc a α-covered if $x(\{\ell \colon a \in \overrightarrow{\mathrm{cov}}(\ell)\}) \geq \alpha$ and α-heavy if $x(\{\ell \colon a \in \overleftarrow{\mathrm{cov}}(\ell)\}) \geq \alpha$. We call a link ℓ α-heavily involved if there is an α-heavy arc a with $a \in \overleftarrow{\mathrm{cov}}(\ell)$.*

Let $\varepsilon \in (0, 1)$ and let $\Delta > 0$. We define the following constants:

- $\gamma \coloneqq \frac{\varepsilon}{2\Delta}$ is used to define whether an arc a is lightly covered, allowing us to cheaply split links ℓ with $a \in \mathrm{cov}(\ell)$.
- $\zeta_1 \coloneqq \frac{2}{\varepsilon}$ is our threshold for an arc to be "heavily covered in the right direction", allowing us to contract it.
- $\zeta_2 \coloneqq \frac{6 \cdot \zeta_1 \cdot \Delta}{\varepsilon \cdot (1-\varepsilon)}$ is our our threshold for an arc to be "heavily covered in the wrong direction".
- $k \coloneqq (1 + \gamma^{-1}) \cdot \zeta_2$ is our bound on the visible width of certain instances that we will target.

Observe that our choice of constants satisfies the following inequalities. In fact, all but (8.3) are actually equalities, but we do not need that fact.

$$2 \cdot \Delta \cdot \gamma \leq \varepsilon \tag{8.1}$$

$$\frac{2}{\varepsilon} \leq \zeta_1 \tag{8.2}$$

$$\zeta_1 < \varepsilon \cdot \zeta_2 \tag{8.3}$$

$$3 \cdot \zeta_1 \cdot \Delta \leq \varepsilon \cdot \frac{1}{2} \cdot (1 - \varepsilon) \cdot \zeta_2 \tag{8.4}$$

$$(1 + \gamma^{-1}) \cdot \zeta_2 \leq k \tag{8.5}$$

Lemma 8.2. *Let (T, L, c, r) be a rooted instance of WDTAP, let x be a solution to* (2.1) *and let F' be a feasible solution to the instance we obtain after contracting all ζ_1-covered arcs. Then we can, in polynomial time, compute a solution F of cost $c(F) \leq c(F') + \varepsilon \cdot c(x)$ to the original instance.*

Proof. We show how to, in polynomial time, compute a link set F'' of cost $c(F'') \leq \varepsilon \cdot c(x)$ that covers all ζ_1-covered arcs. Let $(\bar{T} = (\bar{V}, \bar{A}), \bar{L}, \bar{c}, \bar{r})$ arise from (T, L, c, r) by contracting all arcs that are not ζ_1-covered. Then x corresponds to a solution $\bar{x}$ of cost $\bar{c}(\bar{x}) = c(x)$ to (2.1) for $(\bar{T}, \bar{L}, \bar{c})$ with the property that $\bar{x}(\ell \in \bar{L} \colon a \in \overrightarrow{\mathrm{cov}}(\ell) \geq \zeta_1$ for every $a \in \bar{A}$. In particular, $x' \coloneqq \frac{1}{\zeta_1} \cdot \bar{x}$ is a feasible solution to (2.1) for $(\bar{T}, \bar{L}, \bar{c})$ as well. Obtain x'' from x' by splitting every link in $\mathrm{supp}(x')$ that is not an up- or down-link already at its apex. Then

$$\bar{c}(x'') \leq 2 \cdot \bar{c}(x') = \frac{2}{\zeta_1} \cdot c(x) \leq \varepsilon \cdot c(x)$$

by (8.2). Note that $(\bar{T}, \mathrm{supp}(x''), \bar{c}, \bar{r})$ is a willow (choosing $U = \emptyset$), so we can, in polynomial time, compute an optimum solution $\bar{F}$ to $(\bar{T}, \mathrm{supp}(x''), \bar{c})$ of cost at most $\bar{c}(x'') \leq \varepsilon \cdot c(x)$ by Theorem 5.3. The uncontracted links corresponding to $\bar{F}$ yield the desired link set F''. Setting $F = F' \cup F''$ concludes the proof. ∎

In the following, it will be convenient to make the following assumption.

$$\text{There are no } \zeta_1\text{-covered arcs.} \tag{8.6}$$

Lemma 8.2 essentially tells us that we can assume (8.6) at the cost of a cost increase by $\varepsilon \cdot c(x)$.

Before stating Theorem 8.3, the more formal version of Theorem 4.2, we need to introduce the following notation:

- For a vertex $v \in V \setminus \{r\}$, we let a_v be the arc connecting v to its parent.
- For a vertex $v \in V \setminus \{r\}$ such that a_v is an up-arc, we define $\overrightarrow{\mathrm{viwidth}}(v) \coloneqq \mathrm{viwidth}_{up}(v)$, $\overleftarrow{\mathrm{viwidth}}(v) \coloneqq \mathrm{viwidth}_{down}(v)$, $\overrightarrow{A}_v \coloneqq A_v \cap A_{up}$ and $\overleftarrow{A}_v \coloneqq A_v \cap A_{down}$.
- For a vertex v such that a_v is a down-arc, we let $\overrightarrow{\mathrm{viwidth}}(v) \coloneqq \mathrm{viwidth}_{down}(v)$, $\overleftarrow{\mathrm{viwidth}}(v) \coloneqq \mathrm{viwidth}_{up}(v)$, $\overrightarrow{A}_v \coloneqq A_v \cap A_{down}$ and $\overleftarrow{A}_v \coloneqq A_v \cap A_{up}$.

THEOREM 8.3. *Let (T, L, c, r) be an instance of WDTAP with cost ratio at most Δ and let x be a solution to* (2.1) *satisfying* (8.6). *We can, in polynomial time, compute a splitting σ^* of L and a vertex set $W^* \subseteq V$ with the following properties:*

(i) Let $x^* := \mathrm{split}(x, \sigma^*)$. We have $c(x^*) \leq (1+\varepsilon) \cdot c(x)$.

(ii) W^* consists of up- and down-independent vertices with respect to $\mathrm{supp}(x^*)$.

(iii) Let L' arise from $\mathrm{supp}(x^*)$ *by splitting every W^*-cross-link at its apex. With respect to L', we have the following:*

(a) $\overrightarrow{\mathrm{viwidth}}(v) \leq k$ for every $v \in V$.

(b) $\overleftarrow{\mathrm{viwidth}}(r) \leq k$ and we have $\overleftarrow{\mathrm{viwidth}}(v) \leq k$ for every $v \in V \setminus \{r\}$ such that a_v is not ζ_2-heavy (with respect to x^*).

Note that when saying that L' arises from $\mathrm{supp}(x^*)$ by splitting every W^*-cross-link at its apex, we mean the following: There exists a splitting σ of L such that $L' = \mathrm{supp}(\mathrm{split}(x^*, \sigma))$ and such that for every W^*-cross-link ℓ, $\sigma(\ell)$ consists of up- and down-links only, i.e., ℓ has been split at its apex (and potentially at further vertices).

The rest of this section is dedicated to proving Theorem 8.3. Fix a rooted WDTAP instance $(T = (V, A), L, c, r)$ with cost ratio at most Δ. By rescaling the costs, we may assume without loss of generality that $c \colon L \to [1, \Delta]$. To establish Theorem 8.3, we will traverse the tree T bottom-up, starting from the leaves and working our way up towards the root. Whenever we encounter a vertex v of high visible width, we will try to split links with one endpoint in T_v and one endpoint outside T_v, rendering v up- or down-independent. We introduce the following notation, which slightly differs from the one used in Section 4.

DEFINITION 8.4. *Let $v \in V \setminus \{r\}$. We say that a link $\ell = (u, w)$ points into T_v if $w \in U_v \setminus \{v\}$ and $u \notin U_v$. We say that ℓ points out of T_v if $u \in U_v \setminus \{v\}$ and $w \notin U_v$. We denote the set of links pointing into/ out of T_v by $L_v^{\downarrow}$ and $L_v^{\uparrow}$, respectively.*

PROPOSITION 8.5. *If $L_v^{\downarrow} = \emptyset$, then v is up-independent. If $L_v^{\uparrow} = \emptyset$, then v is down-independent.*

Proof. We only prove the first statement, the second one can be derived analogously. Assume $L_v^{\downarrow} = \emptyset$ and let $\ell = (y, z) \in L$. Assume $\overrightarrow{\mathrm{cov}}(\ell) \cap A_v \cap A_{up} \neq \emptyset$. Then $z \in U_v \setminus \{v\}$. As $\ell \notin L_v^{\uparrow} = \emptyset$, $z \in U_v$. Hence, $\overrightarrow{\mathrm{cov}}(\ell) \subseteq \mathrm{cov}(\ell) \subseteq A_v$.□

We are now ready to define when an arc a_v is "light" with respect to a solution to (2.1), allowing us to split all links covering it in the right or in the wrong direction, respectively, without increasing the cost of the LP solution by too much. As outlined in Section 4, we will charge the cost of the splitting to the coverage of visible arcs in the subtree hanging off v. In doing so, it will be convenient to measure the coverage of these arcs with respect to the *original LP solution* x, whilst defining visibility with respect to the support L' of the *split LP solution* x'.

DEFINITION 8.6. *Let $\gamma \in (0, 1)$, let x be a solution to* (2.1) *and let $L' \subseteq L$. Let $v \in V \setminus \{r\}$. We say that a_v is γ-up-light with respect to x and L' if*

$$x(L_v^{\downarrow}) \leq \gamma \cdot x(\{\ell \in L \colon \overrightarrow{\mathrm{cov}}(\ell) \cap A_{up} \cap A_v^{vis}(L') \neq \emptyset\} \setminus L_v^{\downarrow}).$$

We say that a_v is γ-down-light with respect to x and L' if

$$x(L_v^{\uparrow}) \leq \gamma \cdot x(\{\ell \in L \colon \overrightarrow{\mathrm{cov}}(\ell) \cap A_{down} \cap A_v^{vis}(L') \neq \emptyset\} \setminus L_v^{\uparrow}).$$

Note that L' is only used to specify visibility, however, we evaluate x on all of L.

Next, we define the type of splitting operation that we will perform when encountering a light arc.

DEFINITION 8.7. *Let $v \in V$ and let $L' \subseteq L$. The splitting $\sigma_{v,L'}$ is defined as follows. For $\ell = (u, w) \in L'$ with $v \in \mathrm{in}(P_\ell)$, we define $\sigma_{v,L'}(\ell) = \{(u, v), (v, w)\}$. For every other link ℓ, we define $\sigma_{v,L'}(\ell) = \{\ell\}$.*

Recall that $\mathrm{in}(P_\ell)$ denotes the set of inner vertices of the path P_ℓ. The splitting $\sigma_{v,L'}$ splits every link $\ell \in L'$ with $v \in \mathrm{in}(P_\ell)$ into two shadows; one starting and one ending in v.

The next proposition allows us to bound the cost increase incurred by successive splitting operations.

PROPOSITION 8.8. *Let x be a solution to* (2.1), *let $v \in V$ and let $L' \subseteq L$. Let $x' := \mathrm{split}(x, \sigma_{v,L'})$. Then*

$$c(x') \leq c(x) + \Delta \cdot x(L').$$

Proof. This follows from Proposition 7.9 by observing that $|\sigma_{v,L'}(\ell)| \leq 2$ for $\ell \in L'$ and $|\sigma_{v,L'}(\ell)| = 1$ for $\ell \notin L'$. ∎

The link sets that we will choose as L' will be of the form $L_v^{\downarrow}$ and $L_v^{\uparrow}$, respectively. The following lemma tells us that splitting cannot increase the total x-value on these subsets. (It can, however, decrease it to zero if splits are performed at v.)

PROPOSITION 8.9. *Let x be a solution to* (2.1)*, let σ be a splitting of x and let $x' := \mathrm{split}(x, \sigma)$. Let $v \in V \setminus \{r\}$.*

- *We have $x'(L_v^{\downarrow}) \leq x(L_v^{\downarrow})$ and $x'(L_v^{\uparrow}) \leq x(L_v^{\uparrow})$.*
- *If $\sigma = \sigma_{w,L'}$ and $w \neq v$ or $L' \cap L_v^{\downarrow} = \emptyset$, then $x'(L_v^{\downarrow}) = x(L_v^{\downarrow})$.*
- *If $\sigma = \sigma_{w,L'}$ and $w \neq v$ or $L' \cap L_v^{\uparrow} = \emptyset$, then $x'(L_v^{\uparrow}) = x(L_v^{\uparrow})$.*

Proof. We only prove the statements for $L_v^{\downarrow}$, the proof for $L_v^{\uparrow}$ is analogous. Let $\ell \in L$. We make the following two observations:

- If there is $\ell' \in \sigma(\ell) \cap L_v^{\downarrow}$, then $\ell \in L_v^{\downarrow}$ because ℓ' is a shadow of ℓ.
- For $\ell \in L_v^{\downarrow}$, we have $|\sigma(\ell) \cap L_v^{\downarrow}| \leq 1$ because $a_v \in \mathrm{cov}(\ell')$ for every $\ell' \in \sigma(\ell) \cap L_v^{\downarrow}$, but the sets $(\mathrm{cov}(\ell'))_{\ell' \in \sigma(\ell)}$ are pairwise disjoint. (Recall that $\mathrm{cov}(\ell')$ is the arc set of $P_{\ell'}$).

This yields

$$x'(L_v^{\downarrow}) = \sum_{\ell' \in L_v^{\downarrow}} \sum_{\substack{\ell \in L: \\ \ell' \in \sigma(\ell)}} x_\ell = \sum_{\ell' \in L_v^{\downarrow}} \sum_{\substack{\ell \in L_v^{\downarrow}: \\ \ell' \in \sigma(\ell)}} x_\ell = \sum_{\ell \in L_v^{\downarrow}} |\sigma(\ell) \cap L_v^{\downarrow}| \cdot x_\ell \leq x(L_v^{\downarrow}),$$

proving the first statement (for $L_v^{\downarrow}$). We note that if $\sigma = \sigma_{w,L'}$ and $w \neq v$ or $L' \cap L_v^{\downarrow} = \emptyset$, then for every $\ell \in L_v^{\downarrow}$, there is exactly one $\ell' \in \sigma(\ell)$ with $\ell' \in L_v^{\downarrow}$ and we get equality above. ∎

Algorithm 8.1 shows the splitting procedure that we employ in order to prove Theorem 8.3. We traverse the vertices in $V \setminus \{r\}$ from the leaves towards the root. If a_v is γ-up-light, we split all links pointing into T_v at v, if a_v is γ-down-light, we split links pointing out of T_v. Throughout the algorithm, we keep track of the current (split) LP solution x^*, the splitting σ^* with $x^* = \mathrm{split}(x, \sigma^*)$ and the sets W_{up} and W_{down} of vertices v at which links pointing into and out of T_v have been split, respectively. We write σ_{id} to denote the initial *identity splitting* given by $\sigma_{\mathrm{id}}(\ell) = \{\ell\}$ for every $\ell \in L$. Note that $x = \mathrm{split}(x, \sigma_{\mathrm{id}})$. Note that Algorithm 8.1 runs in polynomial time. We will show that the output (σ^*, x^*, W^*) of

Algorithm 8.1 Light link splitting.

Input: solution x to (2.1)
Output: splitting σ^*, $x^* = \mathrm{split}(x, \sigma^*)$, vertex set W^*

```
σ* ← σ_id, x* ← x, W_up ← ∅, , W_down ← ∅
for v ∈ V \ {r} in order of non-increasing distance to r do
    if a_v is γ-up-light (with respect to x and supp(x*)) then
        σ* ← σ_{v,L_v^↓} ∘ σ*, x* ← split(x*, σ_{v,L_v^↓})
        W_up ← W_up ∪ {v}
    end if
    if a_v is γ-down-light (with respect to x and supp(x*)) then
        σ* ← σ_{v,L_v^↑} ∘ σ*, x* ← split(x*, σ_{v,L_v^↑})
        W_down ← W_down ∪ {v}
    end if
end for
return σ*, x*, W* := W_up ∪ W_down ∪ {r}
```

Algorithm 8.1 meets the requirements of Theorem 8.3. Our first goal is to establish Theorem 8.3 *(i)*.

To this end, for $v \in V \setminus \{r\}$, let $x^{*,v}$ denote value of x^* at the beginning of the iteration of the for-loop where v is considered. Let W_{up} and W_{down} denote the values of the respective sets when the algorithm terminates.

Proposition 8.10. *We have*

$$\begin{aligned} c(x^*) \leq \quad & c(x) \\ & + \Delta \cdot \sum_{v \in W_{up}} \gamma \cdot x(\{\ell \in L \colon \overrightarrow{\mathrm{cov}}(\ell) \cap A_{up} \cap A_v^{vis}(\mathrm{supp}(x^{*,v})) \neq \emptyset\} \setminus L_v^{\downarrow}) \\ & + \Delta \cdot \sum_{v \in W_{down}} \gamma \cdot x(\{\ell \in L \colon \overrightarrow{\mathrm{cov}}(\ell) \cap A_{down} \cap A_v^{vis}(\mathrm{supp}(x^{*,v})) \neq \emptyset\} \setminus L_v^{\uparrow}). \end{aligned}$$

Proof. By Propositions 8.8 and 8.9, we have

$$c(x^*) \leq c(x) + \Delta \cdot \left(\sum_{v \in W_{up}} x^{*,v}(L_v^{\downarrow}) + \sum_{v \in W_{down}} x^{*,v}(L_v^{\uparrow}) \right).$$

By Proposition 8.9, we know that $x^{*,v}(L_v^{\downarrow}) \leq x(L_v^{\downarrow})$ and $x^{*,v}(L_v^{\uparrow}) \leq x(L_v^{\uparrow})$ for all $v \in V \setminus \{r\}$. The desired statement, hence, follows from Definition 8.6. ∎

To derive a good bound on the cost increase from Proposition 8.10, we need to make sure that a link ℓ does not appear in too many of the sets $\{\ell \in L \colon \overrightarrow{\mathrm{cov}}(\ell) \cap A_{up} \cap A_v^{vis}(\mathrm{supp}(x^{*,v})) \neq \emptyset\} \setminus L_v^{\downarrow}$ and $\{\ell \in L \colon \overrightarrow{\mathrm{cov}}(\ell) \cap A_{down} \cap A_v^{vis}(\mathrm{supp}(x^{*,v})) \neq \emptyset\} \setminus L_v^{\uparrow}$, respectively. The next lemma takes care of this.

Proposition 8.11. *Let $\ell \in L$.*

- *There is at most one vertex $v \in W_{up}$ such that $\overrightarrow{\mathrm{cov}}(\ell) \cap A_{up} \cap A_v^{vis}(\mathrm{supp}(x^{*,v})) \neq \emptyset$ and $\ell \notin L_v^{\downarrow}$.*
- *There is at most one vertex $v \in W_{down}$ such that $\overrightarrow{\mathrm{cov}}(\ell) \cap A_{down} \cap A_v^{vis}(\mathrm{supp}(x^{*,v})) \neq \emptyset$ and $\ell \notin L_v^{\uparrow}$.*

Proof. We only prove the statement for $v \in W_{up}$, as the other one can be derived analogously. If there is no such vertex $v \in W_{up}$, there is nothing to show. Next, assume that there is at least one such vertex and let v_0 be the first one considered by the algorithm. Let $\ell = (y, z)$. As ℓ covers an arc in $A_{up} \cap A_v$, we have $z \in U_v \setminus \{v\}$. As $\ell \notin L_v^{\downarrow}$, $y \in U_v$. Hence, $\overrightarrow{\mathrm{cov}}(\ell) \subseteq \mathrm{cov}(\ell) \subseteq A_v$. As $v_0 \in W_{up}$, we know that every link in $L_v^{\downarrow}$ is split at v_0 and after this, we have $x^*(L_{v_0}^{\downarrow}) = 0$. By Proposition 8.9, for every vertex v_1 considered after v_0, we also have $x^{*,v_1}(L^{\downarrow}) = 0$, i.e., $\mathrm{supp}(x^{*,v_1}) \cap L_v^{\downarrow} = \emptyset$. But this tells us that no vertex v_1 considered after v_0 can see any arc in $A_{up} \cap A_v$. Indeed, if v_1 is considered after v_0, then $v_1 \notin U_{v_0}$ because every vertex in $U_{v_0} \setminus \{v_0\}$ has a larger distance to r than v_0. If there were an arc $a \in A_{up} \cap A_v$ visible to v_1 (w.r.t. $\mathrm{supp}(x^{*,v_1})$), then there were a link $\ell' = (y', z') \in \mathrm{supp}(x^{*,v_1})$ such that $a \in \overrightarrow{\mathrm{cov}}(\ell')$ and $v_1 \in \mathrm{in}(\overline{P}_{\ell'})$. In particular, $z' \in U_{v_0} \setminus \{v_0\}$ and $y' \notin U_{v_0}$ (as $\mathrm{in}(\overline{P}_{\ell'}) \subseteq U_{v_0}$ otherwise). So $\ell' \in \mathrm{supp}(x^{*,v_1}) \cap L_v^{\downarrow}$, a contradiction. Hence, $A_v \cap A_{up} \cap A_{v_1}^{vis}(\mathrm{supp}(x^{*,v_1})) = \emptyset$ for every v_1 that is considered after v. As $\overrightarrow{\mathrm{cov}}(\ell) \cap A_{up} \subseteq A_{up} \cap A_v$, this concludes the proof. ∎

We are now ready to prove Theorem 8.3 *(i)*.

Lemma 8.12. $c(x^*) \leq (1 + \varepsilon) \cdot c(x)$.

Proof. We use Proposition 8.10 and that the sets $\{\ell \in L \colon \overrightarrow{\mathrm{cov}}(\ell) \cap A_{up} \cap A_v^{vis}(\mathrm{supp}(x^{*,v})) \neq \emptyset\} \setminus L_v^{\downarrow}$ for $v \in W_{up}$ and $\{\ell \in L \colon \overrightarrow{\mathrm{cov}}(\ell) \cap A_{down} \cap A_v^{vis}(\mathrm{supp}(x^{*,v})) \neq \emptyset\} \setminus L_v^{\uparrow}$ for $v \in W_{down}$ are pairwise disjoint by Proposition 8.11. Hence, Proposition 8.10 yields

$$\sum_{\ell \in L} c(x^*) \leq c(x) + 2 \cdot \Delta \cdot \gamma \cdot x(L) \leq c(x) + 2 \cdot \Delta \cdot \gamma \cdot c(x) \leq (1 + \varepsilon) \cdot c(x),$$

where we used $c(\ell) \geq 1$ for all $\ell \in L$ for the second and (8.1) for the third inequality. ∎

Next, we establish Theorem 8.3 *(ii)*.

Lemma 8.13. *For $v \in W_{up}$, we have $x^*(L_v^{\downarrow}) = 0$ and for $v \in W_{down}$, we have $x^*(L_v^{\uparrow}) = 0$. In particular, every vertex in W^* is up- or down-independent with respect to* $\mathrm{supp}(x^*)$.

Proof. r is both up- and down-independent since $A = A_r$. For $v \in W_{up}$, we have $x^*(L_v^{\downarrow}) = 0$ immediately after splitting all links in $L_v^{\downarrow}$ at v. By Proposition 8.9, this property is preserved until the end, so $\operatorname{supp}(x^*) \cap L_v^{\downarrow} = \emptyset$. By Proposition 8.5, v is up-independent. Analogously, we can establish that every vertex in W_{down} is down-independent. ∎

We are left with proving Theorem 8.3 *(iii)*. Let $L' = \operatorname{supp}(\operatorname{split}(x^*, \sigma))$ be obtained from $\operatorname{supp}(x^*)$ by splitting all W^*-cross-links at their apex.

LEMMA 8.14. *With respect to the link set L', we have* $\operatorname{viwidth}(r) = 0$, $\operatorname{viwidth}_{up}(v) = 0$ *for* $v \in W_{up}$, *and* $\operatorname{viwidth}_{down}(v) = 0$ *for* $v \in W_{down}$.

Proof. Every link ℓ with $r \in \operatorname{in}(\overline{P}_\ell)$ is an r-cross-link. As $r \in W^*$, there is no link in $\ell \in L'$ with $r \in \operatorname{in}(\overline{P}_\ell)$. Hence, no arc is visible from r and $\operatorname{viwidth}(r) = 0$.

Next, $v \in W_{up}$ and let $a \in A_v \cap A_{up}$. We need to show that a is not visible for v. Let $\ell = (y, z) \in L'$ be a link with $a \in \overrightarrow{\operatorname{cov}}(\ell)$. Then $z \in U_v \setminus \{v\}$. As we have observed in the proof of Lemma 8.13, $\operatorname{supp}(x^*) \cap L_v^{\downarrow} = \emptyset$. As $L' = \operatorname{supp}(\operatorname{split}(x^*, \sigma))$, we also have $L' \cap L_v^{\downarrow} = \emptyset$ by Proposition 8.9. Hence, $y \in U_v$, so $\operatorname{apex}(\ell) \in U_v$. As L' contains no v-cross-links, $v \notin \operatorname{in}(\overline{P}_\ell)$. The statement for $v \in W_{down}$ can be derived analogously. ∎

The following lemma concludes the proof of Theorem 8.3 *(iii)*.

LEMMA 8.15. *Let* $v \in V \setminus \{r\}$.

- *If* $\operatorname{viwidth}_{up}(v) > k$ *with respect to L', then* $x^*(L_v^{\downarrow}) > \zeta_2$, *and* a_v *is a* ζ_2*-heavy down-arc.*
- *If* $\operatorname{viwidth}_{down}(v) > k$ *with respect to L', then* $x^*(L_v^{\uparrow}) > \zeta_2$, *and* a_v *is a* ζ_2*-heavy up-arc.*

Proof. We only prove the first statement, the second one can be derived analogously. Let $v \in V \setminus \{r\}$ with $\operatorname{viwidth}_{up}(v) > k$. By Lemma 8.14, $v \notin W_{up}$. We begin by showing the following claim.

CLAIM 8.16. $x(L_v^{\downarrow}) > \zeta_2$.

Proof of claim. Assume towards a contradiction that $x(L_v^{\downarrow}) \leq \zeta_2$.
As $v \notin W_{up}$, we know that

$$x(L_v^{\downarrow}) > \gamma \cdot x(\{\ell \in L \colon \overrightarrow{\operatorname{cov}}(\ell) \cap A_{up} \cap A_v^{vis}(\operatorname{supp}(x^{*,v})) \neq \emptyset\} \setminus L_v^{\downarrow}).$$

As $\operatorname{viwidth}_{up}(v) > k$, let $a_1, \ldots, a_{k+1} \in A_v$ be ancestor-free up-arcs that are visible for v with respect to L'. As L' arises from $\operatorname{supp}(x^{*,v})$ by splitting links, $a_1, \ldots, a_{k+1}$ are also visible for v with respect to $\operatorname{supp}(x^{*,v})$. This implies

$$x(L_v^{\downarrow}) > \gamma \cdot x(\{\ell \in L \colon \overrightarrow{\operatorname{cov}}(\ell) \cap \{a_1, \ldots, a_{k+1}\} \neq \emptyset\} \setminus L_v^{\downarrow}),$$

which yields

$$\begin{aligned} x(\{\ell \in L \colon \overrightarrow{\operatorname{cov}}(\ell) \cap \{a_1, \ldots, a_{k+1}\} \neq \emptyset\}) &\leq x(\{\ell \in L \colon \overrightarrow{\operatorname{cov}}(\ell) \cap \{a_1, \ldots, a_{k+1}\} \neq \emptyset\} \setminus L_v^{\downarrow}) + x(L_v^{\downarrow}) \\ &< (1 + \gamma^{-1}) \cdot x(L_v^{\downarrow}) \leq (1 + \gamma^{-1}) \cdot \zeta_2. \end{aligned}$$

On the other hand, as the arcs $a_1, \ldots, a_{k+1}$ are ancestor-free, there is no link ℓ covering two of them. Using that x is a solution to (2.1), this implies

$$k + 1 \leq \sum_{i=1}^{k+1} x(\{\ell \colon a_i \in \overrightarrow{\operatorname{cov}}(\ell)\}) = x(\{\ell \in L \colon \overrightarrow{\operatorname{cov}}(\ell) \cap \{a_1, \ldots, a_{k+1}\} \neq \emptyset\}) \leq (1 + \gamma^{-1}) \cdot \zeta_2 \overset{(8.5)}{<} k + 1,$$

a contradiction. ∎

As $v \notin W_{up}$, the only splitting at v that we might perform in the course of Algorithm 8.1 is $\sigma_{v, L_v^{\uparrow}}$ (in case $v \in W_{down}$). Using that $L_v^{\downarrow} \cap L_v^{\uparrow} = \emptyset$, by Proposition 8.9, we can infer that $x^*(L_v^{\downarrow}) = x(L_v^{\downarrow}) > \zeta_2$. Finally, we observe that a_v must be a down-arc. Indeed, if a_v were an up-arc, then every link in $L_v^{\downarrow}$ would cover a_v, implying that a_v would be ζ_2-covered for x (and x^*). However, this contradicts (8.6) and (8.3). Hence, a_v is a ζ_2-heavy down-arc. ∎

This concludes the proof of Theorem 8.3.

9 Components and cores: handling heavy coverage in the wrong direction For this section, we again fix a rooted WDTAP instance (T, L, c, r) with cost ratio at most Δ and a solution x to (2.1) satisfying (8.6). Moreover, let σ^*, x^* and W^* be as given by Theorem 8.3. The goal of this section is to prove Theorem 9.7, which allows us to establish strong structural properties with respect to the ζ_2-heavy arcs (for x^*). To state Theorem 9.7, we require the following definitions.

Definition 9.1. *An up-component (down-component) is a (weakly) connected component of the digraph (V, A_{up}) ((V, A_{down})). We denote the collection of up- and down-components by $\mathcal{C}_{up}$ and $\mathcal{C}_{down}$, respectively, and we let $\mathcal{C} \coloneqq \mathcal{C}_{up} \cup \mathcal{C}_{down}$. We say that C is a component if C is an up- or a down-component, i.e., $C \in \mathcal{C}$. For a component C, we let the root r_C of C be the vertex of C closest to the root r of T.*

Definition 9.2. *Let $C = (V', A')$ be a component. We call an arc $a' \in A'$ a base arc if a' is ζ_2-heavy (w.r.t. x^*) and moreover, no arc of C below a' has this property. We denote the set of base arcs of C by B_C.*

Proposition 9.3. *For every component C, B_C is ancestor-free.* □

Corollary 9.4. *Let $C \in \mathcal{C}$ and let $b, b' \in B_C$ with $b \neq b'$. Then $\{\ell \in L \colon b \in \overleftarrow{\mathrm{cov}}(\ell)\} \cap \{\ell \in L \colon b' \in \overleftarrow{\mathrm{cov}}(\ell)\} = \emptyset$.*

Proof. Let $\ell \in L$. All arcs in $\overleftarrow{\mathrm{cov}}(\ell) \cap A_{up}$, as well as all arcs in $\overleftarrow{\mathrm{cov}}(\ell) \cap A_{down}$, share a pairwise ancestral relationship. Then fact that B_C is an ancestor-free set are up-arcs, if $C \in \mathcal{C}_{up}$, and an ancestor-free set of down-arcs, if $C \in \mathcal{C}_{down}$, concludes the proof. □

Definition 9.5. *Let C be a component. The core $\mathring{C}$ of C consists of the union of the paths connecting the (lower vertices of) the base arcs to r_C, if $B_C \neq \emptyset$, and is empty otherwise. Let $\mathring{\mathcal{C}}_{up} \coloneqq \{\mathring{C} \colon C \in \mathcal{C}_{up}, B_C \neq \emptyset\}$, $\mathring{\mathcal{C}}_{down} \coloneqq \{\mathring{C} \colon C \in \mathcal{C}_{down}, B_C \neq \emptyset\}$ and $\mathring{\mathcal{C}} \coloneqq \mathring{\mathcal{C}}_{up} \cup \mathring{\mathcal{C}}_{down}$ be the collection of all non-empty cores.*

For a core $\mathring{C} \in \mathring{\mathcal{C}}$, we call $B_{\mathring{C}} \coloneqq B_C$ the set of base arcs of $\mathring{C}$.

Proposition 9.6. *Let $v \in V \setminus \{r\}$ such that a_v is ζ_2-heavy, and let $C \in \mathcal{C}$ be the component containing a_v. Then a_v is contained in $\mathring{C}$.*

Proof. This is clear if $a_v \in B_C$. Otherwise, there exists a base arc $b \in B_C$ such that a_v lies on the path from the bottom vertex of b to r_C. Hence, a_v is contained in $\mathring{C}$. □

We are now ready to state the main theorem of this section.

Theorem 9.7. *We can, in polynomial time, compute a splitting σ^{**} of L and $x^{**} \coloneqq \mathrm{split}(x^*, \sigma^{**}) = \mathrm{split}(x, \sigma^{**} \circ \sigma^*)$ such that the following properties hold:*

(9.7.1) $c(x^{**}) \leq (1+\varepsilon)^2 \cdot c(x)$

(9.7.2) Let $C \in \mathring{\mathcal{C}}$. There is no $\ell \in \mathrm{supp}(x^{**})$ with $\overrightarrow{\mathrm{cov}}(\ell) \cap A(C) \neq \emptyset$ and $\overleftarrow{\mathrm{cov}}(\ell) \cap A(C) \neq \emptyset$.

(9.7.3) Let $C \in \mathring{\mathcal{C}}_{up}$. Then $\mathrm{supp}(x^{**}) \cap L^{\uparrow}_{r_C} = \emptyset$ and for every $\ell \in \mathrm{supp}(x^{**}) \cap L^{\downarrow}_{r_C}$, $\mathrm{cov}(\ell) \cap A(C) = \emptyset$.
Let $C \in \mathring{\mathcal{C}}_{down}$. Then $\mathrm{supp}(x^{**}) \cap L^{\downarrow}_{r_C} = \emptyset$ and for every $\ell \in \mathrm{supp}(x^{**}) \cap L^{\uparrow}_{r_C}$, $\mathrm{cov}(\ell) \cap A(C) = \emptyset$.

(9.7.4) Let $C \in \mathring{\mathcal{C}}$ and let $\ell \in \mathrm{supp}(x^{**})$. If $\overrightarrow{\mathrm{cov}}(\ell)$ contains an arc of C incident to r_C, then r_C is an endpoint of ℓ.

Before we move on to proving this theorem, we state an application of it.

Corollary 9.8. *Let σ be a splitting of L and let $\ell \in \mathrm{supp}(\mathrm{split}(x^{**}, \sigma))$. Let $C \in \mathring{\mathcal{C}}$ such that $\mathrm{cov}(\ell) \cap A(C) \neq \emptyset$. Then $\mathrm{apex}(\ell) \in V(C)$.*

Proof. We only consider the case where $C \in \mathring{\mathcal{C}}_{up}$, the case $C \in \mathring{\mathcal{C}}_{down}$ can be handled analogously. By (9.7.3) and Proposition 8.9, we know that $\mathrm{supp}(\mathrm{split}(x^{**}, \sigma)) \cap L^{\uparrow}_{r_C} = \emptyset$, so $\ell \notin L^{\uparrow}_{r_C}$. We further show that $\ell \notin L^{\downarrow}_{r_C}$. As $\ell \in \mathrm{supp}(\mathrm{split}(x^{**}, \sigma))$, there is $\ell' \in \mathrm{supp}(x^{**})$ with $\ell \in \sigma(\ell')$. Then ℓ is a shadow of ℓ', implying that also $\mathrm{cov}(\ell') \cap A(C) \neq \emptyset$. By (9.7.3), ℓ' is neither contained in $L^{\uparrow}_{r_C}$ nor in $L^{\downarrow}_{r_C}$, hence, both endpoints of ℓ' must be contained in U_{r_C} (because also $\mathrm{cov}(\ell') \cap A_{r_C} \supseteq \mathrm{cov}(\ell') \cap A(C) \neq \emptyset$). As ℓ is a shadow of ℓ', both endpoints of ℓ are contained in U_{r_C} as well. As there is $a \in A(C) \cap \mathrm{cov}(\ell)$, $\mathrm{apex}(\ell)$ must be an ancestor of $\mathrm{apex}(a)$, but a descendant of r_C. Hence, $\mathrm{apex}(\ell) \in V(C)$. □

The rest of this section is dedicated to the proof of Theorem 9.7. We first conduct a structural analysis of links in $\mathrm{supp}(x^*)$ violating properties (9.7.2), (9.7.3) and (9.7.4). Then, we describe a link splitting procedure (Algorithm 9.1) designed to make sure that the support of the resulting solution x^{**} does not contain any of the "problematic links". Finally, we explain how to charge the cost of the splitting against the total costs of the links "heavily covering base arcs in the wrong direction".

9.1 Structural analysis of problematic links For every $C \in \mathring{\mathcal{C}}$, fix an ordering $B_C = \{b_1^C, \dots, b_{t_C}^C\}$ of its base arcs. We introduce a decomposition of the cores into pairwise arc-disjoint paths, that will allow us to characterize the structure of the "problematic links" and guide our charging procedure when splitting them.

Definition 9.9 (trunk decomposition). *Let $C \in \mathring{\mathcal{C}}$. The trunk decomposition of C is the decomposition of C into arc-disjoint paths $P_1^C, \dots, P_{t_C}^C$ defined as follows:*

- *P_1^C is the path connecting the bottom vertex of b_1^C to r_C.*
- *For $i = 2, \dots, t_C$, let P_i' be the path connecting the bottom vertex of b_i^C to r_C and let P_i^C be the prefix of P_i' ending at the first vertex in $V(P_i') \cap \bigcup_{j=1}^{i-1} V(P_j^C)$.*

The paths $(P_i^C)_{i=1}^{t_C}$ are called the trunks of the trunk decomposition.

Note that we can compute $\mathring{\mathcal{C}}_{up}$, $\mathring{\mathcal{C}}_{down}$ and a trunk decomposition of every core, in polynomial time: following the definition, each step can be done in linear time and the total number of base arcs is bounded by the total number of arcs.

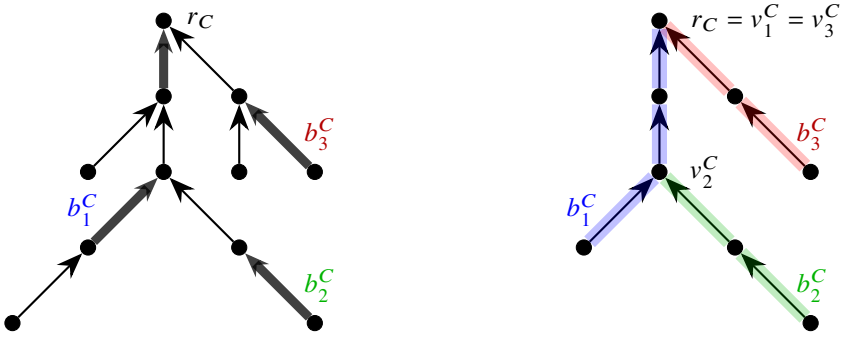


Figure 9.1: An up-component C with root r_C is shown on the left. The ζ_2-heavy arcs are drawn in bold with b_1^C, b_2^C and b_3^C being the base arcs. The core of C is shown on the right, together with its trunk decomposition, indicated by colors. Here, P_1^C (blue) is the parent trunk of P_2^C (green). The sibling arc of P_2^C is b_1^C.

Definition 9.10. *For a trunk P_i^C, we denote its top endpoint by v_i^C and its top arc by a_i^C.*

Definition 9.11. *Let $C \in \mathring{\mathcal{C}}$ and let $(P_i^C)_{i=1}^{t_C}$ be the trunks of the trunk decomposition of C. For $i \in \{1, \dots, t_C\}$ with $v_i^C \neq r_C$, we define the parent trunk of P_i^C to be the trunk P_j^C containing $a_{v_i^C}$. Note that as B_C is ancestor-free, v_i^C cannot be the bottom endpoint of P_j^C. We further call the arc of P_j^C connecting to v_i^C from below the sibling arc of P_i^C and denote it by s_i^C.*

We are now ready to analyze the structure of links that violate the condition in (9.7.2).

Proposition 9.12. *Let $C \in \mathring{\mathcal{C}}$ and let $\ell \in L$ with $\overrightarrow{\mathrm{cov}}(\ell) \cap A(C) \neq \emptyset$ and $\overleftarrow{\mathrm{cov}}(\ell) \cap A(C) \neq \emptyset$. Then:*

(9.12.1) ℓ is neither an up- nor a down-link, i.e., $\mathrm{apex}(\ell) \in \mathrm{in}(P_\ell)$.

(9.12.2) There is a trunk P_i^C such that $\mathrm{apex}(\ell) = v_i^C$ *and either $a_i^C \in \overrightarrow{\mathrm{cov}}(\ell)$, or $v_i^C \neq r_C$ and $s_i^C \in \overrightarrow{\mathrm{cov}}(\ell)$.*

Proof. The fact that $A(C)$ either only contains up-arcs or only down-arcs implies that for an up- or down-link ℓ, $\overrightarrow{\mathrm{cov}}(\ell) \cap A(C) = \emptyset$ or $\overleftarrow{\mathrm{cov}}(\ell) \cap A(C) = \emptyset$. This establishes (9.12.1). To simplify notation, we prove (9.12.2) only for the case where $C \in \mathring{\mathcal{C}}_{up}$; the case $C \in \mathring{\mathcal{C}}_{down}$ can be handled analogously. Let $\ell = (u, v)$, let $a \in \overrightarrow{\mathrm{cov}}(\ell) \cap A(C)$ and let $a' \in \overleftarrow{\mathrm{cov}}(\ell) \cap A(C)$. Then a lies on the $\mathrm{apex}(\ell)$-v-path in T and a' lies on the u-$\mathrm{apex}(\ell)$-path in T. In particular, $\mathrm{apex}(\ell)$ is the lowest common ancestor of (the top vertices of) a and a', and, as C is connected, $\mathrm{apex}(\ell) \in V(C)$. Let a_0 and a_0' be the top arcs of the v-$\mathrm{apex}(\ell)$-path and the u-$\mathrm{apex}(\ell)$-path, respectively. Then $a_0 \in \overrightarrow{\mathrm{cov}}(\ell)$ and $a_0' \in \overleftarrow{\mathrm{cov}}(\ell)$. Let P_j^C be the trunk containing a_0. If $v_j^C = \mathrm{apex}(\ell)$, we let $P_i^C := P_j^C$. Otherwise, $\mathrm{apex}(\ell) \neq r_C$ and $a_{\mathrm{apex}(\ell)} \in A(P_j^C)$. In this case, we let P_i^C be the trunk containing a_0'. Then $v_i^C = \mathrm{apex}(\ell)$ and $a_0 = s_i^C$ is the sibling arc of P_i^C. ∎

The next two propositions help us to understand the structure of links violating the conditions in (9.7.3).

Proposition 9.13.

- *Let $C \in \mathring{\mathcal{C}}_{up}$ and let $\ell \in L^{\uparrow}_{r_C}$. Then $r_C \neq r$, $r_C \in \mathrm{in}(P_\ell)$ and $a_{r_C} \in \overrightarrow{\mathrm{cov}}(\ell)$.*
- *Let $C \in \mathring{\mathcal{C}}_{down}$ and let $\ell \in L^{\downarrow}_{r_C}$. Then $r_C \neq r$, $r_C \in \mathrm{in}(P_\ell)$ and $a_{r_C} \in \overrightarrow{\mathrm{cov}}(\ell)$.*

Proof. We only prove the first statement; the second one follows analogously. Let $C \in \mathring{\mathcal{C}}_{up}$ and let $\ell = (u, v) \in L^{\uparrow}_{r_C}$. Then $u \in U_{r_C} \setminus \{r_C\}$ and $v \in V \setminus U_{r_C}$, so $r_C \in \mathrm{in}(P_\ell)$. Moreover, $\mathrm{apex}(\ell)$ is a strict ancestor of r_C (implying $r_C \neq r$) and a_{r_C} appears on the u-$\mathrm{apex}(\ell)$-path in T. As $C \in \mathring{\mathcal{C}}_{up}$, a_{r_C} is a down-arc, so $a_{r_C} \in \overrightarrow{\mathrm{cov}}(\ell)$. ∎

Proposition 9.14.

- *Let $C \in \mathring{\mathcal{C}}_{up}$ and let $\ell \in L^{\downarrow}_{r_C}$ with $\mathrm{cov}(\ell) \cap A(C) \neq \emptyset$. Then $r_C \in \mathrm{in}(P_\ell)$ and there is a trunk P^C_i with $v^C_i = r_C$ and $a^C_i \in \overrightarrow{\mathrm{cov}}(\ell)$.*
- *Let $C \in \mathring{\mathcal{C}}_{down}$ and let $\ell \in L^{\uparrow}_{r_C}$ with $\mathrm{cov}(\ell) \cap A(C) \neq \emptyset$. Then $r_C \in \mathrm{in}(P_\ell)$ and there is a trunk P^C_i with $v^C_i = r_C$ and $a^C_i \in \overrightarrow{\mathrm{cov}}(\ell)$.*

Proof. Again, we only prove the first statement. Let $C \in \mathring{\mathcal{C}}_{up}$ and let $\ell = (u, v) \in L^{\downarrow}_{r_C}$ with $\mathrm{cov}(\ell) \cap A(C) \neq \emptyset$. Then $v \in U_{r_C} \setminus \{r_C\}$ and $u \notin U_{r_C}$. In particular, $r_C \in \mathrm{in}(P_\ell)$ and moreover, $\mathrm{apex}(\ell)$ is a strict ancestor of r_C. Let $a \in \mathrm{cov}(\ell) \cap A(C)$. Then a lies on the v-r_C-path in T. Let a' be the last arc of this path. As C is connected, $a' \in A(C)$, so there is a trunk P^C_i containing a'. As a' is incident to r_C, $a' = a^C_i$ and $v^C_i = r_C$. Finally, as $C \in \mathring{\mathcal{C}}_{up}$, $a^C_i \in A_{up}$. As a^C_i lies on the v-$\mathrm{apex}(\ell)$-path in T, $a^C_i \in \overrightarrow{\mathrm{cov}}(\ell)$. ∎

Propositions 9.12 to 9.14 and (9.7.4) motivate the splitting procedure presented in the following section.

9.2 The splitting algorithm We obtain σ^{**} and $x^{**} = \mathrm{split}(x^*, \sigma^{**})$ via Algorithm 9.1.

Algorithm 9.1 Core link splitting.

Input: solution x^* to (2.1)
Output: splitting σ^{**} of L, solution $x^{**} = \mathrm{split}(x^*, \sigma^{**})$ to (2.1)

$\sigma^{**} \leftarrow \sigma_{\mathrm{id}}$, $x^{**} \leftarrow x^*$
for $C \in \mathring{\mathcal{C}}$ **do**
  $L' \leftarrow \{\ell \in L \colon a_{r_C} \in \overrightarrow{\mathrm{cov}}(\ell)\}$
  $\sigma^{**} \leftarrow \sigma_{r_C, L'} \circ \sigma^{**}$, $x^{**} \leftarrow \mathrm{split}(x^{**}, \sigma_{r_C, L'})$
  for $i \leftarrow 1$ **to** t_C **do**
    $L' \leftarrow \{\ell \in L \colon a^C_i \in \overrightarrow{\mathrm{cov}}(\ell)\}$
    $\sigma^{**} \leftarrow \sigma_{v^C_i, L'} \circ \sigma^{**}$, $x^{**} \leftarrow \mathrm{split}(x^{**}, \sigma_{v^C_i, L'})$
    if $v^C_i \neq r_C$ **then**
      $L' \leftarrow \{\ell \in L \colon s^C_i \in \overrightarrow{\mathrm{cov}}(\ell)\}$
      $\sigma^{**} \leftarrow \sigma_{v^C_i, L'} \circ \sigma^{**}$, $x^{**} \leftarrow \mathrm{split}(x^{**}, \sigma_{v^C_i, L'})$
    end if
  end for
end for
return σ^{**}, x^{**}

Note that Algorithm 9.1 runs in polynomial time. The following technical claim is useful to further analyze Algorithm 9.1.

Proposition 9.15. *Let $a \in A$ and let u be an endpoint of a. Let x be a solution to* (2.1) *such that for every $\ell \in \mathrm{supp}(x)$ with $a \in \overrightarrow{\mathrm{cov}}(\ell)$, u is an endpoint of ℓ.*

Let $v \in V$, $L' \subseteq L$ and let $x' \coloneqq \mathrm{split}(x, \sigma_{v,L'})$. Then for every $\ell \in \mathrm{supp}(x')$ with $a \in \overrightarrow{\mathrm{cov}}(\ell)$, u is an endpoint of ℓ.

Proof. Let $\ell \in \mathrm{supp}(x')$ with $a \in \overrightarrow{\mathrm{cov}}(\ell)$. If $\ell \in \mathrm{supp}(x)$, the statement follows from our assumption on x. Otherwise, there is $\ell' \in L' \cap \mathrm{supp}(x)$ such that ℓ is a shadow of ℓ' with $a \in \overrightarrow{\mathrm{cov}}(\ell)$. Then also $a \in \overrightarrow{\mathrm{cov}}(\ell')$, so u is an endpoint of ℓ'. As u is an endpoint of ℓ' and a, ℓ is a shadow of ℓ' and $a \in \mathrm{cov}(\ell)$, u must be an endpoint of ℓ as well. ∎

Proposition 9.16. *The solution x^{**} computed by Algorithm* 9.1 *satisfies (9.7.2).*

Proof. Let $C \in \mathring{\mathcal{C}}$ and let P_i^C be a trunk of C. We know that for every link $\ell \in \operatorname{supp}(x^{**})$ with $a_i^C \in \overrightarrow{\operatorname{cov}}(\ell)$, v_i^C is an endpoint of ℓ because this property holds immediately after line 7 of Algorithm 9.1 is executed (for C and i) and it is preserved by later splits by Proposition 9.15.

If $r_i^C \neq r_C$, we further know that for every link $\ell \in \operatorname{supp}(x^{**})$ with $s_i^C \in \overrightarrow{\operatorname{cov}}(\ell)$, v_i^C is an endpoint of ℓ because this property holds immediately after line 10 of Algorithm 9.1 is executed (for C and i) and it is preserved by later splits by Proposition 9.15.

By Proposition 9.12, (9.7.2) is satisfied. ∎

Proposition 9.17. *The solution x^{**} computed by Algorithm* 9.1 *satisfies (9.7.3).*

Proof. We only prove the statement for $C \in \mathring{\mathcal{C}}_{up}$; the case $C \in \mathring{\mathcal{C}}_{down}$ can be handled analogously.

Let $C \in \mathring{\mathcal{C}}_{up}$ with $r_C \neq r$ (otherwise, $L^{\uparrow}_{r_C} = \emptyset$ and $L^{\downarrow}_{r_C} = \emptyset$ and there is nothing to show). We know that every link $\ell \in \operatorname{supp}(x^{**})$ with $a_{r_C} \in \overrightarrow{\operatorname{cov}}(\ell)$ has r_C as an endpoint because this property holds immediately after line 4 of Algorithm 9.1 is executed for C and it is preserved by later splits by Proposition 9.15. By Proposition 9.13, $\operatorname{supp}(x^{**}) \cap L^{\uparrow}_{r_C} = \emptyset$.

Let P_i^C be a trunk of C with $v_i^C = r_C$. We know that for every $\ell \in \operatorname{supp}(x^{**})$ with $a_i^C \in \overrightarrow{\operatorname{cov}}(\ell)$, $v_i^C = r_C$ is an endpoint of ℓ because this property holds immediately after line 7 of Algorithm 9.1 is executed (for C and i) and it is preserved by later splits by Proposition 9.15. By Proposition 9.14, for every $\ell \in \operatorname{supp}(x^{**}) \cap L^{\downarrow}_{r_C}$, $\operatorname{cov}(\ell) \cap A(C) = \emptyset$. ∎

Proposition 9.18. *The solution x^{**} computed by Algorithm* 9.1 *satisfies (9.7.4).*

Proof. Let $C \in \mathring{\mathcal{C}}$ and let $a \in A(C)$ be an arc incident to r_C. Then a is the top arc of some trunk of C ending in r_C, i.e., there is $i \in \{1, \dots, t_C\}$ such that $a = a_i^C$ and $v_i^C = r_C$. Now, every link $\ell \in \operatorname{supp}(x^{**})$ with $a \in \overrightarrow{\operatorname{cov}}(\ell)$ has r_C as an endpoint because this property holds immediately after line 7 of Algorithm 9.1 is executed for i and C, and it is preserved by later splits by Proposition 9.15. ∎

To conclude the proof of Theorem 9.7, it remains to establish (9.7.1), which is the goal of the following section.

9.3 Bounding the cost of the splitting In order to bound the costs of the splitting operations performed in Algorithm 9.1, we first establish a lower bound on the total cost of x^*. Lemma 9.19 allows us to relate the total x^*-value on links who have their apex in a core C to the number t_C of base arcs of C. Corollary 9.21 then gives a lower bound on the cost of x^* in terms of the total number of base arcs in all cores. For the proof of Lemma 9.19, we observe that since there are no ζ_1-covered arcs with respect to x, Proposition 7.5 implies that

(9.1) there are no ζ_1-covered arcs with respect to x^*.

Lemma 9.19. *Let $C \in \mathring{\mathcal{C}}$. Then $x^*(\{\ell \in L \colon \operatorname{apex}(\ell) \in V(C)\}) \geq (1-\varepsilon) \cdot \zeta_2 \cdot t_C$.*

Proof. For $i \in \{1, \dots, t_C\}$, let $L_i := \{\ell \in L \colon b_i^C \in \overleftarrow{\operatorname{cov}}(\ell)\}$. By Corollary 9.4, the sets $(L_i)_{i=1}^{t_C}$ are pairwise disjoint. Moreover, as every base arc is ζ_2-heavy, $x^*(L_i) \geq \zeta_2$ for every $i \in \{1, \dots, t_C\}$.

Claim 9.20. *Let $i \in \{1, \dots, t_C\}$ and let $\ell \in L_i$. If $\operatorname{apex}(\ell) \notin V(C)$, then $r_C \neq r$ and $a_{r_C} \in \overrightarrow{\operatorname{cov}}(\ell)$.*

Proof of claim. To simplify notation, we assume that $C \in \mathring{\mathcal{C}}_{up}$; the case $C \in \mathring{\mathcal{C}}_{down}$ can be handled analogously. Let $\ell = (u, v)$. As $b_i^C \in \overleftarrow{\operatorname{cov}}(\ell)$, the bottom vertex w of b_i^C is an ancestor of u and $\operatorname{apex}(\ell)$ is strict ancestor of w. If $\operatorname{apex}(\ell)$ appears on the w-r_C-path in T, then $\operatorname{apex}(\ell) \in V(C)$. Otherwise, $\operatorname{apex}(\ell)$ is a strict ancestor of r_C and in particular, $r_C \neq r$. Moreover, a_{r_C} is a down-arc that appears on the u-$\operatorname{apex}(\ell)$-path in T, implying $a_{r_C} \in \overrightarrow{\operatorname{cov}}(\ell)$. ∎

By the claim, if $r = r_C$, then

$$x^*(\{\ell \in L \colon \operatorname{apex}(\ell) \in V(C)\}) \geq \sum_{i=1}^{t_C} x^*(L_i) \geq t_C \cdot \zeta_2.$$

Next, assume that $r \neq r_C$. The claim yields

$$x^*(\{\ell \in L \colon \operatorname{apex}(\ell) \in V(C)\}) \geq \sum_{i=1}^{t_C} x^*(L_i) - x^*(\{\ell \in L \colon a_{r_C} \in \overrightarrow{\operatorname{cov}}(\ell)\}) \overset{(9.1)}{>} t_C \cdot \zeta_2 - \zeta_1 \overset{(8.3)}{>} (1-\varepsilon) \cdot \zeta_2 \cdot t_C.$$

∎

Corollary 9.21. *We have* $c(x^*) \geq \frac{1}{2} \cdot (1-\varepsilon) \cdot \zeta_2 \cdot \sum_{C \in \mathring{\mathcal{C}}} t_C$.

Proof. This follows from Lemma 9.19, using that the link costs lie in $[1, \Delta]$ and that for every $\ell \in L$, there is at most one $C \in \mathring{\mathcal{C}}_{up}$ and at most one $C \in \mathring{\mathcal{C}}_{down}$ with $\operatorname{apex}(\ell) \in V(C)$ because the up-cores, as well as the down-cores, are pairwise vertex-disjoint. ∎

Lemma 9.22. *We have* $c(x^{**}) \leq (1+\varepsilon)^2 \cdot c(x)$, *i.e., (9.7.1) holds.*

Proof. Using Proposition 7.5, and Proposition 8.8, we obtain

$$\begin{aligned}
c(x^{**}) &\leq c(x^*) + \Delta \cdot \sum_{C \in \mathring{\mathcal{C}}} x^*(\{\ell \in L \colon a_{r_C} \in \overrightarrow{\operatorname{cov}}(\ell)\}) \\
&\quad + \Delta \cdot \sum_{C \in \mathring{\mathcal{C}}} \left[\sum_{i=1}^{t_C} x^*(\{\ell \in L \colon a_i^C \in \overrightarrow{\operatorname{cov}}(\ell)\}) + \sum_{i=1, v_i^C \neq r_C}^{t_C} x^*(\{\ell \in L \colon s_i^C \in \overrightarrow{\operatorname{cov}}(\ell)\}) \right] \\
&\overset{(9.1)}{\leq} c(x^*) + \zeta_1 \cdot \Delta \cdot |\mathring{\mathcal{C}}| + 2 \cdot \zeta_1 \cdot \Delta \cdot \sum_{C \in \mathring{\mathcal{C}}} t_C \\
&\overset{(*)}{\leq} (1+\varepsilon) \cdot c(x) + 3 \cdot \zeta_1 \cdot \Delta \cdot \sum_{C \in \mathring{\mathcal{C}}} t_C \overset{(8.4)}{\leq} (1+\varepsilon) \cdot c(x) + \varepsilon \cdot \frac{1}{2} \cdot (1-\varepsilon) \cdot \zeta_2 \cdot \sum_{C \in \mathring{\mathcal{C}}} t_C \\
&\leq (1+\varepsilon) \cdot c(x) + \varepsilon \cdot c(x^*) \leq (1+\varepsilon)^2 \cdot c(x),
\end{aligned}$$

where the inequality marked $(*)$ follows from Theorem 8.3 *(i)* and the fact that $t_C \geq 1$ for every core C, the second-to-last inequality follows from Corollary 9.21, and the last inequality follows again from Theorem 8.3 *(i)*. ∎

Combining Propositions 9.16 to 9.18, and Lemma 9.22 proves Theorem 9.7.

10 Best of three solutions For this section, we again fix a rooted WDTAP instance (T, L, c, r) with cost ratio at most Δ and a solution x to (2.1) satisfying (8.6). Moreover, let σ^*, x^* and W^* be as given by Theorem 8.3 and let x^{**} and σ^{**} be as given by Theorem 9.7. The goal of this section is to prove Theorem 7.12 by constructing three different solutions arising from x^{**} by splitting certain links. To this end, we classify and partition the links in the support of x^{**}. We begin by proving the following technical claim, that will be helpful to define disjoint subsets of $\operatorname{supp}(x^{**})$.

Proposition 10.1. *Let* $\ell \in \operatorname{supp}(x^{**})$ *and let* $C \in \mathring{\mathcal{C}}$ *such that* $\operatorname{cov}(\ell) \cap A(C) \neq \emptyset$. *Then* $\operatorname{apex}(\ell) \in V(C)$ *and* $\operatorname{cov}(\ell) \cap A(C)$ *contains an arc incident to* $\operatorname{apex}(\ell)$.

Proof. Let $a \in \operatorname{cov}(\ell) \cap A(C)$ and let v be the bottom vertex of a. Then both $\operatorname{apex}(\ell)$ and r_C appear on P_{vr}, the v-r-path in T. If $\operatorname{apex}(\ell)$ lies on the v-r_C-subpath of P_{vr}, then $\operatorname{apex}(\ell) \in V(C)$ and moreover, the last arc of the v-$\operatorname{apex}(\ell)$-subpath of P_{vr} is contained in $\operatorname{cov}(\ell) \cap A(C)$. Otherwise, $\ell \in L^{\uparrow}_{r_C} \cup L^{\downarrow}_{r_C}$ and $\operatorname{cov}(\ell) \cap A(C) \neq \emptyset$, contradicting (9.7.3). ∎

Definition 10.2. *We define* $\overrightarrow{L} := \{\ell \in L \colon \exists C \in \mathring{\mathcal{C}} \colon \overrightarrow{\operatorname{cov}}(\ell) \cap A(C) \neq \emptyset\}$ *to be the set of links that "cover part of a core in the right direction" and* $\overleftarrow{L} := \{\ell \in L \colon \exists C \in \mathring{\mathcal{C}} \colon \overleftarrow{\operatorname{cov}}(\ell) \cap A(C) \neq \emptyset\}$ *to be the set of links that "cover part of a core in the wrong direction".*

It turns out that the support of x^{**} does not contain any link in $\overrightarrow{L} \cap \overleftarrow{L}$. In fact, $\overrightarrow{L} \cap \overleftarrow{L}$ does not even contain a shadow of a link in $\operatorname{supp}(x^{**})$.

Lemma 10.3. $\overrightarrow{L} \cap \overleftarrow{L} \cap \{\ell \colon \ell$ *is a shadow of a link in* $\operatorname{supp}(x^{**})\} = \emptyset$.

Proof. Assume towards a contradiction that $\ell' \in \overrightarrow{L} \cap \overleftarrow{L}$ is a shadow of $\ell \in \operatorname{supp}(x^{**})$. Then also $\ell \in \overrightarrow{L} \cap \overleftarrow{L}$ and there are $C \in \mathring{\mathcal{C}}$ with $\overrightarrow{\operatorname{cov}}(\ell) \cap A(C) \neq \emptyset$ and $C' \in \mathring{\mathcal{C}}$ with $\overleftarrow{\operatorname{cov}}(\ell) \cap A(C) \neq \emptyset$. By (9.7.2), $\overleftarrow{\operatorname{cov}}(\ell) \cap A(C) = \emptyset$ and $\overrightarrow{\operatorname{cov}}(\ell) \cap A(C') = \emptyset$. Hence, $C \neq C'$ and, in particular, $A(C) \cap A(C') = \emptyset$. By Proposition 10.1, $\operatorname{apex}(\ell) \in V(C) \cap V(C')$ and one of the two incident arcs of $\operatorname{apex}(\ell)$ in $\operatorname{cov}(\ell)$ is contained in $A(C)$, call it a, and the other one is contained in $A(C')$, call it a'. Let $\ell = (u, v)$. Then either $a \in \overrightarrow{\operatorname{cov}}((u, \operatorname{apex}(\ell)))$ and $a' \in \overleftarrow{\operatorname{cov}}((\operatorname{apex}(\ell), v))$, implying that both a and a' are down-arcs, or $a' \in \overleftarrow{\operatorname{cov}}((u, \operatorname{apex}(\ell)))$ and $a \in \overrightarrow{\operatorname{cov}}((\operatorname{apex}(\ell), v))$, implying that both a and a' are up-arcs. In either case, C and C' are distinct cores for the same direction sharing a vertex (namely, $\operatorname{apex}(\ell)$), a contradiction. ∎

To construct the different solutions, we will split certain collections of links at their apex. To describe this operation formally, we introduce the following notation.

Definition 10.4. *Let $L' \subseteq L$. The splitting $\sigma_{L'}$ of L, which splits every link in L' (that is not already an up- or down-link) at its apex, is defined as follows. If $\ell \in L \setminus L'$ or $\ell \in L'$ is an up-link or a down-link, we define $\sigma_{L'}(\ell) = \{\ell\}$. If $\ell = (u, v) \in L'$ is neither an up- nor a down-link, we define $\sigma_{L'}(\ell) = \{(u, \operatorname{apex}(\ell)), (\operatorname{apex}(\ell), v)\}$.*

Let $X \coloneqq \bigcup_{C \in \mathring{\mathcal{C}}} V(C)$ be the set of vertices of all cores. Let L_{cross} consist of all $(W^* \cup X)$-cross-links that are neither contained in $\overrightarrow{L}$ nor in $\overleftarrow{L}$, i.e.,

$$L_{cross} \coloneqq \{\ell \in L \setminus (\overrightarrow{L} \cup \overleftarrow{L}) \colon \ \ell \text{ is a } (W^* \cup X)\text{-cross-link}\}.$$

We further define $L_{rest} \coloneqq L \setminus (L_{cross} \cup \overrightarrow{L} \cup \overleftarrow{L})$. By Lemma 10.3, we know that

$$\operatorname{supp}(x^{**}) = (L_{cross} \cap \operatorname{supp}(x^{**})) \dot\cup (\overrightarrow{L} \cap \operatorname{supp}(x^{**})) \dot\cup (\overleftarrow{L} \cap \operatorname{supp}(x^{**})) \dot\cup (L_{rest} \cap \operatorname{supp}(x^{**})) \tag{10.1}$$

is a partition of $\operatorname{supp}(x^{**})$. We are now ready to construct the three solutions of interest.

Lemma 10.5. *Let $\sigma_1 \coloneqq \sigma_{L \setminus L_{rest}} \circ \sigma^{**} \circ \sigma^*$, let $x_1 \coloneqq \operatorname{split}(x^{**}, \sigma_{L \setminus L_{rest}}) = \operatorname{split}(x, \sigma_1)$ and let $L_1 \coloneqq \operatorname{supp}(x_1)$. Then the visible width of (T, L_1) is at most k. In particular, we can in polynomial time, compute a solution to (T, L_1) of cost at most $c(x_1)$, or find violated visibly k-wide modification inequality for (T, L, c, r).*

Proof. Recall that $L_1 = \operatorname{supp}(x_1)$ and $x_1 = \operatorname{split}(x^*, \sigma_{L \setminus L_{rest}} \circ \sigma^{**})$. By Proposition 7.4 and Theorem 8.3, we know that r, as well as every $v \in V \setminus \{r\}$ for which a_v is not ζ_2-heavy (with respect to x^*), has visible width at most k with respect to L_1. Moreover, $\overrightarrow{\operatorname{viwidth}}(v) \leq k$ for every $v \in V$. Hence, it suffices to show that for every $v \in V \setminus \{r\}$ for which a_v is ζ_2-heavy, $\overleftarrow{\operatorname{viwidth}}(v) \leq k$ (with respect to L_1). In fact, we will show that $\overleftarrow{\operatorname{viwidth}}(v) = 0$ by showing the following claim:

Claim 10.6. *No arc in $\overleftarrow{A}_v$ is visible for v with respect to L_1.*

Proof of claim. Recall that $\overleftarrow{A}_v$ is the set of arcs in A_v that have the opposite orientation of a_v, i.e., that are down-arcs, if a_v is an up-arc, and up-arcs, if a_v is a down-arc. Towards a contradiction, let $a \in \overleftarrow{A}_v$ and assume that there is $\ell \in L_1$ with $a \in \overrightarrow{\operatorname{cov}}(\ell)$ and $v \in \operatorname{in}(\overline{P}_\ell)$. By Proposition 9.6, there exists a core C containing a_v. In particular, $v \in X$. We note that L_1 does not contain any $(W^* \cup X)$-cross-links because every $(W^* \cup X)$-cross-link is contained in $L \setminus L_{rest}$ and has, hence, been split at its apex when constructing L_1. In particular, ℓ cannot be a v-cross-link. As $v \in \operatorname{in}(\overline{P}_\ell)$, $\operatorname{apex}(\ell)$ has to be a strict ancestor of v and exactly one endpoint of ℓ, say u, is contained in U_v. Both a and a_v lie on the u-$\operatorname{apex}(\ell)$-path in T, however, as $a \in \overleftarrow{A}_v$, exactly one of them is an up-arc and exactly one of them is a down-arc. As $a \in \overrightarrow{\operatorname{cov}}(\ell)$, this implies $a_v \in \overleftarrow{\operatorname{cov}}(\ell)$. By Corollary 9.8, $\operatorname{apex}(\ell) \in V(C) \subseteq X$. As L_1 does not contain any $(W^* \cup X)$-cross-link, $\operatorname{apex}(\ell)$ is an endpoint of ℓ.

As all arcs on the v-$\operatorname{apex}(\ell)$-path are contained in $A(C)$ and oriented in the same way as a_v, none of them is contained in $\overrightarrow{\operatorname{cov}}(\ell)$. Hence, $\overline{P}_\ell$ is a subpath of the u-v-subpath in T. Hence, $v \notin \operatorname{in}(\overline{P}_\ell)$, a contradiction. ◻

Hence, (T, L_1, c, r) has visible width at most k and we can find an optimum solution in polynomial time by Corollary 6.8. If the optimum solution has cost at most $c(x_1) = \sum_{\ell \in L} \sum_{\ell' \in \sigma_1(\ell)} c(\ell') \cdot x_\ell$ (by Proposition 7.6), then we have found the desired solution. Otherwise, the set of ζ_1-covered arcs that we initially contracted to obtain (8.6), together with the splitting σ_1 that we applied to get from our initial LP solution x to x_1, yields a violated visibly k-wide modification inequality. ◻

Lemma 10.7. *Let $\sigma_2 \coloneqq \sigma_{\overleftarrow{L} \cup L_{rest}} \circ \sigma^{**} \circ \sigma^*$, let $x_2 \coloneqq \operatorname{split}(x^{**}, \sigma_{\overleftarrow{L} \cup L_{rest}}) = \operatorname{split}(x, \sigma_2)$ and let $L_2 \coloneqq \operatorname{supp}(x_2)$. Then (T, L_2, c, r) is a willow. In particular, we can, in polynomial time, compute a solution of cost at most $c(x_2)$.*

Proof. By Corollary 9.8, applied with $\sigma = \sigma_{\mathrm{id}}$, we know that every link in $\overrightarrow{L} \cap \operatorname{supp}(x^{**})$ is an X-cross-link or an up- or down-link. Hence, every link in $\operatorname{supp}(x_2)$ is an up-link, a down-link, or a $(W^* \cup X)$-cross-link. To establish that (T, L_2, c, r) is a willow, it suffices to show that every vertex in $(W^* \cup X)$ is up- or down-independent with respect to L_2. For vertices in W^*, this follows from Lemma 8.13, and Propositions 8.5 and 8.9. Recall that $X = \bigcup_{C \in \mathring{\mathcal{C}}} V(C)$. We show that for $C \in \mathring{\mathcal{C}}_{up}$, all vertices in $V(C)$ are down-independent. Analogously, one can show that for $C \in \mathring{\mathcal{C}}_{down}$, all vertices in $V(C)$ are up-independent.

Let $C \in \mathring{\mathcal{C}}_{up}$. By (9.7.3), $\operatorname{supp}(x^{**}) \cap L^{\uparrow}_{r_C} = \emptyset$, so also $\operatorname{supp}(x_2) \cap L^{\uparrow}_{r_C} = \emptyset$ by Proposition 8.9. By Proposition 8.5, r_C is down-independent. Next, let $v \in V(C) \setminus \{r_C\}$. Then $a_v \in A(C)$. Assume towards a contradiction there were $\ell = (u, w) \in L_2$ with $\overrightarrow{\operatorname{cov}}(\ell) \cap A_v \cap A_{down} \neq \emptyset$ and $\overrightarrow{\operatorname{cov}}(\ell) \not\subseteq A_v$. The first property implies $u \in U_v \setminus \{v\}$, the second property tells us

that $\text{apex}(\ell)$ is a strict ancestor of v. In particular, $a_v \in \text{cov}(\ell)$. By Corollary 9.8, $\text{apex}(\ell) \in V(C)$. As $C \in \mathring{\mathcal{C}}_{up}$, a_v is an up-arc, so $a_v \in \overleftarrow{\text{cov}}(\ell)$, implying $\ell \in \overleftarrow{L}$. But this implies that $\text{apex}(\ell) = w$ is an endpoint of ℓ because all links in $\overleftarrow{L}$ were split at their apices. Hence, $\text{cov}(\ell) \setminus A_v$ consists of the up-arcs on the v-$\text{apex}(\ell)$-path, implying $\text{cov}(\ell) \setminus A_v \subseteq \overleftarrow{\text{cov}}(\ell)$ and $\overrightarrow{\text{cov}}(\ell) \subseteq A_v$, contradicting our assumptions. □

Before describing the splitting leading to our third solution, we make the following observation:

LEMMA 10.8. *Let $\ell \in \overrightarrow{L} \cap \text{supp}(x^{**})$. Then there exists a unique core $C \in \mathring{\mathcal{C}}$ such that $\overrightarrow{\text{cov}}(\ell) \cap A(C) \neq \emptyset$.*

Proof. By definition of $\overrightarrow{L}$, there exists at least one core with this property. Assume towards a contradiction that there are two distinct cores C_1 and C_2 such that $\overrightarrow{\text{cov}}(\ell) \cap A(C_1) \neq \emptyset$ and $\overrightarrow{\text{cov}}(\ell) \cap A(C_2) \neq \emptyset$. Let $w := \text{apex}(\ell)$. By Corollary 9.8, we know that $w \in V(C_1) \cap V(C_2)$. If $w = r$, then $w = r_{C_1} = r_{C_2}$. Otherwise, if $w \neq r$, a_w is contained in at most one of the sets $A(C_1)$ or $A(C_2)$, assume w.l.o.g. that $a_w \notin A(C_1)$. As $w \in V(C_1)$, but $a_w \notin A(C_1)$, we must, again, have $w = r_{C_1}$.

Let $a \in \overrightarrow{\text{cov}}(\ell) \cap A(C_1)$ and let x be the bottom vertex of a. As $a \in \overrightarrow{\text{cov}}(\ell)$ and $\text{apex}(\ell) = w = r_{C_1}$, ℓ covers all arcs on the x-r_{C_1}-path in C_1, including the arc incident to r_{C_1}. By (9.7.4), $r_{C_1} = \text{apex}(\ell)$ is an endpoint of ℓ, so ℓ is an up-link or a down-link. As $\overrightarrow{\text{cov}}(\ell) \cap A(C_1) \neq \emptyset$ and $\overrightarrow{\text{cov}}(\ell) \cap A(C_2) \neq \emptyset$, C_1 and C_2 must both be down-cores, or both be up-cores, respectively. However, this contradicts $w \in V(C_1) \cap V(C_2)$ because two distinct down-cores/ up-cores are vertex-disjoint. □

For $\ell \in \overrightarrow{L} \cap \text{supp}(x^{**})$, let C_ℓ be the unique core with $\overrightarrow{\text{cov}}(\ell) \cap A(C) \neq \emptyset$. We define a splitting σ_3' of L as follows:

- Let $\ell = (u, v) \in \overrightarrow{L} \cap \text{supp}(x^{**})$ with $C_\ell \in \mathring{\mathcal{C}}_{up}$. Let w be the lowest vertex from $V(C_\ell)$ on the $\text{apex}(\ell)$-v-path in T and define $\sigma_3'(\ell) := \{(u, \text{apex}(\ell)), (\text{apex}(\ell), w), (w, v)\}$.
- Let $\ell = (u, v) \in \overrightarrow{L} \cap \text{supp}(x^{**})$ with $C_\ell \in \mathring{\mathcal{C}}_{down}$. Let w be the lowest vertex from $V(C_\ell)$ on the u-$\text{apex}(\ell)$-path in T and define $\sigma_3'(\ell) := \{(u, w), (w, \text{apex}(\ell)), (\text{apex}(\ell), v)\}$.
- Let $\ell = (u, v) \in L_{cross} \cup L_{rest}$. Define $\sigma_3'(\ell) = \{(u, \text{apex}(\ell)), (\text{apex}(\ell), v)\}$.
- For every other link ℓ, define $\sigma_3'(\ell) = \{\ell\}$.

LEMMA 10.9. *Let $\sigma_3 := \sigma_3' \circ \sigma^{**} \circ \sigma^*$ and let $x_3 := \text{split}(x^{**}, \sigma_3') = \text{split}(x, \sigma_3)$. Let $L_3 := \text{supp}(x_3)$ and let $M \in \{0,1\}^{A \times L_3}$ denote the arc-link-coverage matrix of (T, L_3, c, r). Then M is TU. In particular, we can, in polynomial time, compute a solution to (T, L_3, c, r) of cost at most $c(x_3)$.*

Before we prove Lemma 10.9, we first make the following observations:

PROPOSITION 10.10. *Let $\ell \in L_3$ such that ℓ is neither an up- nor a down-link. Then $\ell \in \overleftarrow{L} \cap \text{supp}(x^{**})$.*

Proof. As $\ell \in L_3$, there is $\ell' \in \text{supp}(x^{**})$ with $\ell \in \sigma_3'(\ell')$. For every link $\ell'' \in \text{supp}(x^{**}) \cap (\overrightarrow{L} \cup L_{cross} \cup L_{rest})$, $\sigma_3'(\ell'')$ consists of up- and down-links only. Hence, $\ell' \in \overleftarrow{L} \cap \text{supp}(x^{**})$. As $\sigma_3'(\ell') = \{\ell'\}$, we have $\ell = \ell'$. □

PROPOSITION 10.11. *Let $\ell \in L_3 \cap \overrightarrow{L}$. Then there exists a unique core $C \in \mathring{\mathcal{C}}$ such that $\overrightarrow{\text{cov}}(\ell) \cap A(C) \neq \emptyset$. Moreover, $\overrightarrow{\text{cov}}(\ell) \subseteq A(C)$ and ℓ is an up- or down-link.*

Proof. Let $\ell \in L_3 \cap \overrightarrow{L}$ and let C be a core with $\overrightarrow{\text{cov}}(\ell) \cap A(C) \neq \emptyset$. Let $\ell' \in \text{supp}(x^{**})$ such that $\ell \in \sigma_3'(\ell')$. Then ℓ is a shadow of ℓ', so $\overrightarrow{\text{cov}}(\ell) \subseteq \overrightarrow{\text{cov}}(\ell')$. In particular, $\overrightarrow{\text{cov}}(\ell') \cap A(C) \neq \emptyset$, $\ell' \in \overrightarrow{L}$ and by Lemma 10.8, $C = C_{\ell'}$ is the unique core C' such that $\overrightarrow{\text{cov}}(\ell') \cap A(C') \neq \emptyset$. Hence, C is also the unique core C' such that $\overrightarrow{\text{cov}}(\ell) \cap A(C') \neq \emptyset$. Let w be as in the definition of $\sigma_3'(\ell')$. The second part of the statement follows from the facts that $\text{apex}(\ell') \in V(C)$ by Corollary 9.8 and that we must have $\ell = (\text{apex}(\ell'), w)$, if C is an up-core, and $\ell = (w, \text{apex}(\ell'))$, if C is a down-core, because this is the only link in $\sigma_3'(\ell')$ covering part of $A(C)$. □

We further introduce the following notation.

DEFINITION 10.12. *We call a matrix $M \in \mathbb{R}^{I \times J}$ a block diagonal matrix with blocks $M[I_s, J_s]$, $s = 1, \dots, t$ if $I = \dot{\bigcup}_{s=1}^{t} I_s$ is a partition of I, $J = \dot{\bigcup}_{s=1}^{t} J_s$ is a partition of J and for $i \in I_{s_1}$ and $j \in J_{s_2}$, $M_{ij} \neq 0$ implies $s_1 = s_2$, i.e., non-zero entries can only occur within one block.*

This definition may differ from notions used in the literature in that we do not require the blocks to be square or have the same size.

Proof of Lemma 10.9. We first establish the following claim:

Claim 10.13. *M is a block diagonal matrix with blocks $M[A(C), \{\ell \in L_3 \colon \overrightarrow{\mathrm{cov}}(\ell) \cap A(C) \neq \emptyset\}]$ for $C \in \mathring{\mathcal{C}}$ and $M[A \setminus \bigcup_{C \in \mathring{\mathcal{C}}} A(C), L_3 \setminus \overrightarrow{L}]$.*

Proof of claim. By definition, the sets $A(C)_{C \in \mathring{\mathcal{C}}}$ are pairwise disjoint. This shows that the arcs sets indexing the rows of the blocks form a partition of A.

By Proposition 10.11, the link sets indexing the columns of the blocks form a partition of L_3.

Finally, we verify the block structure of M. First, let $C \in \mathring{\mathcal{C}}$, let $a \in A(C)$ and let $\ell' \in L_3$ such that $M_{a,\ell'} \neq 0$. Then $a \in \overrightarrow{\mathrm{cov}}(\ell')$, so $\ell' \in \{\ell \in L_3 \colon \overrightarrow{\mathrm{cov}}(\ell) \cap A(C) \neq \emptyset\}$.

Next, let $a \in A \setminus \bigcup_{C \in \mathring{\mathcal{C}}} A(C)$. By Proposition 10.11, we have $a \notin \overrightarrow{\mathrm{cov}}(\ell)$, and, hence, $M_{a,\ell} = 0$ for every $\ell \in L_3 \cap \overrightarrow{L}$. Thus, if $M_{a,\ell} \neq 0$ for some $\ell \in L_3$, then $\ell \in L_3 \setminus \overrightarrow{L}$. ◻

To show that M is TU, it suffices to establish total unimodularity of each of the blocks separately. For the blocks of the form $M[A(C), \{\ell \in L_3 \colon \overrightarrow{\mathrm{cov}}(\ell) \cap A(C) \neq \emptyset\}]$ with $C \in \mathring{\mathcal{C}}$, this follows from Theorem 5.3 and the fact that $\{\ell \in L_3 \colon \overrightarrow{\mathrm{cov}}(\ell) \cap A(C) \neq \emptyset\}$ only consists of up- or down-links by Proposition 10.11.

The block $M[A \setminus \bigcup_{C \in \mathring{\mathcal{C}}} A(C), L_3 \setminus \overrightarrow{L}]$ corresponds to the instance (T', L', c', r') obtained from the tuple $(T, L_3 \setminus \overrightarrow{L}, c, r)$ by contracting all arcs in $\bigcup_{C \in \mathring{\mathcal{C}}} A(C)$. Each (super-)vertex $v \in V(T')$ corresponds to a set $Y_v \subseteq V(T)$ of original vertices. Let $R \coloneqq \{v \in V(T') \colon \exists C \in \mathring{\mathcal{C}} \colon r_C \in Y_v\}$.

Claim 10.14. *Let $\ell' \in L'$ such that ℓ' is neither an up- nor a down-link. Then ℓ' is an R-cross-link.*

Proof of claim. By Proposition 10.10, every such link $\ell' \in L'$ corresponds to a link $\ell \in \overleftarrow{L} \cap \mathrm{supp}(x^{**})$. By definition of $\overleftarrow{L}$, let $C \in \mathring{\mathcal{C}}$ such that $\overleftarrow{\mathrm{cov}}(\ell) \cap A(C) \neq \emptyset$. By Corollary 9.8, $\mathrm{apex}(\ell) \in V(C)$ and all of $V(C)$, including r_C, is contracted into the same super-vertex, which becomes $\mathrm{apex}(\ell')$. ◻

We show that every vertex in R is up- or down-independent, establishing that (T', L', c', r') is a willow and concluding the proof by Theorem 5.3.

Let $v \in R$. Y_v is the vertex set of a connected subgraph of T, consisting of a collection of cores. In particular, there is a unique vertex in Y_v that is closest to the root r of T, and it is the root r_C of a core C. We may assume without loss of generality that $C \in \mathring{\mathcal{C}}_{up}$; the case where $C \in \mathring{\mathcal{C}}_{down}$ can be handled analogously. We show that no link in L' points out of T'_v, establishing that v is down-independent by Proposition 8.5. Assume towards a contradiction there were $\ell' = (u', w') \in L'$ pointing out of T'_v, i.e., $u' \in V(T'_v) \setminus \{v\}$ and $w' \in V(T') \setminus V(T'_v)$. Let ℓ' correspond to the link $\ell = (u, w) \in L_3 \setminus \overrightarrow{L}$ with $u \in Y_{u'}$ and $w \in Y_{w'}$. As $Y_{u'}$, Y_v and $Y_{w'}$ are vertex sets of connected, vertex-disjoint subgraphs of the tree T and r_C is the vertex of Y_v closest to the root of T, $Y_{u'} \subseteq V(T_{r_C}) \setminus \{r_C\}$ and $Y_{w'} \subseteq V(T) \setminus V(T_{r_C})$. Hence, ℓ points out of T_{r_C}, contradicting (9.7.3). ◻

We are now ready to finally prove Theorem 7.12, which we restate for convenience.

Theorem 7.12. *Let $\bar{\varepsilon}, \Delta > 0$. We can compute a constant $k(\bar{\varepsilon}, \Delta)$ with the following property: Given a rooted instance $(\bar{T}, \bar{L}, \bar{c}, \bar{r})$ of WDTAP with cost ratio at most Δ and a feasible solution $\bar{x}$ to* (2.1)*, we can, in polynomial time, either find a solution $S \subseteq \bar{L}$ with $\bar{c}(S) \leq (1.75 + \bar{\varepsilon}) \cdot \bar{c}(\bar{x})$, or find a visibly $k(\bar{\varepsilon}, \Delta)$-wide modification inequality that is violated by $\bar{x}$.*

Proof. Let $\varepsilon \coloneqq \frac{\min\{1, \bar{\varepsilon}\}}{7}$. Define the constants γ, ζ_1, ζ_2 and k as in Section 8 and let $k(\bar{\varepsilon}, \Delta) \coloneqq k$. Let (T, L, c, r) and x arise from $(\bar{T}, \bar{L}, \bar{c}, \bar{r})$ and $\bar{x}$ by contracting the set $\bar{A}$ of ζ_1-covered arcs. Note that x is a solution to (2.1) of cost $c(x) \leq \bar{c}(\bar{x})$ satisfying (8.6). We apply Theorem 8.3 and Theorem 9.7 to obtain splittings σ^* and σ^{**} and solutions x^* and x^{**} to (2.1) for (T, L, c, r). We define L_{cross}, $\overrightarrow{L}$, $\overleftarrow{L}$ and L_{rest} as in the beginning of this section. We apply Lemma 10.5 to, in polynomial time, either compute a solution S_1 to (T, L, c, r) of cost at most $c(x_1)$ or a violated visibly k-wide modification inequality for (T, L, c, r). In the latter case, the corresponding splitting, together with $\bar{A}$, gives rise to a violated visibly k-wide modification inequality for $(\bar{T}, \bar{L}, \bar{c}, \bar{r})$ and $\bar{x}$. Hence, we may assume in the following that we have found a solution S_1 to (T, L, c, r) of cost $c(S_1) \leq c(x_1)$. We further apply Lemmas 10.7 and 10.9 to, in polynomial time, compute solutions S_2 and S_3 to (T, L, c, r) of cost $c(S_2) \leq c(x_2)$ and $c(S_3) \leq c(x_3)$, respectively.

For $L' \in \{L_{cross}, \overleftarrow{L}, \overrightarrow{L}, L_{rest}\}$, we define $C^{**}(L') \coloneqq \sum_{\ell \in L} c(\ell) \cdot x^{**}(\ell)$. By (10.1), we know that

$$c(x^{**}) = C^{**}(L_{cross}) + C^{**}(\overleftarrow{L}) + C^{**}(\overrightarrow{L}) + C^{**}(L_{rest}).$$

By Proposition 7.9 and because $|\sigma_{L\setminus L_{rest}}(\ell)| = 2$ for $\ell \in L \setminus L_{rest}$ and $|\sigma_{L\setminus L_{rest}}(\ell)| = 1$ for $\ell \in L_{rest}$, we have

$$c(x_1) \leq c(x^{**}) + C^{**}(L_{cross}) + C^{**}(\overleftarrow{L}) + C^{**}(\overrightarrow{L}).$$

By Proposition 7.9 and because $|\sigma_{\overleftarrow{L}\cup L_{rest}}(\ell)| = 2$ for $\ell \in \overleftarrow{L} \cup L_{rest}$ and $|\sigma_{\overleftarrow{L}\cup L_{rest}}(\ell)| = 1$ for $\ell \notin \overleftarrow{L} \cup L_{rest}$, we have

$$c(x_2) \leq c(x^{**}) + C^{**}(\overleftarrow{L}) + C^{**}(L_{rest}).$$

By Proposition 7.9 and because $|\sigma'_3(\ell)| = 3$ for $\ell \in \overrightarrow{L} \cap \mathrm{supp}(x^{**})$, $|\sigma'_3(\ell)| = 2$ for $\ell \in L_{cross} \cup L_{rest}$ and $|\sigma'_3(\ell)| = 1$ for every other link ℓ, we have

$$c(x_3) \leq c(x^{**}) + C^{**}(L_{cross}) + 2 \cdot C^{**}(\overrightarrow{L}) + C^{**}(L_{rest}).$$

Let S^* be the best one among the three solutions S_1, S_2 and S_3. Then

$$\begin{aligned} c(S^*) \leq & \frac{1}{4} \cdot c(x_1) + \frac{1}{2} \cdot c(x_2) + \frac{1}{4} \cdot c(x_3) \leq c(x^{**}) + \frac{1}{2} \cdot C^{**}(L_{cross}) + \frac{3}{4} \cdot C^{**}(\overleftarrow{L}) + \frac{3}{4} \cdot C^{**}(\overrightarrow{L}) + \frac{3}{4} \cdot C^{**}(L_{rest}) \\ & \leq \frac{7}{4} \cdot c(x^{**}) \leq \frac{7}{4} \cdot (1+\varepsilon)^2 \cdot c(x) \leq \frac{7}{4} \cdot (1+\varepsilon)^2 \cdot \bar{c}(\bar{x}), \end{aligned}$$

where the second-to-last inequality follows from (9.7.1). By Lemma 8.2, we can extend S^* to a solution S for $(\bar{T}, \bar{L}, \bar{c})$ of cost at most

$$\left(\frac{7}{4} \cdot (1+\varepsilon)^2 + \varepsilon\right) \cdot \bar{c}(\bar{x}) \leq \frac{7}{4} \cdot (1 + 3 \cdot \varepsilon + \varepsilon^2) \cdot \bar{c}(\bar{x}) \leq \frac{7}{4} \cdot (1 + 4 \cdot \varepsilon) \cdot \bar{c}(\bar{x}) = \left(\frac{7}{4} + \bar{\varepsilon}\right) \cdot \bar{c}(\bar{x}).$$

Observing that $\frac{7}{4} = 1.75$ concludes the proof. ∎

Acknowledgments This work originated from a collaboration that included Siyue Liu and R. Ravi, whose early contributions are gratefully acknowledged. We also thank the anonymous reviewers who helped improve the presentation of our paper. This work was supported in part by EPSRC grant EP/X030989/1.

Data Availability Statement No data are associated with this article. Data sharing is not applicable to this article.

References

[1] D. Adjiashvili, *Beating approximation factor two for weighted tree augmentation with bounded costs*, ACM Trans. Algorithms, 15 (2018), https://doi.org/10.1145/3182395.

[2] A. Ahmadi, I. Gholami, M. Hajiaghayi, P. Jabbarzade, and M. Mahdavi, *Breaking a long-standing barrier: 2-ε approximation for Steiner forest*, 2025, https://arxiv.org/abs/2504.11398.

[3] M. Bosch-Calvo, M. Garg, F. Grandoni, F. Hommelsheim, A. J. Ameli, and A. Lindermayr, *A 5/4-approximation for two-edge connectivity*, in Proceedings of the 57th Annual ACM Symposium on Theory of Computing, STOC 2025, Prague, Czechia, June 23-27, 2025, M. Koucký and N. Bansal, eds., ACM, 2025, pp. 653–664, https://doi.org/10.1145/3717823.3718275, https://doi.org/10.1145/3717823.3718275.

[4] F. Cecchetto, V. Traub, and R. Zenklusen, *Bridging the gap between tree and connectivity augmentation: unified and stronger approaches*, in STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021, S. Khuller and V. V. Williams, eds., ACM, 2021, pp. 370–383, https://doi.org/10.1145/3406325.3451086.

[5] M. Charikar, C. Chekuri, T. Cheung, Z. Dai, A. Goel, S. Guha, and M. Li, *Approximation algorithms for directed Steiner problems*, in Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, 25-27 January 1998, San Francisco, California, USA, H. J. Karloff, ed., ACM/SIAM, 1998, pp. 192–200, http://dl.acm.org/citation.cfm?id=314613.314700.

[6] J. Cheriyan and Z. Gao, *Approximating (unweighted) tree augmentation via lift-and-project, part I: stemless TAP*, Algorithmica, 80 (2018), pp. 530–559, https://doi.org/10.1007/S00453-016-0270-4.

[7] J. Cheriyan and Z. Gao, *Approximating (unweighted) tree augmentation via lift-and-project, part II*, Algorithmica, 80 (2018), pp. 608–651, https://doi.org/10.1007/S00453-017-0275-7.

[8] J. Edmonds and R. Giles, *A min-max relation for submodular functions on graphs*, in Studies in Integer Programming, P. Hammer, E. Johnson, B. Korte, and G. Nemhauser, eds., vol. 1 of Annals of Discrete Mathematics, Elsevier, 1977, pp. 185–204, https://doi.org/10.1016/S0167-5060(08)70734-9.

[9] 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 Trans. Algorithms, 5 (2009), pp. 21:1–21:17, https://doi.org/10.1145/1497290.1497297.

[10] S. Fiorini, M. Groß, J. Könemann, and L. Sanità, *Approximating weighted tree augmentation via Chvátal-Gomory cuts*, in Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018, A. Czumaj, ed., SIAM, 2018, pp. 817–831, https://doi.org/10.1137/1.9781611975031.53.

[11] G. N. Frederickson and J. F. JáJá, *Approximation algorithms for several graph augmentation problems*, SIAM J. Comput., 10 (1981), pp. 270–283, https://doi.org/10.1137/0210019.

[12] A. Ghouila-Houri, *Caractérisation des matrices totalement unimodulaires*, Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences (Paris), 254 (1962), pp. 1192–1194.

[13] F. Grandoni, C. Kalaitzis, and R. Zenklusen, *Improved approximation for tree augmentation: saving by rewiring*, in Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, New York, NY, USA, 2018, Association for Computing Machinery, p. 632–645, https://doi.org/10.1145/3188745.3188898.

[14] D. Hathcock and M. Zlatin, *Approximation algorithms for steiner connectivity augmentation*, in 32nd Annual European Symposium on Algorithms, ESA 2024, September 2-4, 2024, Royal Holloway, London, United Kingdom, T. M. Chan, J. Fischer, J. Iacono, and G. Herman, eds., vol. 308 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024, pp. 67:1–67:16, https://doi.org/10.4230/LIPICS.ESA.2024.67.

[15] F. Hommelsheim, A. Lindermayr, and Z. Liu, *A better-than-*$5/4$*-approximation for two-edge connectivity*, arXiv preprint arXiv:2509.19655, (2025).

[16] K. Jain, *Factor 2 approximation algorithm for the generalized steiner network problem*, in 39th Annual Symposium on Foundations of Computer Science, FOCS '98, November 8-11, 1998, Palo Alto, California, USA, IEEE Computer Society, 1998, pp. 448–457, https://doi.org/10.1109/SFCS.1998.743495.

[17] K. V. Klinkby, P. Misra, and S. Saurabh, *Strong connectivity augmentation is FPT*, in Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021, D. Marx, ed., SIAM, 2021, pp. 219–234, https://doi.org/10.1137/1.9781611976465.15.

[18] G. Kortsarz, R. Krauthgamer, and J. R. Lee, *Hardness of approximation for vertex-connectivity network design problems*, SIAM J. Comput., 33 (2004), pp. 704–720, https://doi.org/10.1137/S0097539702416736.

[19] G. Kortsarz and Z. Nutov, *A simplified 1.5-approximation algorithm for augmenting edge-connectivity of a graph from 1 to 2*, ACM Trans. Algorithms, 12 (2016), pp. 23:1–23:20, https://doi.org/10.1145/2786981.

[20] H. Nagamochi, *An approximation for finding a smallest 2-edge-connected subgraph containing a specified spanning tree*, Discret. Appl. Math., 126 (2003), pp. 83–113, https://doi.org/10.1016/S0166-218X(02)00218-4.

[21] R. Ravi, W. Zhang, and M. Zlatin, *Approximation algorithms for steiner tree augmentation problems*, in Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023, N. Bansal and V. Nagarajan, eds., SIAM, 2023, pp. 2429–2448, https://doi.org/10.1137/1.9781611977554.CH94.

[22] A. Schrijver, *Min-max relations for directed graphs*, in North-Holland Mathematics Studies, vol. 66, Elsevier, 1982, pp. 261–280.

[23] V. Traub and R. Zenklusen, *A better-than-2 approximation for weighted tree augmentation*, in 62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022, IEEE, 2021, pp. 1–12, https://doi.org/10.1109/FOCS52979.2021.00010.

[24] V. Traub and R. Zenklusen, *Local search for weighted tree augmentation and Steiner tree*, in Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022, J. S. Naor and N. Buchbinder, eds., SIAM, 2022, pp. 3253–3272, https://doi.org/10.1137/1.9781611977073.128.

A Appendix

A.1 Multi 2-TAP reduces to DTAP In this section we prove that the weighted multi 2-TAP problem reduces to WDTAP. In weighted multi 2-TAP, we are given an undirected tree $T = (V, E)$, and a set of links $L \subseteq \binom{V}{2}$ with positive costs. For every $e \in E$, denote by $S_e \subseteq V$ a shore of the fundamental tree cut induced by e. A multiset (repetition allowed) of undirected links $F \subseteq L$ is a *k-covering* of T if every fundamental cut is covered at least k times by F, that is $|\delta_F(S_e)| \geq k$ for all $e \in E$. The cost of a multi-set of links is the sum of the costs of links in that set, weighted by multiplicity.

Problem A.1 (Weighted Multi 2-TAP). *Given an undirected tree $T = (V, E)$ and a collection of links $L \subseteq \binom{V}{2}$ with positive costs, find the cheapest 2-covering of T.*

The reduction is easiest to explain by introducing an intermediate problem, which we call bi-directed tree cover, and which is equivalent to WDTAP.

Problem A.2 (Bi-directed Tree Cover). *Given an undirected tree $T = (V, E)$ and a collection of directed links $L \subseteq V \times V$ with positive costs, choose a cheapest set of links F so that $|\delta^+_F(S_e)| \geq 1$, and $|\delta^-_F(S_e)| \geq 1$ for all tree edges $e \in E$.*

In other words, the cut induced by each tree edge must be crossed in both directed by the solution F. Bi-directed tree cover is easily seen to be equivalent to WDTAP: on the one hand, it can be reduced to WDTAP by subdividing every tree edge and orienting them in opposite directions. On the other hand, WDTAP can be reduced to bi-directed tree cover by adding a zero-cost directed link ℓ_a parallel to each tree arc $a \in A$ in the same direction as a, and making the tree undirected.

We now show that weighted multi 2-TAP can be reduced to bi-directed tree cover. We replace every link $\ell \in L$ by two directed links ℓ^+ and ℓ^- in opposite directions, each having the same cost as ℓ. Clearly, every bi-directed tree cover solution is a feasible solution to the weighted multi 2-TAP instance with the same cost. The following proposition shows that every weighted multi 2-TAP solution can be oriented into a bi-directed tree cover solution of the same cost.

Proposition A.3. *Given a 2-covering $F \subseteq L$ of a tree $T = (V, E)$, there is an orientation $\vec{F}$ such that $|\delta^+_{\vec{F}}(S_e)|, |\delta^-_{\vec{F}}(S_e)| \geq 1$ for every fundamental cut shore S_e.*

Proof. Let $\mathcal{S} := \{(S_e) \cup (V \setminus S_e) \mid e \in E\}$ be all the shores of fundamental cuts. Let $\vec{H}$ be an arbitrary orientation of the 2-covering F. We seek an integral solution to the following submodular flow polyhedron:

$$|\delta^+_{\vec{H}}(S)| - x(\delta^+_{\vec{H}}(S)) + x(\delta^-_{\vec{H}}(S)) \geq 1, \forall S \in \mathcal{S}.$$

This is indeed a submodular flow because $\mathcal{S}$ is cross-free and thus trivially a crossing family. We can take $x_\ell = \frac{1}{2}$ for every $\ell \in \vec{H}$ to be a fractional feasible solution, and thus by the integrality of the submodular flow polyhedron, there is an integral feasible solution x. Flipping ℓ if and only if $x_\ell = 1$, yields the desired orientation $\vec{F}$. ∎

A.2 Hardness of DTAP In this section, we prove the following result on the hardness of DTAP.

Proposition A.4. *DTAP is NP-hard and APX-hard, even in the unweighted setting.*

Proof. We prove NP-hardness using the same reduction as the one for CSTA in [11]. We reduce 3-dimensional matching (3DM) to unweighted DTAP.

Let $M \subseteq W \times X \times Y$ be an instance of 3DM with $|M| = p$, $W = \{w_i \mid i = 1, ..., q\}$, $X = \{x_i \mid i = 1, ..., q\}$, $Y = \{y_i \mid i = 1, ..., q\}$. We define an instance of DTAP as follows. Let $V = \{r\} \cup \{w_i, x_i, y_i \mid i = 1, ..., q\} \cup \{a_{ijk}, a'_{ijk} \mid (w_i, x_j, y_k) \in M\}$.

$A_T = \{(r, x_i), (r, w_i), (y_i, r) \mid i = 1, ..., q\} \cup \{(a_{ijk}, w_i), (w_i, a'_{ijk}) \mid (w_i, x_j, y_k) \in M\}$.

$A_L = \{(x_j, a_{ijk}), (a'_{ijk}, a_{ijk}), (a'_{ijk}, y_k) \mid (w_i, x_j, y_k) \in M\}$.

We claim that there exists a 3DM of size q if and only if the minimum size of a DTAP solution is $p + q$. Indeed, notice that there are $2p + 2q$ leaves in T, where each leaf needs at least one link to cover it. Thus, the minimum size of a DTAP solution is at least $p + q$. On the one hand, if there is a 3DM M' of size q, we obtain a DTAP solution $L' := \{(x_j, a_{ijk}), (a'_{ijk}, y_k) \mid (w_i, x_j, y_k) \in M'\} \cup \{(a'_{ijk}, a_{ijk}) \mid (w_i, x_j, y_k) \in M \setminus M'\}$ whose size is $2q + (p - q) = p + q$. On the other hand, if there is a DTAP solution L' of size $p + q$, by the previous argument, L' forms a perfect matching on the leaves. Let M' be the edges $(w_i, x_j, y_k) \in M$ such that a_{ijk} is matched to x_j. Clearly, a_{ijk} is matched to x_j if and

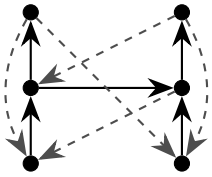
only if a'_{ijk} is matched to y_k. Thus, $|M'| = q$ and it intersects every node in X or Y exactly once. Since for every i, (r, w_i) is covered by the links from a'_{ijk} to y_k, M' intersects every node in W, and thus intersects every node in W exactly once. Therefore, M' is indeed a 3DM of size q. □

Following the methods in [18], the above proof can be extended to show APX-hardness via a reduction from bounded degree 3DM.

A.3 Lower Bound on the Integrality Gap of DTAP We show the following lower bound on the integrality gap of the natural set covering relaxation for DTAP.

PROPOSITION A.5. *The integrality gap of the set covering formulation for DTAP given in (2.1) is at least* $\frac{6}{5}$.

Proof. Consider the following unweighted DTAP instance whose constraint matrix corresponds to a 5-cycle:



Choosing $x_\ell = \frac{1}{2}$ for all $\ell \in L$ yields a solution of cost $\frac{5}{2}$, while the smallest integral solution has cost 3.