Approximation Algorithms for Steiner Connectivity Augmentation

Daniel Hathcock
Carnegie Mellon University, Pittsburgh, PA, USA

Michael Zlatin
Carnegie Mellon University, Pittsburgh, PA, USA

Abstract

We consider connectivity augmentation problems in the Steiner setting, where the goal is to augment the edge-connectivity between a specified subset of terminal nodes.

In the Steiner Augmentation of a Graph problem (k-SAG), we are given a k-edge-connected subgraph H of a graph G. The goal is to augment H by including links from G of minimum cost so that the edge-connectivity between nodes of H increases by 1. This is a generalization of the Weighted Connectivity Augmentation Problem, in which only links between pairs of nodes in H are available for the augmentation.

In the Steiner Connectivity Augmentation Problem (k-SCAP), we are given a Steiner k-edge-connected graph connecting terminals R, and we seek to add links of minimum cost to create a Steiner $(k+1)$-edge-connected graph for R. Note that k-SAG is a special case of k-SCAP.

The results of Ravi, Zhang and Zlatin for the Steiner Tree Augmentation problem yield a $(1.5+\varepsilon)$-approximation for 1-SCAP and for k-SAG when k is odd [20]. In this work, we give a $(1+\ln 2+\varepsilon)$-approximation for the Steiner Ring Augmentation Problem (SRAP). This yields a polynomial time algorithm with approximation ratio $(1+\ln 2+\varepsilon)$ for 2-SCAP. We obtain an improved approximation guarantee for SRAP when the ring consists of only terminals, yielding a $(1.5+\varepsilon)$-approximation for k-SAG for any k.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Approximation Algorithms, Steiner Connectivity, Network Design

Digital Object Identifier 10.4230/LIPIcs.ESA.2024.67

Related Version *Full Version*: `https://arxiv.org/abs/2308.08690`

Funding *Daniel Hathcock*: Supported by the NSF Graduate Research Fellowship grant DGE-2140739.

1 Introduction

1.1 Background

The edge-connectivity of a graph is a common measure of the robustness of the network to edge failures. If a network is k-edge-connected, then it can sustain failures of up to $k-1$ edges without being disconnected. Many problems in the area of network design seek to construct a cheap network which satisfies edge-connectivity requirements between certain pairs of nodes.

This has given rise to many fundamental problems of interest in combinatorial optimization and approximation algorithms. One notable example is the Survivable Network Design Problem (SNDP). In SNDP, we are given a graph with non-negative costs on edges and a connectivity requirement r_{ij} for each pair of vertices $i, j \in V$. The goal is to find a cheapest subgraph of G so that there are r_{ij} pairwise edge-disjoint paths between all pairs of vertices i and j.

Jain [15] gave a polynomial time algorithm for SNDP based on iterative rounding, which achieves an approximation factor of 2. Despite its generality, this algorithm achieves the best-known approximation ratio for a variety of network design problems of particular

32nd Annual European Symposium on Algorithms (ESA 2024).
Editors: Timothy Chan, Johannes Fischer, John Iacono, and Grzegorz Herman; Article No. 67; pp. 67:1–67:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

interest. For example, when $r_{ij} = k$ for all $i, j \in V$, we obtain the weighted minimum cost k-edge-connected spanning subgraph problem, for which no better-than-2 approximation algorithm is known, even when $k = 2$. However, for some special cases, such as the minimum Steiner tree problem, specialized algorithms have been developed to improve upon this ratio. Hence, a major question in the field of network design is: for which network connectivity problems can we achieve an approximation factor below 2?

Recently, there has been major progress towards addressing this question. Consider the special case of SNDP known as the Weighted Connectivity Augmentation Problem (WCAP), in which we seek to increase the edge-connectivity of a given k-edge-connected graph by 1 by adding to the graph a collection of links of minimum cost.

A solution to WCAP must include a link crossing each of the min-cuts of the given graph H. Since the minimum cuts of any graph can be represented by a cactus [9], it is enough to consider the WCAP problem for $k = 2$ and when H is a cactus.[1] In fact, by the addition of zero cost links, the weighted problem further reduces to the case where H is a cycle (the so-called Weighted Ring Augmentation Problem, WRAP) [11]. If k is odd, WCAP reduces to the case where $k = 1$ and H is a tree, yielding the Weighted Tree Augmentation Problem (WTAP).

In a breakthrough result, Traub and Zenklusen [23] employed a relative greedy algorithm using ideas from [6] to give the first approximation algorithm with approximation ratio better than 2 for WTAP. In particular, they achieved an approximation ratio of $(1 + \ln 2 + \varepsilon)$ for WTAP, thereby yielding the same result for WCAP when k is odd. Shortly afterward, they improved upon this result, bringing the approximation factor down to $(1.5 + \varepsilon)$ for WTAP [24] using a local search algorithm. Finally, they turned to the general case of WCAP, adapting the local search method that was used for WTAP to give a $(1.5 + \varepsilon)$-approximation algorithm for the Weighted Ring Augmentation Problem [22]. As discussed above, this gives a $(1.5 + \varepsilon)$-approximation for general WCAP, adding it to the meager list of NP-hard special cases of SNDP which we can approximate to within a factor less than 2.

This brings us to our setting. Note that for the above augmentation problems, the parameter of interest is the *global* edge-connectivity of the graph. However, in many applications, we are not interested in connectivity between all nodes in the graph, but rather only the connectivity between a specified subset of "important" nodes called terminals.

The reasons to be interested in Steiner connectivity are twofold. The first is that utilizing Steiner vertices outside a given network may allow us to augment it more cheaply. For example, in Figure 1, we seek to augment the edge-connectivity of the nodes in a tree from 1 to 2. The ability to use a Steiner vertex outside the tree results in a cheaper augmentation.

The second reason is that we may want to augment the resiliency of a network which already incorporates Steiner vertices. However, we only desire to increase the edge-connectivity between its terminals. This scenario is likely in practice since, as we have seen, using Steiner nodes can establish connectivity while maintaining low costs. Naively, attempting to accomplish this with an algorithm for global connectivity augmentation may result in a needlessly expensive solution. See Figure 2 for an illustration.

In this paper, we develop algorithms for generalizations of WCAP to this Steiner setting. We first consider the Steiner Augmentation of a Graph problem (SAG). In contrast to WCAP, in which we can only add links between vertices of the graph to be augmented, in the SAG problem we can purchase links that are incident to Steiner nodes outside the given graph.

[1] A cactus is a connected graph in which every edge is included in exactly 1 cycle.

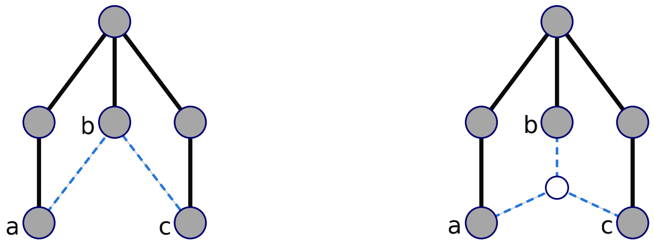

Figure 1 We seek to augment the edge-connectivity between the grey nodes (terminals) in the given tree from 1 to 2. Nodes a, b and c are arranged in an equilateral unit triangle. If only direct links are allowed, the cheapest augmentation has cost 2, but utilizing a Steiner node in the center of this triangle allows us to establish 2-edge-connectivity between the terminals at a cost of $\sqrt{3} \approx 1.73$.

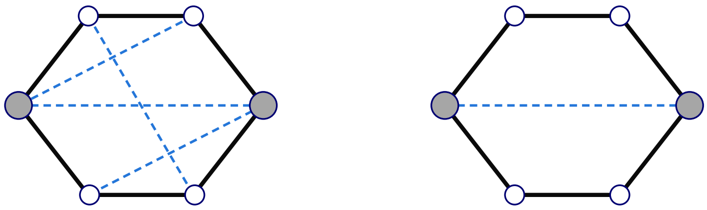

Figure 2 The given graph has 2-edge-connectivity between the large grey nodes (the terminals), which we seek to increase to 3-edge-connectivity. Solving this with global connectivity augmentation results in a solution with 4 links when only 1 is required.

▶ **Problem 1.1** (k-Steiner Augmentation of a Graph)**.** *We are given a k-edge-connected graph $H = (R, E)$, which is a subgraph of $G = (V, E \dot\cup L)$. The links L have non-negative costs $c : L \to \mathbb{R}_{\geq 0}$.*

The goal is to select $S \subseteq L$ of cheapest cost so that the graph $H' = (V, E \cup S)$ has $k+1$ pairwise edge-disjoint paths between u and v for all $u, v \in R$.

More generally, we define the Steiner Connectivity Augmentation Problem (SCAP). The goal is to augment the edge-connectivity between a specified subset of terminals in a graph from k to $k+1$ in the cheapest way.

▶ **Problem 1.2** (k-Steiner Connectivity Augmentation Problem)**.** *We are given a graph $G = (V, E \dot\cup L)$ and a subset of terminals $R \subseteq V$ such that $H := (V, E)$ has k edge-disjoint paths between every pair of vertices in R. We are also given a cost function $c : L \to \mathbb{R}_{\geq 0}$.*

The goal is to select $S \subseteq L$ of cheapest cost so that the graph $(V, E \cup S)$ has $k+1$ pairwise edge-disjoint paths between all pairs of nodes in R.

See Figure 3 for example instances of SAG and SCAP.

Notice that k-SAG is a special case of k-SCAP. In [20], Ravi, Zhang and Zlatin achieve a $(1.5 + \varepsilon)$-approximation for the problem of cheaply augmenting a given Steiner tree to be a 2-edge-connected Steiner subgraph. This is called the Steiner Tree Augmentation Problem (STAP), and is equivalent to 1-SAG. For the same reason that odd connectivity augmentation reduces to WTAP, this yields a $1.5 + \varepsilon$ approximation for k-SAG for all odd k.

Interestingly, this also yields an improved approximation for 1-SCAP. This is because the k-SAG and k-SCAP problems are equivalent when $k = 1$, as any minimal 2-edge-connected Steiner subgraph is also globally 2-edge-connected. However, this unification ceases for $k > 1$, making the higher connectivity setting that we consider much more interesting.

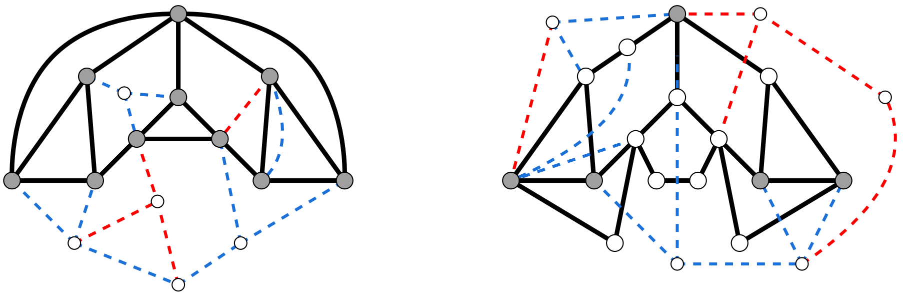

Figure 3 A 3-SAG instance is shown on the left, and a 3-SCAP instance is shown on the right. The shaded nodes are terminals R, the black edges denote the edges of E and the dashed edges represent the links L. In both pictures, the blue dashed links form a feasible solution.

1.2 Our results

In this paper, we give the first approximation algorithm with approximation ratio better than 2 for 2-SCAP, and for k-SAG for any k. To do this we introduce and solve the Steiner Ring Augmentation Problem (SRAP).

▶ **Problem 1.3** (Steiner Ring Augmentation Problem). *We are given a cycle $H = (V(H), E)$, which is a subgraph of $G = (V, E \dot{\cup} L)$. The links L have non-negative costs $c : L \to \mathbb{R}_{\geq 0}$. Furthermore, we are given a set of terminals $R \subseteq V(H)$.*

The goal is to select $S \subseteq L$ of minimum cost so that the graph $H' = (V, E \cup S)$ has 3 *pairwise edge-disjoint paths between u and v for all $u, v \in R$.*

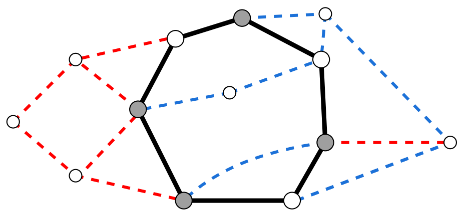

Figure 4 A SRAP instance where the black edges denote the given cycle, the dashed edges are the links, and the blue links form a feasible solution. The shaded nodes are the terminals R.

In terms of approximability, the Steiner Ring Augmentation Problem captures both WCAP and STAP as special cases, in the sense that an α-approximation for SRAP implies the same guarantee for both WCAP and STAP. We show that it also implies improved approximation algorithms for 2-SCAP and k-SAG.

▶ **Lemma 1.4.** *If there is an α-approximation for SRAP, then there is an α-approximation for* 2*-SCAP. If there is an α-approximation for SRAP when $R = V(H)$, then there is an α-approximation for k-SAG.*

Our main result is an improved approximation algorithm for SRAP.

▶ **Theorem 1.5.** *There is a $(1 + \ln 2 + \varepsilon)$-approximation algorithm for SRAP.*

Our result on SRAP implies an improved approximation for k-SCAP when $k = 2$.

▶ **Corollary 1.6.** *There is a $(1 + \ln 2 + \varepsilon)$-approximation algorithm for the 2-Steiner Connectivity Augmentation Problem.*

This result, together with the results in [20] show that a better-than-2 approximation can be achieved for k-SCAP whenever $k \in \{1, 2\}$. Recall that in the case of global connectivity augmentation, these two cases are enough to obtain improved algorithms for any k.

Unfortunately, these reductions no longer hold in the Steiner setting, where the cuts to be covered are much more complicated. The challenge of obtaining a result for k-SCAP when $k \geq 3$ is that the cuts to be covered are no longer a subcollection of the minimum cuts in the given graph. Thus, there is no way to represent these in a cactus or ring structure.

However, in the case of k-SAG, we can replace H with a cactus, and subsequently a ring without changing the structure of its minimum cuts. Hence, the k-SAG problem ultimately reduces to the special case of SRAP where all cycle nodes are terminals. In this case, we can adapt the local search methodology introduced by Traub and Zenklusen [24] to achieve an improved approximation ratio of $(1.5 + \varepsilon)$.

▶ **Theorem 1.7.** *There is a $(1.5 + \varepsilon)$-approximation algorithm for SRAP when $R = V(H)$. Hence there is a $(1.5 + \varepsilon)$-approximation algorithm for the k-SAG problem for any k.*

1.3 Related Work

There is a broad body of work on algorithms for network connectivity, for various notions of robustness and in both the weighted and unweighted settings.

We focus on edge-connectivity. Jain's iterative rounding algorithm achieves an approximation ratio of 2 for the general survivable network design problem [15] in the weighted setting. As discussed above, this implies a 2-approximation for minimum cost k-edge-connected spanning subgraph (min k-ECSS) by taking $r_{ij} = k$ for all pairs of nodes. This is the best currently known approximation ratio for this problem, and can be achieved through a variety of classical methods such as the Primal Dual method, see [25].

The Weighted Connectivity Augmentation Problem (WCAP) is a special case of min k-ECSS where there is a 0 cost $(k-1)$-edge-connected spanning subgraph. For odd k, WCAP reduces to the extensively studied Weighted Tree Augmentation Problem (WTAP). Even WTAP is known to be APX-hard and is NP-hard on various special cases such as when the given tree has diameter four [10][17]. The current best approximation ratio for WTAP is $(1.5 + \varepsilon)$ due to Traub and Zenklusen in [24]. Their approach was inspired by the work of Cohen and Nutov [6] who gave a $(1 + \ln 2)$-approximation for WTAP when the given tree has constant diameter.

It is known that the natural linear programming relaxation for WTAP has integrality gap at most 2 and at least 1.5 [5]. Stronger formulations have been considered for WTAP: Fiorini et. al. introduced the ODD-LP [9] which can be optimized over in polynomial time, despite having exponentially many constraints. The ODD-LP has integrality gap at most $2 - \frac{1}{2^{h-1}}$ for WTAP on instances where the given tree can be rooted to have height h [19]. For unweighted tree augmentation, SDP relaxations with an integrality gap of $(\frac{3}{2} + \varepsilon)$ have been studied [3, 4].

In the case that k is even, WCAP reduces in an approximation preserving way to the Weighted Cactus Augmentation Problem. Traub and Zenklusen broke the barrier of 2 for this problem and gave a $(1.5 + \varepsilon)$-approximation algorithm, matching the approximation ratio for WTAP [22].

Improved ratios can be obtained for these and other problems in the unweighted case. For unweighted 2-ECSS, a factor $\frac{4}{3}$-approximation was known [21, 14] with recent improvements achieving 1.326 [12] and finally $(1.3 + \varepsilon)$ [16]. For unweighted TAP, the state of the art is a 1.393-approximation due to [2], which also holds more generally for the unweighted

Connectivity Augmentation Problem. The first better-than-2-approximation for the Forest Augmentation Problem, which generalizes both (unweighted) 2-ECSS and TAP, was given recently by Grandoni, Jabal Ameli, and Traub [13].

Nutov [18] considered the node-weighted version of SNDP and provided an $O(k \log n)$-approximation for this problem, where k is the maximum connectivity requirement. This implies a tight $O(\log n)$-approximations for the node-weighted versions of all the problems considered in this paper.

1.4 Preliminaries

Suppose $G = (V, E)$ is a graph with vertex set V and edge set E. For a non-empty subset of vertices $C \subsetneq V$, the cut $\delta(C)$ consists of all edges in E with exactly one endpoint in C. For a subset $X \subseteq E$, we denote by $\delta_X(C) := \delta(C) \cap X$. A cut C is a k-cut if $|\delta(C)| = k$.

The edge-connectivity $\lambda(u, v)$ between a pair of vertices $u, v \in V$ is the maximum number of edge-disjoint paths between u and v in G. Equivalently, $\lambda(u, v) = \lambda(v, u)$ is the minimum cardinality of a cut $\delta(C)$ with $u \in C$ and $v \notin C$. A graph is said to be k-edge-connected if $\lambda(u, v) \geq k$ for all pairs $u, v \in V$. Given a subset of terminals $R \subseteq V$, we say that G is Steiner k-edge-connected on R if $\lambda(u, v) \geq k$ for all pairs of terminals $u, v \in R$.

Given a Steiner k-edge-connected graph $G = (V, E)$ on terminals R, fix $r \in R$, and define

$$\mathcal{C}'' := \{C \subseteq V \setminus r : |\delta(C)| = k, C \cap R \neq \emptyset\}$$

to be the family of k-cuts of G which separate some terminal from r. We call the cuts in $\mathcal{C}''$ **dangerous cuts**.

The k-SCAP problem is a hitting set problem where the ground set is the collection of links L, and the sets are $\delta(C)$ where $C \in \mathcal{C}''$. That is, a set of links which is a solution to this hitting set problem will cause the graph to become Steiner $(k + 1)$-edge-connected. The link ℓ "covers" those dangerous cuts C for which $\ell \in \delta(C)$.

Recall the Steiner Ring Augmentation Problem.

▶ **Problem 1.3** (Steiner Ring Augmentation Problem). *We are given a cycle $H = (V(H), E)$, which is a subgraph of $G = (V, E \dot\cup L)$. The links L have non-negative costs $c : L \to \mathbb{R}_{\geq 0}$. Furthermore, we are given a set of terminals $R \subseteq V(H)$.*

The goal is to select $S \subseteq L$ of minimum cost so that the graph $H' = (V, E \cup S)$ has 3 pairwise edge-disjoint paths between u and v for all $u, v \in R$.

We refer to the nodes in R as **terminals**, and the nodes in $V \setminus R$ as **Steiner nodes**. We will also refer to H as the **ring** to distinguish it from a generic cycle. It will be convenient to fix a root $r \in R$ of the ring and an edge $e_r \in E$ incident on r.

We now introduce some terminology which allows us to phrase the SRAP problem as a covering problem on a collection of ring-cuts only. Indeed, given the ring $H = (V(H), E)$, denote the set of min-cuts of H as:

$$\mathcal{C}' = \{C \subseteq V(H) \setminus r : |\delta_E(C)| = 2\}.$$

Since we are only interested in connectivity between the terminals, we only need to cover the subfamily of $\mathcal{C}'$ which separates terminals. Let

$$\mathcal{C} = \{C \subseteq V(H) \setminus r : |\delta_E(C)| = 2, C \cap R \neq \emptyset\}.$$

We call the cuts in $\mathcal{C}$ **dangerous ring-cuts**.

We will use a similar notion of "full components" as introduced in Ravi, Zhang and Zlatin [20] for STAP. Consider any solution $S \subseteq L$ to SRAP.

▶ **Definition 1.8.** *A* ***full component*** *of a SRAP solution S, is a maximal subtree of the solution where each leaf is a ring node (that is, a vertex of $V(H)$), and each internal node is in $V \setminus V(H)$.*

It is a basic fact that any link-minimal SRAP solution can be uniquely decomposed into link-disjoint full components.

▶ **Definition 1.9.** *Let S be a solution to SRAP. We say that a set $A \subseteq V(H)$ is* ***joined*** *by S if there is a full component with leaves A.*

▶ **Definition 1.10.** *We say that a cut $C \in \mathcal{C}$ is* ***covered*** *by a solution S if $A \cap C \neq \emptyset$ and $A \cap \bar{C} \neq \emptyset$ for some subset of ring nodes A which are joined by a full component of S.*

Hence, we can think of the SRAP problem as the problem of hitting the dangerous ring-cuts with full components.

▶ **Lemma 1.11.** *A solution S is feasible for SRAP iff all dangerous ring-cuts are covered by S.*

This motivates the definition of the Hyper-SRAP problem: we are given a ring $H = (V(H), E)$ with terminals $R \subseteq V(H)$, a root vertex $r \in R$, and a collection of hyper-links $\mathcal{L} \subseteq 2^{V(H)}$ with non-negative costs $c : \mathcal{L} \to \mathbb{R}_{\geq 0}$.

Let $\mathcal{C} = \{C \subseteq V(H) \setminus r : |\delta_E(C)| = 2, C \cap R \neq \emptyset\}$ be the set of dangerous ring-cuts. A cut $C \in \mathcal{C}$ is **covered** by a hyper-link ℓ if $\ell \cap C \neq \emptyset$ and $\ell \cap \bar{C} \neq \emptyset$. The Hyper-SRAP problem is to find a minimum cost subset of hyper-links so that all cuts in $\mathcal{C}$ are covered.

We will use the notion of the hyper-link intersection graph to characterize feasible solutions to Hyper-SRAP. First, we define what it means for two hyper-links to be intersecting.

▶ **Definition 1.12.** *Let ℓ and ℓ' be a pair of hyper-links. Let $(v_1, \ldots, v_k)$ be the sequence of vertices of $\ell \cup \ell'$ obtained by traversing the ring (in either direction). Then ℓ and ℓ' are* ***intersecting*** *if there are vertices $v_{i_1}, v_{i_2}, v_{i_3}, v_{i_4}$ with $i_1 < i_2 < i_3 < i_4$ such that $v_{i_1}, v_{i_3} \in \ell$ and $v_{i_2}, v_{i_4} \in \ell'$.*

Given an instance of Hyper-SRAP with ring $H = (V(H), E)$ and hyper-links $\mathcal{L}$, we define the hyper-link intersection graph Γ as follows. For each hyper-link $\ell \in \mathcal{L}$ there is a node v_ℓ. Two nodes v_{ℓ_1} and v_{ℓ_2} are adjacent in the hyper-link intersection graph if and only if ℓ_1 and ℓ_2 are intersecting hyper-links. For ring vertices $u, v \in V(H)$, we say that there is a path from u to v in Γ if there is a path in Γ from a hyper-link containing u to a hyper-link containing v.

▶ **Definition 1.13.** *We say that a full component is* ***γ-restricted*** *if it joins at most γ ring nodes. We say that a solution to SRAP is* ***γ-restricted*** *if it uses only γ-restricted full components. Analogously, we say an instance of Hyper-SRAP is* ***γ-restricted*** *if each hyper-link has size at most γ.*

Similar to the approach developed in [20] for STAP, we can work with γ-restricted solutions to SRAP while losing an arbitrarily small constant in the approximation ratio. This follows from a result of Borchers and Du for Steiner trees [1].

▶ **Lemma 1.14.** *For an instance of SRAP, let S^* be an optimal solution and S_γ be an optimal γ-restricted solution, where $\gamma(\varepsilon) = 2^{\lceil \frac{1}{\varepsilon} \rceil}$ for some $\varepsilon > 0$. Then $\frac{c(S_\gamma)}{c(S^*)} \leq 1 + \varepsilon$.*

Recall that the minimum cost Steiner tree problem can be solved in polynomial time when the number of terminals is constant.

▶ **Theorem 1.15** (Dreyfus and Wagner [8]). *The minimum Steiner tree problem can be solved in time $O(n^3 \cdot 3^p)$ where p is the number of terminals.*

Because Lemma 1.11 shows that the feasibility of a solution only depends on the nodes that are joined by full components of S, we can effectively disregard the Steiner nodes outside the ring and observe that any instance of SRAP is equivalent to an instance of Hyper-SRAP in which the cost of a hyper-link A is the minimum cost Steiner tree connecting A in the graph $(A \cup (V \setminus V(H)), L)$. By Lemma 1.14, and Theorem 1.15, we can perform this reduction from an arbitrary SRAP instance to an instance of γ-restricted Hyper-SRAP in polynomial time while only losing a factor of $(1+\varepsilon)$ in the approximation ratio.

Finally, we will make use of directed solutions to the SRAP problem. If $\vec{F}$ is a collection of directed links between pairs of vertices of the ring, then a dangerous ring-cut $C \in \mathcal{C}$ is covered by $\vec{F}$ if $\delta^-_{\vec{F}}(C) \neq \emptyset$, i.e. if there is an arc in $\vec{F}$ which enters C. Then $\vec{F}$ is a feasible **directed solution** if all dangerous ring-cuts are covered by $\vec{F}$. Analogously, if S is a set of undirected links and $\vec{F}$ is a set of directed links then we will say that $S \cup \vec{F}$ is a feasible **mixed solution** if every dangerous ring-cut is covered by S or by $\vec{F}$.

2 Technical Overview

The main contribution of this article is to prove Theorem 1.5.

▶ **Theorem 1.5.** *There is a $(1 + \ln 2 + \varepsilon)$-approximation algorithm for SRAP.*

We show this implies improved approximation guarantees for both 2-SCAP and k-SAG. Recall that in the k-SCAP problem, we are given a graph H which is Steiner k-edge-connected on the terminal set R. We want to augment this graph by including additional links of minimum cost so that there exists $k+1$ pairwise edge-disjoint paths between every pair of terminals. This is equivalent to covering the dangerous cuts of H with a minimum cost set of links.

Note that if k is larger than the global edge-connectivity of H, then there may be exponentially many dangerous cuts. Indeed even when $|R| = 2$, the number of minimum $\{s,t\}$-cuts in a graph G on n nodes may be exponential in n. This shows that, unlike in global connectivity augmentation, the set of cuts to be covered in the k-SCAP problem cannot be represented by a cactus structure that is efficiently computable. Although it is true that there are polynomially many dangerous cuts up to separating the same subset of terminals [7], this is not sufficient for our purposes.

However, in the case of $k = 2$, we show that we may take H to be a globally 2-edge-connected graph rather than merely Steiner 2-edge-connected. This allows us to compactly represent the cuts to be covered by a cactus on a set of nodes, some of which are terminals. We then use the technique introduced by Gálvez et. al. [11] to replace the cactus with a ring by adding in links of 0 cost. This brings the problem into the SRAP framework. In the case of k-SAG, we are guaranteed that H is k-edge-connected so we can directly replace H with a cactus with the same cut structure. This yields:

▶ **Lemma 1.4.** *If there is an α-approximation for SRAP, then there is an α-approximation for 2-SCAP. If there is an α-approximation for SRAP when $R = V(H)$, then there is an α-approximation for k-SAG.*

2.1 A Relative Greedy Algorithm for SRAP

In order to prove Theorem 1.5, we give a relative greedy algorithm which follows the methodology developed by Traub and Zenklusen in [22] for the Weighted Ring Augmentation Problem. However, there are several key ingredients which are needed to obtain the result in the Steiner setting. See Algorithm 1.

Our algorithm begins by preprocessing the given SRAP instance to obtain what we call a “complete” instance. This involves several rounds of metric completion in addition to the shadow-completion which is used in [22]. Complete instances are discussed in detail in Section 2.2.

Next, we compute an initial, highly structured, directed 2-approximation for SRAP, which we call an R-special solution. Roughly speaking, an R-special solution is a directed solution which is a planar r-out-arborescence, has out-degree at most 2, and is only incident on the terminals R. We formally define the properties that are required of this initial solution in Section 2.3, and show how to compute one in polynomial time on any complete SRAP instance.

We now iteratively improve upon our initial 2-approximation by adding and dropping links while maintaining feasibility. We first utilize Lemma 1.14 and Theorem 1.15 to compute the costs of all hyper-links of size at most γ. Then, in each iteration, we greedily select an “α-thin” subset of hyper-links to add to our solution, dropping links from the initial directed solution while maintaining feasibility. The collection is chosen so as to minimize the ratio of costs between the hyper-links added and the directed links which can be dropped as a result of adding these hyper-links. We select the collection of hyper-links which minimizes this ratio over all α-thin subsets of hyper-links. For technical reasons, in the case that the minimum ratio is above 1, we perform an ad hoc operation which drops at least one directed link without increasing our overall cost.[2] At the end of the algorithm, we include all links in each full component corresponding to chosen hyper-links.

In Section 2.4, we show that step 6 can be executed in polynomial time. We also show in Section 2.4 the main decomposition theorem for Hyper-SRAP, which is necessary for proving the desired approximation guarantee.

▶ **Theorem 2.1.** *Algorithm 1 is a* $(1 + \ln 2 + \varepsilon)$*-approximation algorithm for SRAP.*

2.2 Complete Instances of SRAP

We will perform several preprocessing steps on our instance to ensure that certain links are available, which will be necessary to prove the existence of an R-special solution. After these steps, we will obtain an equivalent “completed” instance which can be efficiently computed.

We perform three operations: metric completion, shadow completion, and then a second metric completion on the resulting directed links. We call the added links from these steps L^1, L^2, and L^3 respectively.

For the first metric completion, we place a new undirected link between each pair of vertices of the ring $u, v \in V(H)$, with cost equal to the shortest path from u to v which uses only links (if there is no link-path from u to v the cost is infinity). This does not affect the cost of the optimal solution, so we may assume without loss of generality that the instance is metric complete. After this step, there is an undirected link between every pair of vertices in the ring.

[2] In this step, κ_f is a particular set of undirected links such that $f \in \text{drop}(\kappa_f)$ and $c(f) \leq c(\kappa_f)$.

Algorithm 1 Relative greedy algorithm for SRAP.

Input: A complete instance of SRAP with graph $G = (V, E\dot{\cup}L)$, ring $H = (V(H), E)$, terminals $R \subseteq V(H)$ and $c : L \to \mathbb{R}$. Also an $\varepsilon > 0$.
Output: A solution $S \subseteq L$ with $c(S) \leq (1 + \ln(2) + \varepsilon) \cdot c(\mathrm{OPT})$.

1. Compute a 2-approximate R-special directed solution $\vec{F}_0$ (Theorem 2.4).
2. Let $\varepsilon' := \frac{\varepsilon/2}{1+\ln 2+\varepsilon/2}$ and $\gamma := 2^{\lceil 1/\varepsilon' \rceil}$.
3. For each $A \subseteq V(H)$ where $|A| \leq \gamma$, compute the cheapest full component joining A and denote the cost by c_A.
4. Create an instance of γ-restricted Hyper-SRAP on the ring $H = (V(H), E)$ with hyper-links $\mathcal{L} = \{\ell_A : A \subseteq V(H), |A| \leq \gamma\}$. Set the cost of hyper-link ℓ_A to be c_A.
5. Initialize $S_0 := \emptyset$, and let $\alpha := 4\lceil 4/\varepsilon \rceil$.
6. While $\vec{F}_i \neq \emptyset$:
   - Increment i by 1.
   - Compute the α-thin subset of hyper-links $Z_i \subseteq \mathcal{L}$ minimizing $\frac{c(Z_i)}{c(\mathrm{drop}_{\vec{F}_0}(Z_i) \cap \vec{F}_{i-1})}$.
   - If $\frac{c(Z_i)}{c(\mathrm{drop}_{\vec{F}_0}(Z_i) \cap \vec{F}_{i-1})} > 1$, then update $Z_i = \kappa_f$ for some $f \in \vec{F}_{i-1}$.
   - Let $S_i := S_{i-1} \cup Z_i$ and let $\vec{F}_i := \vec{F}_{i-1} \setminus \mathrm{drop}_{\vec{F}_0}(Z_i)$.
7. **Return** A SRAP solution with full components corresponding to the hyper-links in $S := S_i$.

For shadow completion, we fix a root $r \in V(H)$ and an edge $e_r \in E$ incident to r. For each link $\ell = (u, v)$ with $u, v \in V(H)$, we will add to the instance a collection of directed links known as the **shadows** of ℓ. These will all have the same cost as ℓ. The shadows of ℓ consist of the two directed links (u, v) and (v, u) as well as all shortenings of these two directed links. If (u, v) is a directed link, then (s, t) is called a **shortening** of (u, v) if $v = t$ and s is a vertex on the path from u to v in $E \setminus e_r$. With this definition, it is not hard to see that a shadow of ℓ covers a subset of the cuts in $\mathcal{C}$ that ℓ covers. Hence, we may perform this step without affecting the cost of the optimal solution.

Finally, we do a second round of metric completion on the newly added directed links, similar to the first. For every pair of vertices $u, v \in V(H)$, if there is a directed path from u to v, we add a directed link (u, v) with cost equal to the cost of the shortest such path.

After these operations, we are left with an instance of SRAP such that any solution to this instance can be converted into a solution to the original SRAP instance with the same cost. Hence, we may assume that these operations have been performed on the given SRAP instance without loss of generality. See Figure 5 for an example of this preprocessing.

Furthermore, we claim that any further iterations of these two preprocessing operations (metric completion and shadow completion) will not change the instance.

▶ **Definition 2.2.** *An instance of SRAP is called a* ***complete instance*** *if for every $u, v \in V(H)$, there is a directed link (u, v) whose cost is equal to the least-cost directed path from u to v, and for every directed link (u, v), the instance contains all shortenings of (u, v) with at most cost $c((u, v))$.*

▶ **Lemma 2.3.** *Any SRAP instance can be made complete by performing metric completion, then shadow completion, then a second metric completion.*

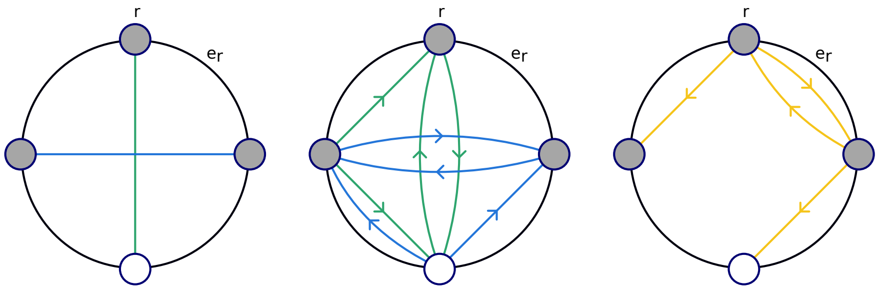

Figure 5 An example of a SRAP instance undergoing preprocessing steps to obtain a complete instance. The leftmost SRAP instance has two undirected links ℓ_1 (green) and ℓ_2 (blue) in L. There are no links added in L^1. The middle picture shows the directed links added in L^2, where the green arcs are shadows of ℓ_1 and have cost $c(\ell_1)$, and the blue arcs are shadows of the ℓ_2 with cost $c(\ell_2)$. Finally, the third picture shows the (undominated) directed links added in L^3 in yellow. Each of these arcs have cost $c(\ell_1) + c(\ell_2)$. The final completed SRAP instance contains all of these links.

2.3 An R-special 2-approximate solution

We show that for any complete instance of SRAP, we can compute a directed 2-approximate arborescence solution which is only incident on terminals. We call a directed solution which satisfies the properties in Theorem 2.4 an ***R*-special** solution. Let OPT denote the cost of the optimal solution.

▶ **Theorem 2.4.** *Given a complete instance of SRAP, there is a polynomial time algorithm which yields a directed solution $\vec{F}$ of cost at most $2OPT$ such that:*
1. *$\vec{F}$ is only incident on the terminals R*
2. *$(R, \vec{F})$ is an r-out arborescence.*
3. *$(R, \vec{F})$ is planar when $V(H)$ is embedded as a circle in the plane.*
4. *For any $v \in V$, no two directed links in $\delta^+_{\vec{F}}(v)$ go in the same direction along the ring.*

The proof of Theorem 2.4 is significantly more involved than the directed 2-approximation for standard WRAP. Unlike in WRAP, the optimal solution for SRAP may use links between Steiner nodes outside the ring in order to increase the connectivity between pairs of terminals in the ring. Nonetheless, using analagous techniques to [20] for STAP, we can obtain a 2-approximate solution consisting of directed links only between ring nodes. Essentially, taking a directed cycle on the nodes joined by each full component of the optimal solution yields a directed solution of at most 2 times the cost. We can do the same for SRAP, yielding:

▶ **Lemma 2.5.** *Given a complete instance of SRAP, there is a directed solution of cost at most $2OPT$ consisting of a collection of directed cycles on the ring nodes.*

At this point we could iteratively shorten the solution to obtain a 2-approximate planar r-out arborescence as in [22]. However, this is not sufficient for our purposes as this directed solution still uses Steiner nodes *inside* the ring.

Hence, to prove Theorem 2.4, we will begin with the 2-approximate directed solution consisting of a collection of directed cycles on the ring nodes as guaranteed by Lemma 2.5. Then, we prove a "cycle merging lemma" which allows us to iteratively merge these cycles to eventually obtain a feasible directed solution which is a *single* directed cycle of at most the cost. See Figure 6 for an example of cycle merging.

▶ **Lemma 2.6** (Cycle Merging Lemma). *Consider a complete SRAP instance on the ring $H = (V(H), E)$. Let $\vec{F}_S$ and $\vec{F}_A$ be two directed cycles of links on nodes $S, A \subseteq V(H)$, respectively, where $r \in S$. If S and A are intersecting as hyper-links, then there exists a directed link set $\vec{F}_{S\cup A}$ which forms a directed cycle on $S \cup A$ of cost at most $c(\vec{F}_S) + c(\vec{F}_A)$.*

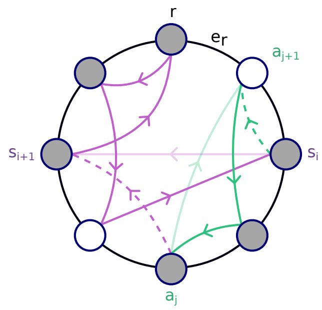

Figure 6 An example of cycle merging, with $\vec{F}_S$ in solid purple and $\vec{F}_A$ in solid green. Note that dropping the faded links (s_i, s_{i+1}) and (a_j, a_{j+1}) and adding the shortenings drawn as dotted arcs creates the single cycle $\vec{F}_{S \cup A}$.

Iteratively applying Lemma 2.6 to the directed solution from Lemma 2.5 yields:

▶ **Lemma 2.7.** *Given a complete instance of SRAP, let the optimal solution have cost OPT. Then there exists a directed solution of cost at most $2OPT$ whose links form a single directed cycle containing r.*

Finally, we can use the links added in the second metric completion step to shortcut over the Steiner nodes in the ring to obtain a directed cycle solution that only touches terminals.

▶ **Lemma 2.8.** *Given an instance of SRAP, if the optimal solution has cost OPT, then there is a directed solution of cost at most $2OPT$ whose links consist of a single directed cycle with node set R.*

Having shown that there exists a directed 2-approximate solution to any complete SRAP instance which is incident only on the terminals R, we now proceed to show how to compute one in polynomial time, and that we can find one with additional structure.

Recall that the Weighted Ring Augmentation Problem (WRAP) is a special case of SRAP in which all nodes are terminals, and there are no nodes outside of the ring H. A directed solution $\vec{F}$ to WRAP is **non-shortenable** if it is feasible but deleting or strictly shortening any link $f \in \vec{F}$ results in an infeasible solution. Traub and Zenklusen proved [22, Theorem 2.6] that a non-shortenable directed solution to WRAP is a planar r-out arborescence in which all nodes have out-degree at most 2. In other words, it is $V(H)$-special.

We can now obtain our R-special solution by computing the cheapest directed solution on the induced WRAP instance with ring nodes R, and then iteratively shortening the directed links as long as feasibility is maintained. This yields an R-special solution and due to Lemma 2.8, it has cost at most 2OPT. Hence, Theorem 2.4 follows.

2.4 Decomposition and Optimization Theorems

An R-special directed solution is necessary because it allows us to leverage a decomposition result on directed solutions with respect to the optimum. Because of the decomposition result, it can be argued that every iteration of the algorithm will find an improving local move as long as the current solution is expensive. Traub and Zenklusen prove a decomposition result of this kind to bound the approximation ratio of the relative greedy algorithm for WRAP [22]. In our setting, we need an analagous decomposition theorem for the hyper-links which arise from considering the induced γ-restricted Hyper-SRAP instance computed in step 4 of Algorithm 1. However, we do not need to use the γ-restricted structure; we prove the decomposition theorem for arbitrary Hyper-SRAP instances with respect to any R-special directed solution.

First we need to define the notion of an α-thin collection of hyper-links. Recall that $\mathcal{C}'$ is the set of minimum cuts of the ring H which do not include the root.

▶ **Definition 2.9.** *A collection of hyper-links $K \subseteq \mathcal{L}$ is α-thin if there exists a maximal laminar subfamily $\mathcal{D}$ of $\mathcal{C}'$ such that for each $C \in \mathcal{D}$, the number of hyper-links in K which cover C is at most α.*

With this definition, we can prove Theorem 2.10 for Hyper-SRAP.

▶ **Theorem 2.10** (Decomposition Theorem)**.** *Given an instance of Hyper-SRAP $(H = (V(H), E), R, \mathcal{L})$, suppose $\vec{F}_0$ is an R-special directed solution and $S \subseteq \mathcal{L}$ is any solution. Then for any $\varepsilon > 0$, there exists a partition $\mathcal{Z}$ of S into parts so that:*
- *For each $Z \in \mathcal{Z}$, Z is α-thin for $\alpha = 4\lceil 1/\varepsilon \rceil$.*
- *There exists $Q \subseteq \vec{F}_0$ with $c(Q) \leq \varepsilon \cdot c(\vec{F}_0)$, such that for all $f \in \vec{F}_0 \setminus Q$, there is some $Z \in \mathcal{Z}$ with $f \in drop_{\vec{F}_0}(Z)$. That is, $\vec{F}_0 \setminus Q \subseteq \bigcup_{Z \in \mathcal{Z}} drop_{\vec{F}_0}(Z)$.*

In the above theorem, for a collection of hyper-links K and an R-special directed solution $\vec{F}_0$, the notation $\text{drop}_{\vec{F}_0}(K)$ denotes a set of directed links from $\vec{F}_0$ which can be dropped while preserving feasibility of $\vec{F}_0 \cup K$. For each directed link $f \in \vec{F}_0$, we describe a collection of dangerous ring-cuts for which f is "responsible", denoted $\mathcal{R}(f)$, and defined formally in the full paper.

Then

$$\text{drop}_{\vec{F}_0}(K) := \{f \in \vec{F}_0 : |\delta_K(C)| \geq 1 \text{ for all } C \in \mathcal{R}_{\vec{F}_0}(f)\}$$

is the set of all directed arcs such that the cuts they are responsible for are covered by K.

This definition was used by Traub and Zenklusen [22] for WRAP. They show that a link (u, v) can be dropped if and only if v is connected to a "v-good" vertex in the link-intersection graph restricted to K. In [22], a v-good vertex is a vertex which is not a descendant of v with respect to the initial 2-approximate arborescence.

This is no longer the case in our setting, since it is possible to drop (u, v) even if K connects v to some Steiner node in the ring. Hence, we redefine the notions of v-good and v-bad to characterize when a directed link can be dropped.

▶ **Definition 2.11.** *Let $v \in R$ and consider the maximal interval $I_v \subseteq V(H)$ containing v such that I_v does not contain a terminal which is a non-descendant of v in $(R, \vec{F}_0)$. We say that the nodes in I_v are **v-bad**, and all nodes in $V(H) \setminus I_v$ are **v-good**.*

We prove the following.

▶ **Lemma 2.12.** *For a collection of hyper-links K, a directed link (u, v) is in $drop_{\vec{F}_0}(K)$ if and only if $\Gamma(K)$ contains a path from a hyper-link containing v to a hyper-link containing a v-good vertex w.*

Note that Lemma 2.12 does not necessarily hold if $\vec{F}_0$ is not R-special, since a directed link entering a Steiner node could be droppable even if K has no hyper-links containing it. We also show that an R-special solution $\vec{F}_0$ can be augmented with artificial links to obtain a $V(H)$-special solution $\vec{F}'$ so that the set of v-bad nodes in $\vec{F}_0$ correspond to the set of descendants of v in the arborescence $\vec{F}'$.

We will use our characterization of when a directed arc is droppable to prove Theorem 2.10. We follow the approach in [22] which proves the result when all hyper-links have size 2 and $R = V(H)$. We construct a "dependency graph" which allows us to partition the links of S into the desired α-thin pieces. In our hyper-link setting, the nodes of the dependency graph correspond to "festoons" composed of hyper-links rather than festoons of links of size 2.

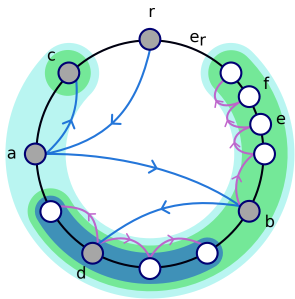

Figure 7 An example of an R-special solution with $R = \{r, a, b, c, d\}$ and its extension to an artificial $V(H)$-special solution. The artificial links are purple. The r-bad interval is always $V(H)$. The a-bad interval is shown in cyan. The b-bad and c-bad intervals are green, and the d-bad interval is dark blue.

Lemma 1.14 and Theorem 1.15 allow us to convert a given instance of SRAP into an equivalent Hyper-SRAP instance efficiently. It also ensures that each local move of the algorithm runs in polynomial time. In each local move, the algorithm chooses amongst all α-thin collections of hyper-links K, the choice which minimizes the ratio between the cost of the hyper-links in K, and the cost of the directed links which will be dropped as a result of adding K to the solution. We show that this operation can be performed in polynomial time as long as all hyper-links have size at most γ.

▶ **Theorem 2.13.** *Given an instance of γ-restricted Hyper-SRAP, an R-special directed solution $\vec{F}_0$, a set of directed links $\vec{F} \subseteq \vec{F}_0$, and an integer $\alpha \geq 1$, there is a polynomial time algorithm which finds a collection of hyper-links K minimizing*

$$\frac{c(K)}{c(drop_{\vec{F}_0}(K) \cap \vec{F})}$$

over all α-thin subsets of hyper-links.

Theorems 2.4, 2.10, and 2.13 together are essentially enough to employ a relative greedy strategy to obtain a $(1 + \ln 2 + \varepsilon)$-approximation for SRAP. Beginning with an R-special 2-approximate directed solution, we iteratively add to this solution a greedily chosen α-thin collection of hyper-links in each step. When a particular collection of hyper-links K is chosen to be added, the directed links in $\text{drop}_{\vec{F}_0}(K)$ are dropped, and this process is repeated until $\vec{F}_0$ becomes empty.

2.5 A Local Search Improvement for k-SAG

Additionally, in the case that all ring nodes are terminals, we can use a local search framework introduced in [24] to give an algorithm with an improved approximation guarantee of $(1.5+\varepsilon)$. Together with Lemma 1.4, this yields:

▶ **Theorem 1.7.** *There is a $(1.5 + \varepsilon)$-approximation algorithm for SRAP when $R = V(H)$. Hence there is a $(1.5 + \varepsilon)$-approximation algorithm for the k-SAG problem for any k.*

The key idea in the improvement is to not only consider dropping links from the initial directed solution, but to drop undirected links which were added in previous iterations by associating to each undirected link a witness set of directed links which indicate when it can be dropped.

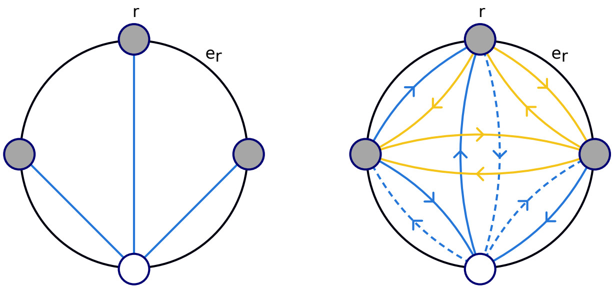

Figure 8 An example of a SRAP instance on the left, where all three undirected links have cost 1. The grey nodes are terminals. The completed instance is shown on the right, where the blue links have cost 1 and the orange links have cost 2 (only the directed links are shown). The dashed blue arcs form a feasible directed solution of cost 3, but there is no R-special solution of cost at most 3.

We are able to do this in the case that $V(H) = R$ because in this case, *any* feasible directed solution can be transformed into an R-special solution of at most the cost. However, this is not the case for general SRAP (see Figure 8).

This illustrates why it may be surprising that Theorem 2.4, our key technical contribution, holds. While it is not true that any feasible directed solution can be made R-special, the proof of Theorem 2.4 shows that a directed solution which consists of a collection of directed cycles on the ring *can* be made R-special, and this is enough to prove the desired guarantees with respect to the optimum.

References

1. Al Borchers and Ding-Zhu Du. Thek-steiner ratio in graphs. *SIAM Journal on Computing*, 26(3):857–869, 1997. doi:10.1137/S0097539795281086.
2. Federica Cecchetto, Vera Traub, and Rico Zenklusen. Bridging the gap between tree and connectivity augmentation: unified and stronger approaches. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 370–383, 2021. doi:10.1145/3406325.3451086.
3. Joseph Cheriyan and Zhihan Gao. Approximating (unweighted) tree augmentation via lift-and-project, part I: stemless TAP. *Algorithmica*, 80(2):530–559, 2018. doi:10.1007/s00453-016-0270-4.
4. Joseph Cheriyan and Zhihan Gao. Approximating (unweighted) tree augmentation via lift-and-project, part II. *Algorithmica*, 80(2):608–651, 2018. doi:10.1007/s00453-017-0275-7.
5. Joseph Cheriyan, Howard Karloff, Rohit Khandekar, and Jochen Könemann. On the integrality ratio for tree augmentation. *Operations Research Letters*, 36(4):399–401, 2008. doi:10.1016/j.orl.2008.01.009.
6. Nachshon Cohen and Zeev Nutov. A $(1 + \ln 2)$-approximation algorithm for minimum-cost 2-edge-connectivity augmentation of trees with constant radius. *Theoret. Comput. Sci.*, 489/490:67–74, 2013. doi:10.1016/j.tcs.2013.04.004.
7. Yefim Dinitz and Alek Vainshtein. The connectivity carcass of a vertex subset in a graph and its incremental maintenance. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 716–725, 1994. doi:10.1145/195058.195442.
8. Stuart E Dreyfus and Robert A Wagner. The steiner problem in graphs. *Networks*, 1(3):195–207, 1971. doi:10.1002/net.3230010302.
9. Tamás Fleiner and András Frank. A quick proof for the cactus representation of mincuts. *EGRES Quick Proof*, 3:2009, 2009.
10. Greg N Frederickson and Joseph Ja'Ja'. Approximation algorithms for several graph augmentation problems. *SIAM Journal on Computing*, 10(2):270–283, 1981. doi:10.1137/0210019.

11 Waldo Gálvez, Fabrizio Grandoni, Afrouz Jabal Ameli, and Krzysztof Sornat. On the cycle augmentation problem: hardness and approximation algorithms. *Theory of Computing Systems*, 65(6):985–1008, 2021. doi:10.1007/s00224-020-10025-6.

12 Mohit Garg, Fabrizio Grandoni, and Afrouz Jabal Ameli. Improved approximation for two-edge-connectivity. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 2368–2410. SIAM, 2023. doi:10.1137/1.9781611977554.ch92.

13 Fabrizio Grandoni, Afrouz Jabal Ameli, and Vera Traub. Breaching the 2-approximation barrier for the forest augmentation problem. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 1598–1611. ACM, 2022. doi:10.1145/3519935.3520035.

14 Christoph Hunkenschröder, Santosh S. Vempala, and Adrian Vetta. A 4/3-approximation algorithm for the minimum 2-edge connected subgraph problem. *ACM Trans. Algorithms*, 15(4):55:1–55:28, 2019. doi:10.1145/3341599.

15 Kamal Jain. A factor 2 approximation algorithm for the generalized steiner network problem. *Combinatorica*, 21(1):39–60, 2001. doi:10.1007/s004930170004.

16 Yusuke Kobayashi and Takashi Noguchi. An approximation algorithm for two-edge-connected subgraph problem via triangle-free two-edge-cover. In Satoru Iwata and Naonori Kakimura, editors, *34th International Symposium on Algorithms and Computation, ISAAC 2023, December 3-6, 2023, Kyoto, Japan*, volume 283 of *LIPIcs*, pages 49:1–49:10. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ISAAC.2023.49.

17 Guy Kortsarz, Robert Krauthgamer, and James R Lee. Hardness of approximation for vertex-connectivity network design problems. *SIAM Journal on Computing*, 33(3):704–720, 2004. doi:10.1137/S0097539702416736.

18 Zeev Nutov. Approximating steiner networks with node-weights. *SIAM Journal on Computing*, 39(7):3001–3022, 2010. doi:10.1137/080729645.

19 Ojas Parekh, R. Ravi, and Michael Zlatin. On small-depth tree augmentations. *CoRR*, abs/2111.00148, 2021. doi:10.48550/arXiv.2111.00148.

20 R Ravi, Weizhong Zhang, and Michael Zlatin. Approximation algorithms for steiner tree augmentation problems. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2429–2448. SIAM, 2023. doi:10.1137/1.9781611977554.ch94.

21 András Sebö and Jens Vygen. Shorter tours by nicer ears: 7/5-approximation for the graph-tsp, 3/2 for the path version, and 4/3 for two-edge-connected subgraphs. *Comb.*, 34(5):597–629, 2014. doi:10.1007/s00493-014-2960-3.

22 Vera Traub and Rico Zenklusen. A (1.5+\epsilon)-approximation algorithm for weighted connectivity augmentation. *arXiv preprint arXiv:2209.07860*, 2022.

23 Vera Traub and Rico Zenklusen. A better-than-2 approximation for weighted tree augmentation. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1–12, 2022. doi:10.1109/FOCS52979.2021.00010.

24 Vera Traub and Rico Zenklusen. Local search for weighted tree augmentation and steiner tree. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3253–3272. SIAM, 2022. doi:10.1137/1.9781611977073.128.

25 David P Williamson, Michel X Goemans, Milena Mihail, and Vijay V Vazirani. A primal-dual approximation algorithm for generalized steiner network problems. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 708–717, 1993. doi:10.1145/167088.167268.