On Survivable Set Connectivity*

Parinya Chalermsook[†] Fabrizio Grandoni[‡] Bundit Laekhanukit[§]

Abstract

In the *Set Connectivity* problem, we are given an n-node edge-weighted undirected graph and a collection of h set pairs $(\mathcal{S}_i, \mathcal{T}_i)$, where $\mathcal{S}_i$ and $\mathcal{T}_i$ are subsets of nodes. The goal is to compute a min-cost subgraph H so that, for each set pair $(\mathcal{S}_i, \mathcal{T}_i)$, there exists at least one path in H between some node in $\mathcal{S}_i$ and some node in $\mathcal{T}_i$.

In this paper, we initiate the study of the *Survivable Set Connectivity* problem (SSC), i.e., the generalization of Set Connectivity where we are additionally given an integer requirement $k_i \geq 1$ for each set pair $(\mathcal{S}_i, \mathcal{T}_i)$, and we want to find a min-cost subgraph H so that there are at least k_i edge-disjoint paths in H between $\mathcal{S}_i$ and $\mathcal{T}_i$. We achieve the following main results:

- We show that there is no poly-logarithmic approximation for SSC unless NP has a quasi-polynomial time algorithm. This result is based on a reduction from the Minimum Label Cover problem, and the result holds even for the special case where $\mathcal{S}_i = \{r\}$ for all i, i.e., for the high-connectivity variant of the classical Group Steiner Tree problem. More precisely, we prove an approximability lower bound of $2^{\log^{1-\epsilon} n}$ for SSC, for any constant $\epsilon > 0$, which is almost polynomial on n. A technical novelty of our proof is the first use of a *padding scheme technique* for an edge-connectivity problem on undirected graphs. (Prior to our results, the applications of this technique only pertain to either node-connectivity problems or problems on directed graphs).
- We present a bicriteria approximation algorithm for SSC that computes a solution H of cost at most poly-logarithmically larger than the optimal cost and provides a connectivity at least $\Omega(k_i/\log n)$ for each set pair $(\mathcal{S}_i, \mathcal{T}_i)$. The main algorithmic idea is to solve a standard LP relaxation to the problem, and then embed the resulting fractional capacities into a tree via Räcke's cut-based tree embeddings. Based on that, we generate a random collection of Group Steiner Tree-like fractional solutions, which can then be handled by the rounding scheme of [Garg, Konjevod and Ravi – SODA'98]. The prior work on Set Connectivity and Group Steiner Tree used Bartal's distance-based tree embeddings which do not seem to generalize to the k-connectivity versions of these problems.

Finally, we remark an interesting contrast demonstrated by our results: While our hardness result almost rules out "polynomial" approximation ratios, relaxing connectivity constraints allows us to obtain "poly-logarithmic" bounds. This naturally suggests that relaxing connectivity requirements might be a proper way in getting big improvements, even beyond (non-bicriteria) lower bounds, for other connectivity problems, especially those whose approximability lower bounds are derived from Minimum Label Cover.

1 Introduction

In the *Set Connectivity* problem (SC) (a.k.a. *Generalized Connectivity* problem [1]), we are given an n-node undirected graph $G = (V, E)$ with edge costs $c : E \to \mathbb{R}^+$ and a collection of h set pairs $(\mathcal{S}_1, \mathcal{T}_1), \ldots, (\mathcal{S}_h, \mathcal{T}_h)$, where $\mathcal{S}_i, \mathcal{T}_i \subseteq V$ and $\mathcal{S}_i \cap \mathcal{T}_i = \emptyset$ for all i. Our goal is to compute a subgraph H of minimum cost $c(H) = \sum_{e \in E(H)} c(e)$ such that, for each set pair $(\mathcal{S}_i, \mathcal{T}_i)$, there exists at least one path in H between some node in $\mathcal{S}_i$ and some node in $\mathcal{T}_i$. The current best approximation guarantees for SC are $O(\max_i\{|\mathcal{S}_i| + |\mathcal{T}_i|\})$ by Fukunaga and Nagamochi [17], and $O(\log^2 n \log^2 h)$ by Chekuri et al. [8].

In this paper, we address the generalization of SC to a high connectivity setting that we call the *Survivable Set Connectivity* problem (SSC). Here we are given the same input as in SC together with an integer requirement $k_i \geq 1$ for each set pair $(\mathcal{S}_i, \mathcal{T}_i)$. Our goal now is to compute a subgraph H of minimum cost

*This work was partially done while the first and third authors were working at IDSIA. Partially supported by the ERC Starting Grant NEWNET 279352, Swiss National Science Foundation project 200020_144491/1, Natural Sciences and Engineering Research Council of Canada Grant 288334 and Dr & Mrs Milton Leong fellowship.

[†]Max-Planck-Institut für Informatik, Saarbrücken, Germany, Email:parinya@mpi-inf.mpg.de.

[‡]IDSIA, University of Lugano, Switzerland, Email:fabrizio@idsia.ch.

[§]IDSIA & McGill University, Montreal, Canada, Email:blaekh@cs.mcgill.ca.

$c(H)$ such that, for each set pair $(\mathcal{S}_i, \mathcal{T}_i)$, there are at least k_i edge-disjoint paths with one endpoint in $\mathcal{S}_i$ and the other endpoint in $\mathcal{T}_i$. Note that one can assume w.l.o.g. that the requirements k_i are *uniform*: Indeed, let $k := \max_i\{k_i\}$. It is sufficient to add $k - k_i$ dummy nodes to each set $\mathcal{T}_i$ and connect them to an arbitrary node in $\mathcal{S}_i$ with edges of cost 0. We therefore focus on the uniform version of the problem.

To the best of our knowledge, SSC was not addressed before. However, many results are known for interesting special cases of the problem (besides the already mentioned SC results). If all sets $\mathcal{S}_i$ and $\mathcal{T}_i$ have cardinality 1, SSC is equivalent to the *Survivable Network Design* problem (a.k.a. *Steiner Network* problem). Following a sequence of results [19, 34], Jain [22] gave the current-best 2-approximation for this problem. The best known approximation guarantee is 2 even when $k = 1$ (the *Steiner Forest* problem). Better approximations are known if the problem is further restricted, e.g., for the *Steiner Tree* problem [5].

Another interesting special case is obtained by requiring $\mathcal{S}_i = \{r\}$ for all i. This can be seen as a survivable generalization of the classical *Group Steiner Tree* problem (GST) that we call the *Survivable Group Steiner Tree* problem (SGST). (Sets $\mathcal{G}_i := \mathcal{T}_i$ define the groups that we need to connect to the root r). Similarly to SSC, we can assume w.l.o.g. uniform requirements. GST is a well-studied problem (see [7, 9, 18, 21, 33] and references therein). In particular, an $O(\log^2 n \log h)$-approximation was given by Garg, Konjevod, and Ravi [18], and an approximability lower bound of $\Omega(\log^{2-\varepsilon} n)$ was given by Halperin and Krauthgamer [21]. The special case of SGST with $k = 2$ has been studied independently by Gupta, Krishnaswamy and Ravi [20] and by Khandekar, Kortsarz, and Nutov [23]. The algorithm proposed by Gupta et al. [20] has an approximation guarantee of $\tilde{O}(\log^3 n \cdot \log h)$ while the algorithm of Khandekar et al. [23] has a guarantee of $O(g_{max} \cdot \log^2 n)$, where $g_{max} = \max_i\{|\mathcal{G}_i|\}$ is the maximum cardinality of any group. To the best of our knowledge, no non-trivial approximation algorithm is known for SGST for $k > 2$.

1.1 Our Contributions and Techniques

In this paper, we provide both negative and positive results on the approximability of SSC.

1.1.1 Inapproximability Results.

We prove some hardness of approximation results for SGST (which of course extend to SSC). As a poly-logarithmic approximation algorithm is known for SGST for the special cases $k = 1$ and $k = 2$, a natural question is whether a poly-logarithmic approximation algorithm for SGST exists for any $k > 2$.

Our first main result (see Section 3) is a negative answer to the above question. In particular, we present an approximation-preserving reduction from the *Minimum Label Cover* problem (MLC) to SGST. In MLC, we are given a directed bipartite graph $Q = (A \cup B, F)$, $F \subseteq A \times B$, a set of labels L, and a set of projective constraints $\{\pi_{ab}\}_{ab \in F}$, $\pi_{ab} : L \to L$. A *labeling* is a function $\mathcal{L} : A \cup B \to 2^L$. Intuitively, each node v is assigned a set of labels $\mathcal{L}(v) \subseteq L$. We say that $\mathcal{L}$ *covers* edge $ab \in F$ if there exist labels $\ell \in \mathcal{L}(a)$ and $\ell' \in \mathcal{L}(b)$ such that $\ell' = \pi_{ab}(\ell)$. The goal is to find a labeling $\mathcal{L}$ that covers all the edges of minimum cost $c(\mathcal{L}) = \sum_{v \in A \cup B} |\mathcal{L}(v)|$.

THEOREM 1.1. *Given an instance (Q, L, π) of MLC, with size m and maximum degree Δ, there is a polynomial-time algorithm that computes an instance of SGST of size $O(m)$ and with $k = O(|L|\Delta)$ so that the optimal solutions of the two instances have the same cost.*

Roughly speaking, the standard label cover construction [3, 32] implies that, for any integer $\ell > 0$, one can transform a 3-SAT instance on n variables into an MLC instance of size $n^{O(\ell)}$, with $|L|, \Delta \leq 2^{O(\ell)}$, such that approximating MLC better than $2^{\alpha \ell}$ would allow us to solve 3-SAT. Here $\alpha > 0$ is a universal constant. Combining this with Theorem 1.1, we get a k^δ approximability lower bound for SGST for some constant $\delta > 0$ and also a lower bound of $2^{\log^{1-\epsilon} n}$ for any $\epsilon > 0$, which is "almost polynomial" in n. A tighter hardness of $k^{1/6-\epsilon}$, assuming NP $\neq$ ZPP, can be obtained by taking the MLC instance from Theorem 4.2 in [26].

COROLLARY 1.1. *Unless NP has a quasi-polynomial time algorithm, there is no polynomial-time $\min\{k^\delta, 2^{\log^{1-\epsilon} n}\}$ approximation algorithm for SGST for any $k \geq k_0$, where $\delta, k_0 > 0$ are two universal constants and $\epsilon > 0$ is any constant. Moreover, assuming NP $\neq$ ZPP, there is no polynomial-time $k^{1/6-\epsilon}$-approximation algorithm for SGST, for any constant $k \geq k_0$.*

The above hardness result also applies to the connectivity-augmentation version of the problem, where we want to add a minimum-cost subset of edges to a given graph so that the connectivity from each group to the root increases from k to $k + 1$.

Our reduction follows the padding scheme technique introduced by Kortsarz et al. [24] and used in proving some recent hardness results for network design problems in high connectivity settings [6, 10, 13, 26]. We first sketch the key ideas and then highlight how our approaches depart from previous work. The basic idea of this technique is to construct a base graph where a pair of labels that covers a given edge e in the MLC instance corresponds to a proper "canonical path" from the root to some group. If the solution were restricted to use only canonical paths, this reduction would immediately give the desired hardness results. However, the problem here is that group connectivity might be achieved via non-canonical paths. To circumvent this problem, we augment the graph with a proper padding gadget which guarantees that canonical paths are indeed needed. The padding gadget in [24] exploits some properties of node-connectivity which do not exist in edge-connectivity. We carefully exploit the structure of the graph and of the groups to mimic the "behavior" of node-connectivity. In more detail, we construct an instance of SGST so that a min-cut separating each group and the root has size exactly k. For a node-connectivity problem, one can use this cut to force $k-1$ paths (of each source-sink pair) to use almost every node except a particular subset of nodes S. So, the k^{th} path must lie inside S and is eventually forced to be canonical. For an edge-connectivity problem like SGST, paths can share nodes, so the k^{th} path can still leave S. We resolve this issue by exploiting degree-3 nodes in our gadgets: these nodes cannot appear as internal nodes of two edge-disjoint paths, thus preventing the k^{th} path from leaving S.

In view of Theorem 1.1, it makes sense to consider a bicriteria version of SSC where one relaxes the connectivity constraints in order to aim at better approximation guarantees. Let OPT denote the cost of the optimum solution. We say that an algorithm is a bicriteria (α, β)-approximation algorithm if it always returns a subgraph H of cost at most $\alpha \cdot \mathsf{OPT}$, and there are at least k/β edge-disjoint paths between each set pair $(\mathcal{S}_i, \mathcal{T}_i)$. It turns out that this relaxation still captures non-trivial challenges as shown by the following result (see Section 4).

Theorem 1.2. *For any constant $\varepsilon > 0$, and for any $\alpha, \beta > 0$ such that $\alpha \cdot \beta = O(\log^{2-\epsilon} n)$, there is no polynomial time (α, β)-approximation algorithm for* SGST *(hence for* SSC*) unless* NP *has a quasi-polynomial time algorithm. This holds even when the input graph is a tree.*

In particular, this theorem implies that we cannot expect an $(\alpha, O(\log n))$-approximation algorithm for SGST (hence for SSC) if $\alpha \leq \log^{1-\epsilon} n$.

1.1.2 Approximation Algorithms.

On the positive side, we show that, when a connectivity relaxation of $O(\log n)$ is allowed, SSC admits a poly-logarithmic approximation (see Section 2). We next sketch how our bicriteria approximation algorithm works.

For the ease of presentation, let us focus on SGST. A natural approach is to exploit the natural cut-based LP relaxation for this problem. However, it is not clear how to round the corresponding fractional solution even in the simpler case of GST. In [18], the authors solved this issue (for GST) by embedding the input graph into a tree T using distance-based random tree embeddings, a.k.a. Bartal's trees [4, 16] [1]. This introduces a factor $O(\log n)$ in the approximation ratio. Then they solve the standard LP relaxation for GST on T and round the fractional solution using a dependent rounding procedure, referred to as RoundGKR. This way, each group is guaranteed to be connected to the root with probability at least $\Omega(1/\log n)$, and the expected cost of the partial solution is upper bounded by the LP value. By repeating the RoundGKR procedure for $O(\log n \log h)$ times, with high probability all groups are connected to the root.

One might try to apply the same approach to SGST, where in the second stage one uses the natural generalization of the standard LP relaxation. However, the initial distance-based embedding might hide the cut structure of the graph: While bounding the cost of each partial solution is easy, it seems hard to guarantee any connectivity larger than one in the original graph. One possibility to circumvent this problem is to use a tree-embedding that preserves both "cost" and "connectivity" upto some factor. Unfortunately, no such tree-embedding with reasonable guarantees exists for general graphs [2].

To cope with this issue, we let the LP deal with the cost, and rather use cut-based tree-embeddings, a.k.a. Räcke's trees [31], to deal with the connectivity. Specifically, we first solve the standard LP on general graphs. Then we think of variables x_e on edges as capacities and embed them into a Räcke's tree T. On this tree T, we apply RoundGKR, and the resulting partial solution is mapped back into the corresponding subgraph of the original graph. We are able to show that each such subgraph avoids routing paths using any given subset F of $O(k/\log n)$ edges with sufficiently large probability. Intuitively, this is due to the fact that the *image* F_T of F in T has a small capacity with large enough probability. Hence there are good chances that, by the properties of RoundGKR, the considered subgraph connects a given group to the root *even after removing* edges in F.

Therefore, by repeating the process a sufficiently large number of times, each separating cut contains a sufficient number of edges (namely, $\Omega(k/\log n)$ many

[1] The tight embedding in [16] is sometimes called FRT tree embedding.

edges) with large probability. This leads to the following result.

THEOREM 1.3. *For any $k \geq 1$, there is a Las Vegas polynomial-time $(O(\log^2 n(\frac{\log h}{k}+1), O(\log n))$-approximation algorithm for* SGST.

In the case of SSC, we need one more idea. Also, in this case, we initially solve the standard cut-based relaxation of the problem and use it (after scaling by a factor $\Theta(k)$) to construct a random Räcke's tree embedding T (that we can assume to have $O(\log n)$ height). We recall that the leaves of T are in one-to-one correspondence with the nodes of G. In particular, let $\mathcal{S}'_i$ (resp., $\mathcal{T}'_i$) be the leaves of T associated to $\mathcal{S}_i$ (resp., $\mathcal{T}_i$). Independently for each $q = 0, 1, \ldots, O(\log n)$ and for each node v in T, we execute the following procedure. Let T_v be the subtree rooted at v. With probability $\Theta(2^{-q})$, we q-mark v. If v is q-marked, we scale the capacities of T_v by a factor $O(2^q)$ and then run RoundGKR on the resulting scaled capacities for $O(\log n)$ iterations. The idea in the analysis is to consider a unit flow from $\mathcal{S}'_i$ to $\mathcal{T}'_i$, and a decomposition of this flow into flow paths. A fraction ϕ_v (possibly very small or zero) of this flow is carried by flow paths that *turn* at node v (meaning that the highest level node in those paths is v). For some value of q, if v is q-marked, then the scaled capacities support a unit flow from $\mathcal{S}'_i$ to $\mathcal{T}'_i$ passing through v, and hence RoundGKR has a large enough probability to connect $\mathcal{S}'_i$ to $\mathcal{T}'_i$ in at least one of the iterations. This happens also if we set to zero the edge capacities in T *corresponding to* a given set F of $O(\frac{k}{\log n})$ edges of G. Altogether, we obtain the following result.

THEOREM 1.4. *For any $k \geq 1$, there is a Las Vegas polynomial-time $(O(\log^4 n(\frac{\log h}{k}+1)), O(\log n))$-approximation algorithm for* SSC.

1.2 Related Work We have already mentioned parts of the known results on edge-connectivity problems for special cases of SSC. The node-connectivity version of SGST with $k = 2$ was studied by Khandekar, Kortsarz, and Nutov [23]. The authors showed that the problem admits an $O(g_{max} \cdot \log^2 n)$-approximation, where g_{max} is the maximum cardinality of any group. They also showed that a variant of the same problem on directed graphs is at least as hard as MLC. The node-cost version of GST was studied in [14] by Demaine, Hajiaghayi and Klein who gave an $O(\log n \operatorname{polylog}\log n)$-approximation algorithm for the special case where the input graph is planar and each group is the set of nodes on a face of the given planar embedding. No non-trivial algorithm is known for any variant of SGST for $k > 2$.

Another generalization of GST and SC is the *Set Connector* problem which has been studied in [8, 17]. Given collections of subsets of vertices $\{\mathcal{V}_i\}_{i=1}^q$, the goal is to connect each subset $X \in \mathcal{V}_i$ to another subset $Y \in \mathcal{V}_i$, for all collections V_i. The best known approximation ratio for this problem is $O(\log^2 n \log^2(qn))$ [8] and $O(\max_i \sum_{X\in\mathcal{V}_i} |X|)$ [17].

The approximability situation becomes less clear if one considers node-connectivity rather than edge-connectivity. The best known approximation factor for the node-connectivity version of Steiner Network, due to Chuzhoy and Khanna [12], is $O(k^3 \log n)$. A well studied special case is to find the min-cost spanning subgraph that is k-node connected; this special case admits a poly-logarithmic approximation in general and a 6-approximation when $k < n^{1/4}$ (see, e.g., [11, 15, 30]). Other special cases of node-connectivity problems have also been considered in literature. In particular, the rooted connectivity problem, where the requirements are between a root and terminals, has been studied in [6, 12, 27, 28], and the subset k-connectivity problem, where the requirements are on every pair of terminals, has been studied in [10, 25, 29].

1.3 Preliminaries

1.3.1 LP-Relaxations. We consider the following natural cut-based LP relaxation for SSC as shown in Figure 1. The variable x_e indicates whether edge e is selected in the solution. We denote the set of edges with exactly one endpoint in S (the edges *crossing* the cut S) by

$$\delta(S) = \delta_G(S) = \{vw \in E : v \in S, w \notin S\}.$$

The cut constraints intuitively say that any cut that separates $\mathcal{S}_i$ from $\mathcal{T}_i$ must have capacity at least k (treating x as capacities). Though the above LP has an exponential number of constraints, we can solve it in polynomial time since the corresponding separation problem can be reduced to the minimum cut problem. The LP-relaxation (SGST-LP) for SGST is the special case of the above relaxation with the restrictions $\mathcal{S}_i = \{r\}$ and $\mathcal{T}_i = \mathcal{G}_i$ for all i.

1.3.2 Räcke's Trees Let $G = (V, E)$ be an n-node undirected graph with edge capacities $x_e \geq 0$, $e \in E$. A Räcke's tree embedding of (G, x) consists of a triple (T, y, map), where $T = (V_T, E_T)$ is a rooted tree, y is a capacity function on the edges of T, and $\mathsf{map} : V_T \cup E_T \to V \cup 2^E$. For each tree-node $v \in V_T$, $\mathsf{map}(v) \in V$. Furthermore, map restricted to the leaves of T induces a bijection w.r.t. V (i.e., each leaf of T is in one-to-one correspondence with the nodes in V).

$$(\mathsf{SSC\text{-}LP}) \begin{cases} \min & \sum_{e \in E} c(e) \cdot x_e \\ \text{s.t.} & \sum_{e \in \delta(S)} x_e \geq k \quad \forall (\mathcal{S}_i, \mathcal{T}_i), \forall S \subseteq V : \mathcal{S}_i \subseteq S, \mathcal{T}_i \subseteq (V - S) \\ & 1 \geq x_e \geq 0 \qquad \forall e \in E \end{cases}$$

Figure 1: LP-relaxation for SSC

We will use $\mathsf{map}^{-1}(v)$ to indicate the leaf node $u \in V_T$ corresponding to node $v \in V$, and let $\mathsf{map}^{-1}(W) = \bigcup_{v \in W} \mathsf{map}^{-1}(v)$ for any $W \subseteq V$. Finally, for each tree-edge $e = uv \in E_T$, $\mathsf{map}(e) \subseteq E$ is a path in G between $\mathsf{map}(u)$ and $\mathsf{map}(v)$. For each tree-edge $f \in E_T$, one has $y_f = \sum_{e \in \delta_G(S)} x_e$, where $(S, V-S)$ is the bipartition of V which is induced by the leaves of the two connected components of $T - \{f\}$. For each edge $e \in E$, define $\mathsf{map}^{-1}(e) = \{f \in E_T : e \in \mathsf{map}(f)\}$, and let $\mathsf{map}^{-1}(F) = \cup_{e \in F} \mathsf{map}^{-1}(e)$ for any $F \subseteq E$. For a given tree embedding, the *load* on $e \in E$ is defined as $\mathsf{load}(e) = \sum_{f \in \mathsf{map}^{-1}(e)} y_f$, and the *relative load* is $\mathsf{rload}(e) = \frac{\mathsf{load}(e)}{x_e}$.

There is a natural way to map the flows back and forth between G and T. Let $\{f_i\}_i$ be a (fractional) multi-commodity flow in G, where $f_i : E \to \mathbb{R}^+$ is the flow for commodity i between nodes s_i and t_i. We simply route f_i units of flow along the unique path in T between $\mathsf{map}^{-1}(s_i)$ and $\mathsf{map}^{-1}(t_i)$. Let us call this flow $\{f_i^T\}_i$. On the other hand, given a multi-commodity flow $\{f_i^T\}_i$ on T, we convert it into a flow $\{f_i\}_i$ on G as follows. Let $f_i(e_T)$ be the value of f_i along edge $e_T \in E_T$. Then we route a flow of $f_i(e_T)$ along the path $\mathsf{map}(e_T)$. The next lemma is proved in [31].

LEMMA 1.1. ([31]) *Let (G, x) and (T, y, map) be defined as above. Given a feasible multi-commodity flow $\{f_i\}_i$ on G, the associated multi-commodity flow $\{f_i^T\}_i$ on T is also feasible. Given a feasible multi-commodity flow $\{f_i^T\}_i$ on T, the associated multi-commodity flow $\{f_i\}_i$ on G can be routed if each edge capacity x_e of G is multiplied by its relative load $\mathsf{rload}(e)$ in T.*

Räcke described in [31] a distribution $\mathcal{D}$ over tree embeddings (T, y, map) (which can be sampled in polynomial time) such that

$$\max_{e \in E} \{\mathbf{E}[\mathsf{rload}(e)]\} \leq \alpha = O(\log n).$$

The following property is implicit in Räcke's construction, and it will be crucial in our analysis.

LEMMA 1.2. *All the trees T in the support of the Räcke's tree distribution by [31] have height $O(\log(nC))$, where C is the ratio of the largest to smallest capacity in the graph G.*

Proof. (sketch) Shortly, the construction in [31] invokes as a subroutine the algorithm for the Minimum Cost Communication Tree problem (MCCT) in [16] by setting an appropriate length function ℓ on the edges of the graph. The latter construction is known to produce a tree of height $O(\log(nL))$, where L is the ratio of largest to smallest length in the graph. The length function used in [31] is of the form

$$\ell(f) = \frac{1}{x_f} \frac{e^{(M\lambda)_f}}{\sum_{f' \in E} e^{(M\lambda)_{f'}}},$$

where $M\lambda$ is a vector indexed by edges whose entries are, by construction, bounded by $O(\log n)$. Therefore, L is bounded by $O((nC)^{O(1)})$. ■

1.3.3 GKR Algorithm. Let us denote by RoundGKR one iteration of the rounding algorithm for GST on trees in [18]. RoundGKR works as follows. Let $T = (V_T, E_T)$ be the input tree that we think of as being rooted at r. For each edge $e \in E_T$, denote by $p(e) \in E_T$ the parent edge of e, if any. Let y be a given fractional solution to the standard cut-based LP for GST (i.e., SGST-LP with $k = 1$). Without loss of generality, we can assume that $y_e \leq y_{p(e)}$ (enforcing this constraint does not compromise the feasibility of the fractional solution). Initially, the solution set of edges is $\mathcal{S} = \emptyset$. We process edges $e \in E_T$ in non-decreasing order of their distances from r. When e is processed, we add e to $\mathcal{S}$ with probability $y_e / y_{p(e)}$ if $p(e)$ has been chosen in the solution; otherwise, e will not be added to $\mathcal{S}$. It can be seen (by induction on the order of edges) that each edge e is chosen with probability y_e: For each edge adjacent to the root, this is clear. Otherwise, we can apply the induction hypothesis and obtain

$$\begin{aligned} \mathbf{Pr}[e \in \mathcal{S}] &= \mathbf{Pr}[p(e) \in \mathcal{S}] \cdot \mathbf{Pr}[e \in \mathcal{S} | p(e) \in \mathcal{S}] \\ &= y_{p(e)} \cdot \frac{y_e}{y_{p(e)}} \\ &= y_e. \end{aligned}$$

The authors proved that each group $\mathcal{G}_i$ is connected to the root with probability at least $\Omega(1/\log n)$. Their analysis can be easily generalized to prove the following claim[2].

LEMMA 1.3. ([18]) *Suppose that, for some $F_T \subseteq E_T$, capacities y support a flow of value at least 1 from $\mathcal{G}_i$ to r in $T - F_T$. If we run RoundGKR on an instance with input tree T using the fractional solution y, then the solution returned by RoundGKR contains a path $P \subseteq T - F_T$ between $\mathcal{G}_i$ and r with probability at least $\Omega(1/\log n)$.*

2 Bicriteria Approximation for SSC

Algorithm 1 Algorithm for SSC.

Set $H \leftarrow \emptyset$.
Compute the optimal fractional solution x to SSC-LP, set to zero any $x_e < \frac{k}{n^2}$ and multiply each entry by $\frac{4}{k}$. Let x' be the resulting capacities.
Compute a distribution $\mathcal{D}$ of Räcke's tree embeddings for (G, x').
for τ rounds **do**
  Sample $(T = (V_T, E_T), y, \mathsf{map})$ from $\mathcal{D}$.
  for $q = 0, \ldots, 2\log_2 n$ and **for** each $v \in V_T$ **do**
    With probability $\min\{1, 4 \cdot 2^{-q}\}$, q-mark v.
    if v is q-marked **then**
      Set $y^{v,q} \leftarrow 2^{q+1} y$ and $E^{v,q} \leftarrow \emptyset$.
      for τ' iterations **do**
        $E^{v,q} \leftarrow E^{v,q} \cup \mathsf{RoundGKR}(T_v, v, y^{v,q})$.
      $H \leftarrow H \cup \mathsf{map}(E^{v,q})$.
return H.

2.1 Algorithm. Our bi-criteria approximation algorithm for SSC is Algorithm 1 in the figure. The algorithm initially solves SSC-LP and rounds the fractional solution in order to guarantee that set pairs have a (sufficiently large) constant fractional connectivity. Then it computes a Räcke's tree embedding (here we implicitly consider the construction in [31]). Subsequently, there is a sequence of τ *rounds*, where $\tau = O(\log h + k)$ is a parameter to be fixed later. In each round, the algorithm initially samples one Räcke's tree. Then, for each internal node v and for each value of an index q, with some probability (depending on q) the algorithm executes the following steps. It rounds the capacities of the subtree rooted at v (by a factor depending on q), and for τ' *iterations* it applies the subroutine RoundGKR (which returns a subset of selected tree edges) to the mentioned subtree using the mentioned rounded capacities. Here $\tau' = O(\log n)$ is another parameter to be fixed later. The final solution is the union of the *image* in the original graph of the tree edges selected by any execution of RoundGKR.

2.2 Analysis. We start by bounding the expected cost of the computed solution.

LEMMA 2.1. $\mathbf{E}[c(H)] = O(\log^4 n(\frac{\log h}{k}+1)) \cdot c(x)$, *where* $c(x) = \sum_{e \in E} c(e) x_e$.

Proof. We first analyze the probability that a given tree edge is selected in a given round. Let $(T = (V_T, E_T), y, \mathsf{map})$ be the corresponding tree embedding. Note that any $e \in E_T$ is selected iff it belongs to $E^{q,v}$ for some q and some (q-marked) ancestor $v \in V_T$ of e.

For each value of q and for each ancestor $v \in V_T$ of e, with probability at most $4 \cdot 2^{-q}$ node v is q-marked, in which case we perform τ' iterations of RoundGKR on T_v. In each such iteration, if any, edge e is selected with probability at most $y_e^{v,q} \le 2^{q+1} y_e$. Altogether, edge e is selected in the considered round with probability at most $\mathsf{height}(T) \cdot \sum_q 4 \cdot 2^{-q} \cdot 2^{q+1} y_e \cdot \tau' = O(\log^3 n) y_e$.

We next analyze the probability that an edge $f \in E$ is selected in the considered round. This happens iff RoundGKR selects any edge in $\mathsf{map}^{-1}(f) = \{e \in E_T : f \in \mathsf{map}(e)\}$. The latter probability is in expectation (over the distribution $\mathcal{D}$) at most

$$\begin{aligned} O(\log^3 n) \cdot \mathbf{E}\left[\sum_{e \in \mathsf{map}^{-1}(f)} y_e\right] &= O(\log^3 n) \cdot \mathbf{E}[\mathsf{load}(f)] \\ &= O(\log^3 n)\mathbf{E}[\mathsf{rload}(f) x'_f] \\ &\le O(\log^4 n) \cdot \frac{4}{k} \cdot x_f. \end{aligned}$$

Thus, the expected cost of the edges of G added to the solution in a given round is at most

$$O(\frac{\log^4 n}{k} \cdot \sum_{e \in E} c(e) x_e).$$

The claim follows since there are $\tau = O(\log h + k)$ rounds. ■

We next analyze the edge connectivity provided by the algorithm. We need the following lemma. Let $F \subseteq E$ be a given subset of edges. Consider a tree embedding (T, y, map) sampled from $\mathcal{D}$. Let $F_T = \mathsf{map}^{-1}(F) = \{e \in E_T : \mathsf{map}(e) \cap F \ne \emptyset\}$. We say that

[2] The main difference w.r.t. their analysis is that we can set the value of y_e to 0 for $e \in F$. This way we bound the probability to connect $\mathcal{G}_i$ to r without using edges of F. To get the claimed success probability of $\Omega(1/\log n)$, the only important property is that the residual flow between $\mathcal{G}_i$ and r supported by y is at least one.

the sampled tree embedding is *bad* for F if the total capacity $\sum_{e \in F_T} y_e$ of F_T w.r.t. y is at least $1/2$, and *good* otherwise. Intuitively, if a tree embedding is good, we can discard all the edges in F_T without decreasing the connectivity between set pairs too much. This is useful since, if a leaf-to-leaf path P in T uses no edges in F_T, the subgraph of G corresponding to P connects the endpoints of P without using any edge of F.

Lemma 2.2. *The probability that a tree embedding (T, y, map) sampled from $\mathcal{D}$ is bad for a collection F of at most $\frac{k}{16\alpha}$ edges is at most $\frac{1}{2}$.*

Proof. Let $\mathcal{B}$ denote the event in the claim. First, we notice that, if $\mathcal{B}$ happens, then it must be the case that

$$\sum_{e \in F} \mathsf{load}(e) = \sum_{e \in F} \sum_{f \in \mathsf{map}^{-1}(e)} y_f \geq \sum_{f \in F_T} y_f \geq \frac{1}{2}.$$

Therefore,

$$\mathbf{Pr}[\mathcal{B}] \leq \mathbf{Pr}[\sum_{e \in F} \mathsf{load}(e) \geq 1/2].$$

By the properties of $\mathcal{D}$,

$$\mathbf{E}[\mathsf{load}(e)] \leq \alpha \cdot x'_e \leq \frac{4\alpha}{k} x_e.$$

This implies that

$$\mathbf{E}[\sum_{e \in F} \mathsf{load}(e)] \leq \frac{4\alpha}{k}|F| \leq \frac{1}{4}.$$

By Markov's inequality,

$$\mathbf{Pr}[\sum_{e \in F} \mathsf{load}(e) \geq \frac{1}{2}] \leq \frac{1}{2}$$

as claimed. ■

We are now ready to bound the connectivity provided by the algorithm.

Lemma 2.3. *With probability at least $\frac{3}{4}$ all the set pairs are at least $\frac{k}{16\alpha} + 1$ edge connected in H.*

Proof. It is sufficient to prove that, for all F, $|F| \leq \frac{k}{16\alpha}$ and for all $i = 1, \ldots, h$, subgraph $H - F$ still connects $\mathcal{S}_i$ to $\mathcal{T}_i$. Consider any such pair (F, i). Define the event $\mathcal{A}_{F,i}$ as the event that "in a given round with sampled tree embedding (T, y, map), the algorithm selects some path P in T connecting $\mathcal{S}'_i := \mathsf{map}^{-1}(\mathcal{S}_i)$ with $\mathcal{T}'_i := \mathsf{map}^{-1}(\mathcal{T}_i)$, and P contains no edges in $F_T = \mathsf{map}^{-1}(F) = \{e \in E_T : \mathsf{map}(e) \cap F \neq \emptyset\}$". Note that, by definition, the edges $\mathsf{map}(P)$ in G connect $\mathcal{S}_i$ and $\mathcal{T}_i$ avoiding the edges in F. We will show below that $\mathbf{Pr}[\mathcal{A}_{F,i}] \geq \beta$ for some constant $\beta > 0$. Since we run the algorithm independently for τ rounds, the probability that $H - F$ does not connect $\mathcal{S}_i$ to $\mathcal{T}_i$ is at most $(1 - \beta)^\tau$. The claim follows from union bound by choosing $\tau = \frac{1}{\beta} \ln(4hn^{\frac{k}{16\alpha}}) = O(\log h + k)$ since there are at most $hn^{\frac{k}{16\alpha}}$ choices for (F, i).

By Lemma 2.2, we know that $\mathbf{Pr}[\mathcal{G}] \geq \frac{1}{2}$, where $\mathcal{G}$ denotes the event that the considered tree embedding is good. Let us condition on $\mathcal{G}$. Let y' be obtained from y by setting to zero the entries in F_T. Since the embedding is assumed to be good, y' supports a flow of at least $2 - 1/2 = 3/2$ between $\mathcal{S}'_i$ and $\mathcal{T}'_i$. Consider a unit flow supported by y' between $\mathcal{S}'_i$ and $\mathcal{T}'_i$, and a decomposition of such flow into flow paths $P_1, \ldots, P_z$, where we discard flow paths carrying a flow of value $\frac{1}{2n^2} \leq \frac{1}{2|E_T|}$ or smaller. Observe that each path P_j carries a flow $f_j > \frac{1}{2n^2}$, starts at some leaf node $s_j \in \mathcal{S}'_i$ goes up till some node v_j, and then goes down towards some leaf node $t_j \in \mathcal{S}'_i$ (we say that P_j *turns* at node v_j). Let ϕ_v be the total amount of flow turning at v. Note that

$$\sum_{v \in V_T} \phi_v = \sum_{j=1}^{z} f_j \geq \frac{3}{2} - \frac{1}{2n^2}|E_T| \geq 1.$$

For each node $v \in V_T$, let $q_v \in \{0, \ldots, 2\log_2 n\}$ be an integer such that $\phi_v \in (2^{-q_v-1}, 2^{-q_v}]$.

Consider the event $\mathcal{M}$ that some node $v \in V_T$ is q_v-marked. We claim that $\mathbf{Pr}[\overline{\mathcal{M}}|\mathcal{G}] \leq e^{-3/4}$. Let X_v be an indicator variable that is 1 if node $v \in V_T$ is q_v-marked and 0 otherwise. Observe that

$$\mathbf{Pr}\left[\overline{\mathcal{M}}|\mathcal{G}\right] = \mathbf{Pr}\left[\sum_{v \in V_T} X_v = 0\right].$$

The random variables X_v are independent, and

$$\mathbf{E}\left[\sum_{v \in V_T} X_v\right] = \sum_{v \in V_T} 4 \cdot 2^{-q_v} \geq \sum_{v \in V_T} 4 f_v \geq 4.$$

So, by Chernoff's bound,

$$\mathbf{Pr}\left[\sum_{v \in V_T} X_v \leq 1\right] \leq e^{-\frac{1}{3}(\frac{3}{4})^2 4} = e^{-3/4}.$$

Next, let us condition on $\mathcal{M}$, i.e., there is some node $v \in V_T$ that is q_v-marked. Consider the event $\mathcal{C}$ that $\mathcal{S}'_i$ and $\mathcal{T}'_i$ are connected by the union of the solutions computed by RoundGKR in the τ' iterations (in the considered round) on node v for $q = q_v$. Observe that the flow supported by scaled capacities y^{v,q_v} in T_v between v and $\mathcal{S}'_i$ (resp., $\mathcal{T}'_i$) is at least $2^{q_v+1}\phi_v \geq 1$. In that case, by Lemma 1.3, RoundGKR selects a path between v and $\mathcal{S}'_i$ (resp., $\mathcal{T}'_i$) with probability at least $1-(1-\Omega(\frac{1}{\log n}))^{\tau'}$ in some iteration. By choosing $\tau' = O(\log n)$ large enough, we can make the probability that this event does not happen smaller than any given constant $\varepsilon > 0$. As a consequence, $\mathbf{Pr}[\overline{\mathcal{C}}|\mathcal{G},\mathcal{M}] \le 2\varepsilon$. The claim follows since

$$\begin{aligned}\mathbf{Pr}[\mathcal{A}_{F,i}] &\ge 1-\mathbf{Pr}[\overline{\mathcal{G}}]-\mathbf{Pr}[\mathcal{G}]\cdot\mathbf{Pr}[\overline{\mathcal{M}}|\mathcal{G}] \\ &\quad -\mathbf{Pr}[\mathcal{G}]\cdot\mathbf{Pr}[\mathcal{M}|\mathcal{G}]\cdot\mathbf{Pr}[\overline{\mathcal{C}}|\mathcal{G},\mathcal{M}] \\ &\ge 1-\frac{1}{2}-e^{-3/4}-2\varepsilon \\ &=: \beta > 0.\end{aligned}$$

■

By Lemmas 2.1 and 2.3, the above algorithm is a Monte Carlo $(O(\log^4 n(\frac{1}{k}\log h+1)), O(\log n))$-approximation algorithm for SSC. Since we can check (in polynomial time) whether the computed solution has the desired cost and connectivity, the mentioned Monte Carlo algorithm can be easily turned into a Las Vegas one, thus proving Theorem 1.4.

2.2.1 The case of SGST. A similar but simpler approach works in the case of SGST, thus yielding a bicriteria approximation for the problem. We start from the optimal solution x to SGST-LP and scale it by $\frac{2}{k}$, thus obtaining edge capacities x'. We compute a distribution $\mathcal{D}$ of tree embeddings for the pair (G, x'). Then, for $O(\log n(\log h + k))$ iterations we sample a tree embedding (T, y, map) from $\mathcal{D}$, and run RoundGKR on tree T w.r.t. the fractional solution y. An analysis analogous to the SSC case proves Theorem 1.3.

3 Approximation Hardness of SGST

In this section, we present our hardness result for SGST by showing a reduction from MLC to SGST, thus proving Theorem 1.1. As SGST is a special case of SSC, this also implies the approximation hardness for SSC.

To simplify the discussion, we present the reduction for the *non-uniform* variant of SGST, where each group $\mathcal{G}_i \in \mathcal{G}$ has its own connectivity requirement k_i. As mentioned, this variant is equivalent to the uniform one.

We focus on a specific instance (Q, L, π) of MLC, where $Q = (A \cup B, F)$, and denote by $\mathsf{OPT}_{\mathsf{MLC}}$ its optimal cost. We show how to construct in polynomial time an instance $(G, c, \mathcal{G}, r, \{k_i\}_i)$ of (non-uniform) SGST such that its optimal cost $\mathsf{OPT}_{\mathsf{SGST}}$ satisfies $\mathsf{OPT}_{\mathsf{SGST}} = \mathsf{OPT}_{\mathsf{MLC}}$.

Our construction consists of two steps. In the first step, we construct a base instance G_{base} from the input instance of MLC. Every solution of the MLC instance can be mapped into a subgraph of G_{base} that contains a proper *canonical path* for each $ab \in F$.

In the second step, we replace some edges of the base instance with a *padding gadget*, and we define groups and connectivity requirements properly, hence obtaining an instance of SGST. Our construction ensures that a canonical path for each $ab \in F$ is indeed needed to obtain the desired connectivity.

3.0.2 Base Instance. We define a graph $G_{base} = (V_{base}, E_{base})$, where

$$\begin{aligned}V_{base} &= \{r\} \cup A \cup \mathcal{A} \cup B \cup \mathcal{B}, \\ \mathcal{A} &= \{a^\ell \mid a \in A, \ell \in L\} \text{ and} \\ \mathcal{B} &= \{b^\ell \mid b \in B, \ell \in L\}.\end{aligned}$$

For a given $a \in A$, we let $\mathcal{A}_a$ be the subset of nodes of type a^ℓ. Set $\mathcal{B}_b$ is defined analogously for each $b \in B$.

Set E_{base} contains all edges of type ra, aa^ℓ, and $b^\ell b$. Furthermore, for each edge $ab \in F$ and each $\ell \in L$, E_{base} contains the edges $a^\ell b^{\ell'}$ where $\ell' = \pi_{ab}(\ell)$. The edges e of type aa^ℓ and $b^\ell b$ have costs $c(e) = 1$ while all other edges have costs $c(e) = 0$.

The intuition behind this base construction is that a (canonical) path of type $r, a, a^\ell, b^{\ell'}, b$, with $\ell' = \pi_{ab}(\ell)$, corresponds to a labeling of a and b that satisfies edge $ab \in F$. So we wish a solution $H \subseteq G_{base}$ to include at least one such canonical path for every edge $ab \in F$. We next show how to modify G_{base}, and how to define groups and connectivity requirements, so that this property is enforced.

3.0.3 Padding Scheme. Now we construct the final graph G from the graph G_{base}. We replace each edge $a^\ell b^{\ell'}$, where $\ell' = \pi_{ab}(\ell)$, by a *padding gadget* H^ℓ_{ab}. The padding gadget contains nodes r, a^ℓ, $b^{\ell'}$, which are already in G_{base}, and seven new nodes $x^\ell_{ab}, y^{\ell r}_{ab}, y^{\ell a}_{ab}, y^{\ell b}_{ab}, z^{\ell r}_{ab}, z^{\ell a}_{ab}, z^{\ell b}_{ab}$. Furthermore, it contains all the edges of type $x^\ell_{ab} y^{\ell u}_{ab}$ and $y^{\ell u}_{ab} z^{\ell u}_{ab}$, $u \in \{r, a, b\}$, plus edges $r y^{\ell r}_{ab}$, $a^\ell y^{\ell a}_{ab}$, and $b^{\ell'} y^{\ell b}_{ab}$. All these edges $e \in H^\ell_{ab}$ have costs $c(e) = 0$. See Figure 2 for an illustration of the construction.

Then, for each edge $ab \in F$ (of the MLC instance), we define a group

$$\begin{aligned}\mathcal{G}_{ab} = \{b\} &\cup \{z^{\ell r}_{ab} \mid \ell \in L\} \\ &\cup \{z^{\ell a}_{a\tilde{b}} \mid a\tilde{b} \in F, \tilde{b} \in B - \{b\}, \ell \in L\} \\ &\cup \{z^{\ell b}_{\tilde{a}b} \mid \tilde{a}b \in F, \tilde{a} \in A - \{a\}, \ell \in L\}.\end{aligned}$$

Let $\deg_Q(v)$ denote the degree in Q of any $v \in A \cup B$. We set the (non-uniform) connectivity requirement of each group $\mathcal{G}_{ab}$ to be $k_{ab} = |L| \cdot (\deg_Q(a) + \deg_Q(b) - 1) + 1$. A path having a subpath of type

$(r, a, a^\ell, y^{\ell a}_{ab}, x^\ell_{ab}, y^{\ell b}_{ab}, b^{\ell'}, b)$ with $\ell' = \pi_{ab}(\ell)$ is called a *canonical path* for ab.

The next two lemmas prove that the SGST instance that we constructed is feasible, and that canonical paths are needed. The proof of Theorem 1.1 follows easily.

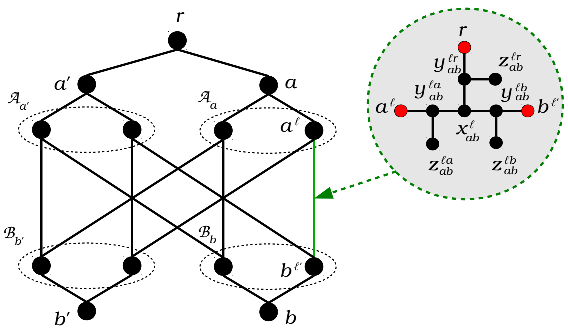

Figure 2: The padding gadget is shown on the top right. (some superscript ℓ and subscript ab are omitted) In the final construction, each edge $a^\ell b^{\ell'}$ is replaced with the padding gadget.

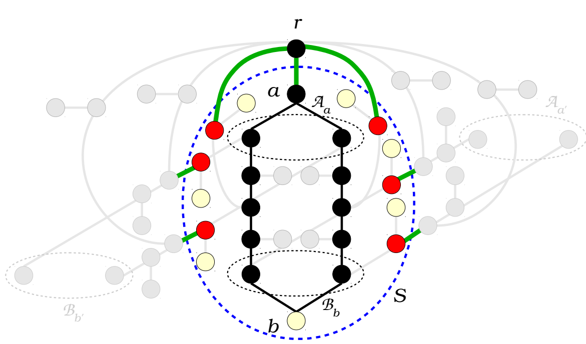

Figure 3: The blue circle depicts set S. The yellow nodes belong to $\mathcal{G}_{ab}$. The green edges are edges leaving S, and the red nodes are nodes of degree 3 that have been used by some other paths and thus cannot be used by the path p using edge ra.

Lemma 3.1. *For any $ab \in F$, there are k_{ab} edge-disjoint paths $\mathcal{P}$ in G from group $\mathcal{G}_{ab}$ to r.*

Proof. Let us choose $\ell_a, \ell_b \in L$ such that $\ell_b = \pi_{ab}(\ell_a)$. Consider the following sets of $r, \mathcal{G}_{ab}$-paths:

- $\mathcal{P}_1 = \{(r, y^{\ell r}_{ab}, z^{\ell r}_{ab}) \mid \ell \in L\}$
- $\mathcal{P}_2 = \{(r, y^{\ell r}_{a\tilde{b}}, x^\ell_{a\tilde{b}}, y^{\ell a}_{a\tilde{b}}, z^{\ell a}_{a\tilde{b}}) \mid a\tilde{b} \in F, \ell \in L, \tilde{b} \in B - \{b\}\}$
- $\mathcal{P}_3 = \{(r, y^{\ell r}_{\tilde{a}b}, x^\ell_{\tilde{a}b}, y^{\ell b}_{\tilde{a}b}, z^{\ell b}_{\tilde{a}b}) \mid \tilde{a}b \in F, \ell \in L, \tilde{a} \in A - \{a\}\}$
- $\mathcal{P}_4 = \{(r, a, a^{\ell_a}, y^{\ell_a a}_{ab}, x^{\ell_a}_{ab}, y^{\ell_a b}_{ab}, b^{\ell_b}, b)\}$

It is easy to check that they are edge disjoint and connect r to some node in $\mathcal{G}_{ab}$. One has $|\mathcal{P}_1| = |L|$, $|\mathcal{P}_2| = |L| \cdot (\deg_Q(a) - 1)$, $|\mathcal{P}_3| = |L| \cdot (\deg_Q(b) - 1)$, and $|\mathcal{P}_4| = 1$. Hence, the total number of paths is $|L| \cdot (\deg_Q(a) + \deg_Q(b) - 1) + 1 = k_{ab}$. ■

Lemma 3.2. *Given any collection $\mathcal{P}$ of k_{ab} edge-disjoint simple paths in G from $\mathcal{G}_{ab}$ to r, at least one path $p \in \mathcal{P}$ must be a canonical path for ab.*

Proof. Consider the set $\delta(S)$ of edges crossing the cut

$$\begin{aligned} S = \{a, b\} &\cup \mathcal{A}_a \cup \mathcal{B}_b \\ &\cup \{x^\ell_{ab}, y^{\ell r}_{ab}, y^{\ell a}_{ab}, y^{\ell b}_{ab}, z^{\ell r}_{ab}, z^{\ell a}_{ab}, z^{\ell b}_{ab} \mid \ell \in L\} \\ &\cup \{y^{\ell a}_{a\tilde{b}}, z^{\ell a}_{a\tilde{b}} \mid a\tilde{b} \in F, \ell \in L, \tilde{b} \in B - \{b\}\} \\ &\cup \{y^{\ell b}_{\tilde{a}b}, z^{\ell b}_{\tilde{a}b} \mid \tilde{a}b \in F, \ell \in L, \tilde{a} \in A - \{a\}\} \end{aligned}$$

Set $\delta(S)$ contains precisely k_{ab} edges, namely:

- $|L|$ edges of type $ry^{\ell r}_{ab}$,
- $|L| \cdot (\deg_Q(a) - 1)$ edges of type $y^{\ell a}_{a\tilde{b}} x^\ell_{a\tilde{b}}$,
- $|L| \cdot (\deg_Q(b) - 1)$ edges of type $y^{\ell b}_{\tilde{a}b} x^\ell_{\tilde{a}b}$, and
- edge ra.

Therefore, each edge of $\delta(S)$ belongs to exactly one path in $\mathcal{P}$ (see Figure 3). We claim that the path $p \in \mathcal{P}$ using edge ra is a canonical path for ab. W.l.o.g., we assume that p is simple.

To prove the claim, we will crucially use the fact that any node with degree 3 can be an internal node of at most one path in $\mathcal{P}$. This implies that p cannot contain these nodes:

1. Nodes of type $y^{\ell r}_{ab}, y^{\ell a}_{a\tilde{b}}, y^{\ell b}_{\tilde{a}b}$ (these nodes are incident to edges of $\delta(S)$ which must be used by other paths in $\mathcal{P}$).

2. Nodes of type $z^{\ell a}_{ab}$ and $z^{\ell b}_{ab}$ (these nodes have degree 1 and are not contained in $\mathcal{G}_{ab}$).

Next consider the nodes of p in increasing distance from r. By the above argument, the 2nd node is a, and the 3rd one has to be of type aa^{ℓ} for some $\ell \in L$. The 4th node cannot be of type $y^{\ell a}_{a\tilde{b}}$ for $\tilde{b} \neq b$ by (1). Therefore, it has to be $y^{\ell a}_{ab}$. The 5th node cannot be $z^{\ell a}_{ab}$ by (2); hence, it has to be x^{ℓ}_{ab}. The 6th node cannot be $y^{\ell r}_{ab}$ by (1), so it has to be $y^{\ell b}_{ab}$. Since $z^{\ell b}_{ab}$ cannot belong to p by (2), the 7th node must be $b^{\pi_{ab}(\ell)}$. The 8th node cannot be of type $y^{\ell b}_{\tilde{a}b}$, $\tilde{a} \neq a$, by (1), so it has to be either b (in which case the claim holds) or $y^{\tilde{\ell} b}_{ab}$ where $\pi_{ab}(\tilde{\ell}) = \pi_{ab}(\ell)$. But the second case cannot happen since, by (1), (2) and the fact that π_{ab} is a projection, either p would not be simple or it should continue with nodes $x^{\tilde{\ell}}_{ab}$ and $y^{\tilde{\ell} a}_{ab}$, and end at $a^{\tilde{\ell}}$ (not in $\mathcal{G}_{ab}$). ∎

Proof. (of Theorem 1.1) Consider the above (polynomial-time) construction of an SGST instance. It is easy to check that its size is $O(m)$ and that $k \leq |L|(2\Delta - 1) + 1 = O(|L|\Delta)$. Let $\mathsf{OPT}_{\mathsf{SGST}}$ and $\mathsf{OPT}_{\mathsf{MLC}}$ be the optimal costs of the SGST and MLC instances, respectively.

Let us prove $\mathsf{OPT}_{\mathsf{SGST}} \leq \mathsf{OPT}_{\mathsf{MLC}}$. Let $\mathcal{L}^*$ be a minimum cost labeling of the MLC instance. Consider the SGST solution $\mathcal{S}^*$ consisting of all the edges of cost 0, plus all the edges of type aa^{ℓ} and $b^{\ell'}b$ such that $\ell \in \mathcal{L}(a)$ and $\ell' \in \mathcal{L}(b)$. Clearly, the cost of $\mathcal{S}^*$ is $\mathsf{OPT}_{\mathsf{MLC}}$. We now argue that $\mathcal{S}^*$ is feasible. Consider any group $\mathcal{G}_{ab}$. In the proof of Lemma 3.1, the only edges of cost 1 used in the construction of a feasible set of paths are any two edges aa^{ℓ_a} and $b^{\ell_b}b$ such that $\ell_b = \pi_{ab}(\ell_a)$. Such a pair of edges must be present in $\mathcal{S}^*$ since otherwise $\mathcal{L}^*$ would not cover edge ab.

To prove the converse (i.e., $\mathsf{OPT}_{\mathsf{SGST}} \geq \mathsf{OPT}_{\mathsf{MLC}}$), let E^* be the set of edges used by the optimal SGST solution. Consider the following labeling $\mathcal{L}$: $\mathcal{L}(a) = \{\ell \mid aa^{\ell} \in E^*\}$ and $\mathcal{L}(b) = \{\ell \mid b^{\ell}b \in E^*\}$ for all $a \in A$ and $b \in B$. Clearly, $c(\mathcal{L}) = OPT_{\mathsf{SGST}}$. Let us now show that $\mathcal{L}$ is feasible, i.e., any edge $ab \in F$ is covered by $\mathcal{L}$. Let $\mathcal{P}$ be any k edge disjoint paths connecting r to $\mathcal{G}_{ab}$. By Lemma 3.2, some path $p \in \mathcal{P}$ contains a subpath of the form $(r, a, a^{\ell}, y^{\ell a}_{ab}, x^{\ell}_{ab}, y^{\ell b}_{ab}, b^{\ell'}, b)$, where $\ell' = \pi_{ab}(\ell)$ by the definition of G_{base}. By construction, $\ell \in \mathcal{L}(a)$ and $\ell' \in \mathcal{L}(b)$. Hence, edge ab is covered. ∎

4 Bicriteria Hardness for Survivable Group Steiner Tree

In this section, we present the bicriteria hardness result for SGST (and hence SSC), thus proving Theorem 1.2. Our result is based on the hardness of GST by Halperin and Krauthgamer [21] as in the following theorem.

THEOREM 4.1. ([21]) *Unless* NP *has a quasi-polynomial time randomized algorithm, there is no polynomial time $O(\log^{2-\varepsilon} n)$ approximation algorithm for* GST*, for any $\varepsilon > 0$, even when the input graph is a tree.*

The key idea is that one can construct an SGST instance, with $k = \beta$, by making β copies of a GST instances and unifying the root node. Any (α, β) approximation algorithm for SGST applied to the mentioned instance would compute a solution H of cost at most $\alpha \cdot \beta$ times the optimal GST cost, and H would induce a feasible GST solution (since $k/\beta = 1$).

Now we proceed to the formal proof.

Proof. (of Theorem 1.2) Assume by contradiction that the claim is not true, and let $\mathcal{A}$ be an (α, β)-approximation algorithm that contradicts the claim. We show how to derive a polynomial-time $O(\log^{2-\varepsilon} n)$ approximation algorithm for GST, hence contradicting Theorem 4.1.

Let $\mathcal{S} = (T, c, \mathcal{G}, r)$ be the input instance of GST, where T is a tree, of optimal cost OPT_{GST}. (W.l.o.g. $\mathcal{S}$ is feasible). We construct an instance $\mathcal{S}' = (T', c', \mathcal{G}', r', k)$ of SGST, with $k = \beta$ and T' a tree, as follows. For each $i = 1, \ldots, \beta$, we create a copy T^i of T; let r_i be the copy of r in T^i. Then we create a super root r' which is adjacent to each r_i via zero-cost edges. To define the groups, for each $\mathcal{G}_j \in \mathcal{G}$, let $\mathcal{G}_{j,i} \subseteq V(T^i)$ denote the corresponding subset of nodes in T^i. We define $\mathcal{G}'_j = \bigcup_{i=1}^{\beta} \mathcal{G}_{j,i}$ and add $\mathcal{G}'_j$ to our collection of groups $\mathcal{G}'$. Notice that $|\mathcal{G}'| = |\mathcal{G}| =: h$. Observe also that each group is β-connected to r' in T'. Finally, it is easy to see that the optimal cost $\mathsf{OPT}_{\mathsf{SGST}}$ of $\mathcal{S}'$ is at most $\beta \cdot \mathsf{OPT}_{\mathsf{GST}}$ (just include the edges incident to r plus a copy of the optimal solution to $\mathcal{S}$ in each T^i).

Let us apply algorithm $\mathcal{A}$ to $\mathcal{S}'$, hence obtaining a solution APX' that is guaranteed to be an (α, β)-approximation. Consider the solution APX to the GST instance where we include an edge e in APX iff some copy e' of e appears in the solution APX'. Observe that APX is a feasible solution to $\mathcal{S}$; indeed, APX' must connect each group to the root (since $\frac{k}{\beta} = 1$) in at least one copy. Therefore, for each j, there is a path in APX' between $\mathcal{G}_{j,i}$ and r_i for some i. Thus, APX is a feasible GST solution for $\mathcal{S}$. Also, $c(\mathsf{APX}) \leq c(\mathsf{APX}') \leq \alpha \cdot \mathsf{OPT}_{\mathsf{SGST}} \leq \alpha \cdot \beta \cdot \mathsf{OPT}_{\mathsf{GST}} \leq O(\log^{2-\varepsilon} n) \cdot \mathsf{OPT}_{\mathsf{GST}}$. ∎

5 Discussions and Open Problems

We have presented a bicriteria approximation algorithm for SSC with complementary hardness results. Our negative result refutes poly-logarithmic approximation factors for standard (non-bicriteria) approximation algorithms. Indeed, assuming the conjecture that MLC has polynomial hardness, we could rule out sub-polynomial approximations for SSC. In contrast, our bicriteria approximation gives a poly-logarithmic guarantee on the cost (with a loss of an $O(\log n)$ factor in the connectivity). It is surprising that we could obtain a cost-guarantee far beyond the hardness bound.

Our techniques can be extended to obtain upper and lower bounds on the approximability of a generalization of SSC with degree-bounds on the nodes. We postpone the details to the journal version of the paper.

An interesting question is whether one could obtain poly-logarithmic bicriteria approximations for other network design problems whose hardness are derived from MLC.

Another natural question following our results is to design a (non-trivial) non-bicriteria polynomial approximation algorithm for SSC when the maximum connectivity requirement is $k \geq 3$. To the best of our knowledge, this question remains open. Also, it would be interesting to see whether SSC admits a non-bicriteria poly-logarithmic approximation when k is bounded by a constant, e.g., when $k = 3$. For $k = O(1)$, the best known hardness for SSC (and SGST) is $\log^{2-\epsilon} n$ whereas our result only gives a k^{δ}-hardness when k is a large enough constant. It might be possible that the hardness depends on both k and n. A less ambitious still very interesting goal is whether it is possible to achieve a bicriteria $(f(k)\,\mathrm{polylog}(nh), O(1))$-approximation algorithm.

There are several natural variants of SSC that one might consider. For example, one might ask for node-connectivity rather than edge-connectivity.

All the above questions also make sense in the special case of SGST.

References

[1] Noga Alon, Baruch Awerbuch, Yossi Azar, Niv Buchbinder, and Joseph Naor. A general approach to online network optimization problems. *ACM Transactions on Algorithms*, 2(4):640–660, 2006. 1

[2] Reid Andersen and Uriel Feige. Interchanging distance and capacity in probabilistic mappings. *CoRR*, abs/0907.3631, 2009. 3

[3] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, 1998. 2

[4] Yair Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *37th Annual Symposium on Foundations of Computer Science, FOCS '96, Burlington, Vermont, USA, 14-16 October, 1996*, pages 184–193, 1996. 3

[5] Jaroslaw Byrka, Fabrizio Grandoni, Thomas Rothvoß, and Laura Sanità. Steiner tree approximation via iterative randomized rounding. *J. ACM*, 60(1):6, 2013. 2

[6] Tanmoy Chakraborty, Julia Chuzhoy, and Sanjeev Khanna. Network design for vertex connectivity. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 167–176, 2008. 2, 4

[7] Moses Charikar, Chandra Chekuri, Ashish Goel, and Sudipto Guha. Rounding via trees: Deterministic approximation algorithms for group steiner trees and k-median. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 114–123, 1998. 2

[8] Chandra Chekuri, Guy Even, Anupam Gupta, and Danny Segev. Set connectivity problems in undirected graphs and the directed steiner network problem. *ACM Transactions on Algorithms*, 7(2):18, 2011. 1, 4

[9] Chandra Chekuri, Guy Even, and Guy Kortsarz. A greedy approximation algorithm for the group steiner problem. *Discrete Applied Mathematics*, 154(1):15–34, 2006. 2

[10] Joseph Cheriyan, Bundit Laekhanukit, Guyslain Naves, and Adrian Vetta. Approximating rooted steiner networks. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 1499–1511, 2012. 2, 4

[11] Joseph Cheriyan and László A. Végh. Approximating minimum-cost k-node connected subgraphs via independence-free graphs. *SIAM J. Comput.*, 43(4):1342–1362, 2014. 4

[12] Julia Chuzhoy and Sanjeev Khanna. An O(k^{3}log n)-approximation algorithm for vertex-connectivity survivable network design. *Theory of Computing*, 8(1):401–413, 2012. 4

[13] Julia Chuzhoy, Yury Makarychev, Aravindan Vijayaraghavan, and Yuan Zhou. Approximation algorithms and hardness of the k-route cut problem. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 780–799, 2012. 2

[14] Erik D. Demaine, MohammadTaghi Hajiaghayi, and Philip N. Klein. Node-weighted steiner tree and group steiner tree in planar graphs. In *Automata, Languages and Programming, 36th International Colloquium, ICALP 2009, Rhodes, Greece, July 5-12, 2009, Proceedings, Part I*, pages 328–340, 2009. 4

[15] Jittat Fakcharoenphol and Bundit Laekhanukit. An

$O(\log^2 k)$-approximation algorithm for the k-vertex connected spanning subgraph problem. *SIAM J. Comput.*, 41(5):1095–1109, 2012. 4

[16] Jittat Fakcharoenphol, Satish Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *J. Comput. Syst. Sci.*, 69(3):485–497, 2004. 3, 5

[17] Takuro Fukunaga and Hiroshi Nagamochi. The set connector problem in graphs. In *Integer Programming and Combinatorial Optimization, 12th International IPCO Conference, Ithaca, NY, USA, June 25-27, 2007, Proceedings*, pages 484–498, 2007. 1, 4

[18] Naveen Garg, Goran Konjevod, and R. Ravi. A polylogarithmic approximation algorithm for the group steiner tree problem. *J. Algorithms*, 37(1):66–84, 2000. 2, 3, 5, 6

[19] Michel X. Goemans, Andrew V. Goldberg, Serge A. Plotkin, David B. Shmoys, Éva Tardos, and David P. Williamson. Improved approximation algorithms for network design problems. In *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms. 23-25 January 1994, Arlington, Virginia.*, pages 223–232, 1994. 2

[20] Anupam Gupta, Ravishankar Krishnaswamy, and R. Ravi. Tree embeddings for two-edge-connected network design. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 1521–1538, 2010. 2

[21] Eran Halperin and Robert Krauthgamer. Polylogarithmic inapproximability. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing, June 9-11, 2003, San Diego, CA, USA*, pages 585–594, 2003. 2, 10

[22] Kamal Jain. A factor 2 approximation algorithm for the generalized Steiner network problem. *Combinatorica*, 21(1):39–60, 2001. 2

[23] Rohit Khandekar, Guy Kortsarz, and Zeev Nutov. Approximating fault-tolerant group-steiner problems. *Theor. Comput. Sci.*, 416:55–64, 2012. 2, 4

[24] Guy Kortsarz, Robert Krauthgamer, and James R. Lee. Hardness of approximation for vertex-connectivity network design problems. *SIAM J. Comput.*, 33(3):704–720, 2004. 2, 3

[25] Bundit Laekhanukit. An improved approximation algorithm for the minimum cost subset k-connected subgraph problem. *Algorithmica*, pages 1–20, 2014. 4

[26] Bundit Laekhanukit. Parameters of two-prover-one-round game and the hardness of connectivity problems. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 1626–1643, 2014. 2

[27] Zeev Nutov. A note on rooted survivable networks. *Inf. Process. Lett.*, 109(19):1114–1119, 2009. 4

[28] Zeev Nutov. Approximating minimum-cost connectivity problems via uncrossable bifamilies. *ACM Transactions on Algorithms*, 9(1):1, 2012. 4

[29] Zeev Nutov. Approximating subset k-connectivity problems. *J. Discrete Algorithms*, 17:51–59, 2012. 4

[30] Zeev Nutov. Approximating minimum-cost edge-covers of crossing biset-families. *Combinatorica*, 34(1):95–114, 2014. 4

[31] Harald Räcke. Optimal hierarchical decompositions for congestion minimization in networks. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 255–264, 2008. 3, 5, 6

[32] Ran Raz. A parallel repetition theorem. *SIAM J. Comput.*, 27(3):763–803, 1998. 2

[33] Aravind Srinivasan. New approaches to covering and packing problems. In *Proceedings of the Twelfth Annual Symposium on Discrete Algorithms, January 7-9, 2001, Washington, DC, USA.*, pages 567–576, 2001. 2

[34] David P. Williamson, Michel X. Goemans, Milena Mihail, and Vijay V. Vazirani. A primal-dual approximation algorithm for generalized Steiner network problems. *Combinatorica*, 15(3):435–454, 1995. Preliminary version in STOC 1993. 2