Greedy Algorithms for Online Survivable Network Design

Sina Dehghani
University of Maryland, College Park, MD 20742, USA

Soheil Ehsani
University of Maryland, College Park, MD 20742, USA

MohammadTaghi Hajiaghayi
University of Maryland, College Park, MD 20742, USA

Vahid Liaghat
Facebook, Building 25, 190 Jefferson Dr, Menlo Park, CA 94025, USA

Saeed Seddighin
University of Maryland, College Park, MD 20742, USA

Abstract

In an instance of the *network design* problem, we are given a graph $G = (V, E)$, an edge-cost function $c : E \to \mathbb{R}^{\geq 0}$, and a connectivity criterion. The goal is to find a minimum-cost subgraph H of G that meets the connectivity requirements. An important family of this class is the *survivable network design problem* (SNDP): given non-negative integers r_{uv} for each pair $u, v \in V$, the solution subgraph H should contain r_{uv} edge-disjoint paths for each pair u and v.

While this problem is known to admit good approximation algorithms in the offline case, the problem is much harder in the online setting. Gupta, Krishnaswamy, and Ravi [14] (STOC'09) are the first to consider the online survivable network design problem. They demonstrate an algorithm with competitive ratio of $O(k \log^3 n)$, where $k = \max_{u,v} r_{uv}$. Note that the competitive ratio of the algorithm by Gupta *et al.* grows linearly in k. Since then, an important open problem in the online community [22, 14] is whether the linear dependence on k can be reduced to a logarithmic dependency.

Consider an online greedy algorithm that connects every demand by adding a minimum cost set of edges to H. Surprisingly, we show that this greedy algorithm significantly improves the competitive ratio when a congestion of 2 is allowed on the edges or when the model is stochastic. While our algorithm is fairly simple, our analysis requires a deep understanding of k-connected graphs. In particular, we prove that the greedy algorithm is $O(\log^2 n \log k)$-competitive if one satisfies every demand between u and v by $r_{uv}/2$ edge-disjoint paths. The spirit of our result is similar to the work of Chuzhoy and Li [7] (FOCS'12), in which the authors give a polylogarithmic approximation algorithm for edge-disjoint paths with congestion 2.

Moreover, we study the greedy algorithm in the online stochastic setting. We consider the i.i.d. model, where each online demand is drawn from a single probability distribution, the unknown i.i.d. model, where every demand is drawn from a single but unknown probability distribution, and the *prophet* model in which online demands are drawn from (possibly) different probability distributions. Through a different analysis, we prove that a similar greedy algorithm is constant competitive for the i.i.d. and the prophet models. Also, the greedy algorithm is $O(\log n)$-competitive for the unknown i.i.d. model, which is almost tight due to the lower bound of [9] for single connectivity.

2012 ACM Subject Classification Theory of computation → Online algorithms

Keywords and phrases survivable network design, online, greedy

Digital Object Identifier 10.4230/LIPIcs.ICALP.2018.152

45th International Colloquium on Automata, Languages, and Programming (ICALP 2018).
Editors: Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella;
Article No. 152; pp. 152:1–152:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Funding Supported in part by NSF CAREER award CCF-1053605, NSF BIGDATA grant IIS-1546108, NSF AF:Medium grant CCF-1161365, DARPA GRAPHS/AFOSR grant FA9550-12-1-0423, and another DARPA SIMPLEX grant. Portions of this research were completed while the authors were visitors at the Simons Institute for the Theory of Computing.

Acknowledgements The authors would like to thank Seffi Naor for several fruitful discussions about the prophet versions of the online problems.

1 Introduction

In an instance of the *network design* problem, we are given a graph $G = (V, E)$, an edge-cost function $c : E \to \mathbb{R}^{\geq 0}$, and a connectivity criteria. The goal is to find a minimum-cost subgraph H of G that satisfies the connectivity requirements. An important family of this class is the *survivable network design problem* (SNDP): Given non-negative integers r_{uv} for each pair $u, v \in V$, the solution subgraph H should contain r_{uv} edge-disjoint paths for each pair u and v. SNDP arises in fault tolerance management and thus is of much interest in design community: the connectivity of nodes u and v in H is resilient to even $(r_{uv} - 1)$ edge failures. This problem clearly generalizes the *Steiner tree*[1] and *Steiner forest*[2] problems.

For a non-empty cut $S \subset V$, let $\delta(S)$ denote the set of edges with exactly one endpoint in S. SNDP falls in the general class of network design problems that can be characterized by *proper cut functions*. A function $f : 2^V \to \mathbb{Z}^{\geq 0}$ defined over cuts in the graph is *proper*, if it is symmetric ($f(S) = f(V \setminus S)$ for all $S \subset V$) and it satisfies maximality ($f(S \cup T) \leq \max\{f(S), f(T)\}$ for all $S \cap T = \phi$). For SNDP, one can choose $f(S) = \max_{u \in S, v \notin S} r_{uv}$ for every cut S. Given a proper function f over cuts in the graph, the goal is to find a minimum-cost subgraph H such that

$$|E(H) \cap \delta(S)| \geq f(S) \qquad \forall \textit{non-empty } S \subset V \ .$$

Over the past decades, the offline SNDP and proper cut functions have been extensively studied especially as an important testbed for primal-dual and iterative rounding methods (see e.g. [10, 11, 13, 18, 26, 27]). In this paper, we consider SNDP in the *online* setting: we are given a graph $G = (V, E)$ and an edge-cost function c in advance. We receive an online sequence of demands in the form of tuples $(u, v, r_{uv}) \in V \times V \times \mathbb{Z}^{\geq 0}$. We start with an empty subgraph H. Upon the arrival of a demand (u, v, r_{uv}), we need to immediately *augment* H such that there exist at least r_{uv} edge-disjoint paths between u and v in H. The goal is to minimize the cost of H. The *competitive ratio* of an algorithm is defined as the maximum ratio of the cost of its output and that of an optimal offline solution, over all possible input instances.

The online 1-connectivity problems, in which $r_{uv} \in \{0, 1\}$ for all pairs, have been extensively studied in the last decades. Imase and Waxman [17] (SIAM'91) were first to consider the edge-weighted Steiner tree problem. They used a dual-fitting argument to show that the natural greedy algorithm is $O(\log n)$-competitive where n denotes the number of vertices[3]. Their result is asymptotically tight. Later, Berman and Coulston [3] (STOC'97) and Awerbuch, Azar, and Bartal [2] (TCS'04) demonstrated an $O(\log n)$-competitive algorithm for

[1] In the Steiner tree problem, given a set of terminal nodes $T \subset V$, the goal is to find a minimum-cost subgraph connecting all terminals.

[2] In the Steiner forest problem, given a set of pairs of vertices $s_i, t_i \in V$, the goal is to find a minimum-cost subgraph in which every pair is connected.

[3] In fact, the competitive ratio is $O(\log \min\{n, \mathcal{D}\})$ where $\mathcal{D}$ is the number of demand requests. However,

the more general Steiner forest problem by designing an elegant online primal-dual technique. The latter also shows that the greedy algorithm achieves the competitive ratio of $O(\log^2 n)$ for Steiner forest. Indeed, due to the simplicity of greedy approaches, an important open problem is to settle the competitiveness of the greedy algorithm for Steiner forest. In the recent years, several primal-dual techniques are developed for solving node-weighted variants [1, 15, 22], and prize-collecting variants [25, 16] of 1-connectivity problems.

Gupta, Krishnaswamy, and Ravi [14] (SIAM'12) were first to consider the online survivable network design problem. They demonstrate an elegant algorithm with competitive ratio of $\tilde{O}(k \log^3 n)$, where $k = \max_{u,v} r_{uv}$. The crux of their analysis is to use distance-preserving tree-embeddings in an online setting. More precisely, they first pick a random distance-preserving spanning subtree $T \subseteq G$. They satisfy a connectivity demand r_{uv} by iteratively increasing the connectivity of u and v. In each iteration, they show that it is sufficient to use cycles that are formed by an edge $e = (a, b) \notin T$ and the $\{a, b\}$-path in T; hence, reducing the number of *options* for satisfying a connectivity demand. This would enable them to use a set cover approach to solve the problem in an online manner and achieve the first competitive algorithm for online SNDP.

Single-source SNDP is a variant of SNDP where all demands share the same endpoint. Naor, Panigrahi, and Singh [22] (FOCS'11) partially improve the results of Gupta et al. [14] by demonstrating a *bi-criteria* competitive algorithm for single-source SNDP using structural properties of a single-source optimal solution. A bi-criteria competitive ratio of (α, β) for SNDP implies that the solution produced by the online algorithm achieves a connectivity of $\lfloor \frac{r_i}{\beta} \rfloor$ for every demand σ_i and is at most a factor of α more expensive than the optimal offline solution for connectivity of r_i. The algorithm by Naor et al. achieves the competitive ratio of $(O(\frac{k \log n}{\epsilon}), 2 + \epsilon)$ for any $\epsilon > 0$. They also study and give bi-criteria algorithms for the vertex-connectivity problem.

The competitive ratio of algorithms by Gupta et al. and Naor et al. grow linearly in k. This seems to be inherent to their methods since they may need to solve a set-cover-like problem in each iteration of incrementing the connectivity of a demand; hence, losing a polylogarithmic factor in each iteration. One would need a new approach to break the linear dependency on k. Indeed, both factors of $O(\log^3 n)$ and $O(k)$ are not plausible in practice, and an important open problem in the online community [22, 14] is whether the linear dependency on k can be reduced to a logarithmic dependency.

We circumvent this problem within the class of greedy algorithms. This class has been a center of attention in many applications due to its simplicity. We would like to note that the previous algorithms for the problem are based on fairly complex techniques such tree-embedding and reductions to online set cover. Despite their theoretical provable guarantees, these approaches are not efficiently implementable for large networks and become extremely harder to analyze when more parameters of the system are involved, e.g. degree constraints. For these reasons, we are interested in the theory behind simple algorithms for the online SNDP. In this paper, we study both the cases with and without the presence of priori information about connectivity demands.

No Information Setting: for this traditional online setting, we show that the greedy approach is promising although the *classic greedy algorithm*[4] fails to give a competitive

to simplify the comparison with results for SNDP, in this paper we ignore the distinction between this factor and $O(\log n)$.

[4] Which augments the solution with the cheapest set of edges that satisfy full connectivity demands.

solution. In Section 2, we give a hardness instance for the classic greedy algorithm and present our modified versions of it which have polylogarithmic competitive ratio. In particular, we demonstrate a deterministic algorithm with a bi-criteria competitive ratio of $(O(\log^2 n \log_{1+\epsilon} k), 2+\epsilon)$ for any constant $\epsilon > 0$. For the single-rooted variant, the competitive ratio is $(O(\log n \log_{1+\epsilon} k), 2+\epsilon)$. Besides, our hardness instance shows a loss of $\Omega(n)$ on the weight criterion if satisfying full connectivities greedily.

Partial Information Setting: one of the recent trends in the study of online problems is to consider a stochastic model for the online demands. This, in particular, is to model the scenarios in which the algorithm designer has sufficient data available in hand to be able to make predictions, i.e. fit distributions, for future requests. In this paper, we consider an online stochastic model inspired by the well-known *prophet inequality* problem[5] in which different demands are drawn independently from different distributions. We call it the *prophet setting* not only to highlight the similarity between this online stochastic model and prophet inequalities but also to distinguish it from other online stochastic models[6].

In the second half of this paper, we study online SNDP in the prophet setting for both cases of known and unknown distributions. For the most general case in prophet setting in which the distributions are known and (possibly) different, we show that the classic greedy algorithm that satisfies full connectivity results in a constant approximation solution. Furthermore, we explore the connection between the *known i.i.d.* case and the general case and present a framework that transforms an algorithm for the former into one for the latter by loosing only a constant factor on the approximation ratio. Lastly but more interestingly, we prove that the classic greedy algorithm is $O(\log n)$ approximation[7] for the *unknown i.i.d.* case[8]. This result shows that a simple algorithm can significantly outperform the previous complex algorithms when the demands are drawn from the same distribution without even learning the underlying distribution!

1.1 Our Results and Techniques

Let $\sigma_i = (u_i, v_i, r_i)$ denote the i-th connectivity demand. Consider the following intuitive greedy approach. Upon the arrival of σ_i, we augment the solution subgraph H, by finding the minimum-cost set of edges whose addition to H creates r_i edge-disjoint paths between u_i and v_i. Awerbuch et al. [2] (TCS'04) show that if all the demands require 1-connectivity (i.e., $r_i = 1$ for every i), this algorithm achieves a competitive ratio of $O(\log^2 n)$. This leads to a natural question that whether greedy works for higher connectivity problems as well. However, we show an instance of online SND in Section 2, for which the greedy algorithm has a competitive ratio of $\Omega(n)$. Indeed, the connectivity demands in the instance are either zero or two, hence greedy is not competitive even for low connectivity demands. However, on the positive side, we show that *greedy-like* algorithms do surprisingly well in both the stochastic version of the problem and the case when a small congestion on the edges is acceptable.

[5] Given n distributions $D_1, \ldots, D_n$ on real numbers and an online sequence of random draws $X_i \sim D_i$, we have to make an immediate and irrevocable selection X_τ that maximizes the ratio $\mathbb{E}[X_\tau]/\mathbb{E}[\max_{1\le i\le n} X_i]$.
[6] Such as "the (two stage) stochastic version" of Gupta et al [14].
[7] Which is almost tight
[8] When unknown, it is natural to assume the distributions are i.i.d. Otherwise, the algorithm can be easily tricked.

1.1.1 Allowing Small Congestion

We show that a greedy algorithm does surprisingly well, if we relax the connectivity requirements by a constant factor. Let α denote an arbitrary scale factor. We define an *α-scaled* variant of the greedy algorithm in which the goal is to find only $\lfloor \frac{r_i}{\alpha} \rfloor$ disjoint paths between the endpoints of σ_i. Our main result states that the scaled greedy algorithm is polylogarithmic competitive.

▶ **Theorem 1.** *For any constant $\epsilon > 0$, the $(2+\epsilon)$-scaled greedy algorithm is $(O(\log^2 n \log_{1+\epsilon} k), 2+\epsilon)$-competitive. For the single-source variant, the competitive ratio is $(O(\log n \log_{1+\epsilon} k), 2+\epsilon)$.*
Furthermore, for uniform SNDP, 2-scaled greedy is $(O(\log^2 n), 2)$-competitive.

We start by demonstrating a deep connection between the greedy method for SNDP and the *Steiner packing problems.* The Steiner packing problems are motivated by vast applications in VLSI-layout and have been used as an algorithmic toolkit in computer science. In the *Steiner tree packing* problem, we are given a graph $G = (V, E)$ and a set S of vertices and the goal is to find the *Steiner decomposition number* (SDN), the maximum number of edge-disjoint subgraphs that each connects the vertices of S. We note that a minimal connecting subgraph is a *Steiner tree* with respect to S. In the *Steiner forest packing* problem, we are given a set of demand pairs $u_i, v_i \in V$ and the goal is to find SDN, the maximum number of edge-disjoint subgraphs that in each the demand pairs are connected.

For simplicity, let us assume we have a *uniform* instance. In a uniform instance, we are given an integer k in advance and for any demand (u, v) we must k-connect u to v. Let opt denote the optimal SNDP solution, with the Steiner decomposition number q. In Section 2, we show that the $(\frac{k}{q})$-scaled greedy algorithm approximates opt up to logarithmic factors. Intuitively, every forest in the Steiner forest decomposition, gives us a path to satisfy a demand. Hence, we need to bound the overall cost of satisfying demands in all the q forests. The crux of our analysis is then to charge the cost of the scaled greedy to that of a parallel set of greedy algorithms that solve 1-connectivity instances on every forest. Finally, to get a polylogarithmic competitive algorithm, we need to find a universal lower bound on the SDN number q with respect to k.

It is shown that finding SDN is NP-hard and cannot be computed in polynomial time unless P=NP[6] (Algorithmica'06). Given that there exist q disjoint Steiner forests connecting a set of demands, it is straight forward to show the graph is q-connected on the demands. Therefore, a natural upper bound on SDN is the minimum connectivity of the endpoints of demands. For the case of spanning trees (Steiner tree with $S = V(G)$), it is proven that the above upper bound also provides a good approximation guarantee for the problem. In other words, any k-connected graph can be decomposed into $k/2$ edge-disjoint spanning trees[23]. This is also followed by a matching upper bound. The problem is much subtler when S does not encompass all vertices of the graph. The first lower bound for the Steiner tree packing problem was achieved by Petingi and Rodriguez[24] (CON'03) who proved every S-k-connected[9] has $\lfloor (2/3)^{(|V(G)-|S|)} k/2 \rfloor$ disjoint Steiner trees. This was later improved by Kriesell [20] (JCT'03), Jain, Mahdian, and Salavatipour [19] (SODA'03), Lau [21] (FOCS'04), and DeVos, McDonald, and Pivotto [8] (Man'13), the most recent of which shows for every S-$(5k+4)$-connected graph, we can find k edge-disjoint Steiner trees. However, the main conjecture is that, similar to the case of spanning trees, every S-k-connected graph admits a $k/2$-disjoint Steiner tree decomposition [20].

[9] A graph which is k-connected on a set of vertices S

For a set of demand pairs (u_i, v_i)'s, let $\mathcal{T}$ denote the set of Steiner forests that satisfy all the demands. In the *fractional Steiner forest packing* problem, the output is a fractional assignment x over $\mathcal{T}$ such that for every edge e, $\sum_{T\in\mathcal{T}:e\in T} x_T$ is not more than one. The goal is to find a fractional Steiner forest decomposition with the maximum total sum of weights in x. While the term fractional Steiner forest packing is not explicitly used in the previous work, it follows from the arguments of [4, 8] that the conjecture of Kriesell holds for the fractional variant.

▶ **Theorem 2** (proven in [4]). *Given a set of demand pairs (u_i, v_i), if G is k-connected for every demand pair, then the fractional Steiner decomposition number is at least $k/2$.*

Indeed, in Section 2, we use a dependent rounding method to show that the connection between SDN and the competitiveness of the greedy approach holds even for the stronger fractional variant of SDN. Hence, Theorem 2 implies that the 2-scaled greedy algorithm, achieves a polylogarithmic competitive ratio for the uniform SNDP. Finally, we prove that the scaled greedy is also competitive for the non-uniform variant if one is willing to lose an extra $O(\log k)$ factor in the competitive ratio (Theorem 1).

1.1.2 Online Stochastic SNDP

A single-source *uniform* instance of online SNDP is an instance in which for every demand σ_i, $u_i = u$, $r_i = k$ for some vertex $u \in V$ and integer k. For a non-uniform variant, let $k = \max_i r_i$. Let D be a given probability distribution over V. In i.i.d. SNDP, at each online step i, a random connectivity demand $\sigma_i = (u, v_i, k)$ arrives, where v_i is drawn independently at random from distribution D. We call the problem unknown distribution SNDP if the probability distribution D is not given in advance. Another interesting generalization of the i.i.d. model, which we call the prophet SNDP is defined as follows. In prophet SNDP, instead of only a single probability distribution D, we are given T probability distributions $D_1, \ldots, D_T$, such that the i-th demand is $\sigma_i = (u, v_i, k)$, where v_i is drawn independently at random from distribution D_i. In all three variants of the stochastic SNDP, the competitive-ratio is defined as the expected cost of an algorithm $\mathcal{A}$ over the expected cost of an optimal offline algorithm while the distributions are chosen by an adversary. More precisely let $E[A(\omega)]$ and $E[opt(\omega)]$ denote the expected cost of an algorithm $\mathcal{A}$ and the expected cost of an optimal offline algorithm for an online scenario ω, respectively. Thus the competitive-ratio of algorithm $\mathcal{A}$ is defined as follows.

$$cr(\mathcal{A}) := \max_D \frac{E_{\omega\sim D}[\mathcal{A}(\omega)]}{E_{\omega\sim D}[opt(\omega)]}.$$

We first provide an oblivious[10] greedy algorithm for the i.i.d. SNDP. This algorithm starts with a sampling of the demands and finding a 2-approximation solution for them using the algorithm of Jain [18]. Let us call this the backbone solution. Then, to satisfy each demand v, we simply connect it to the backbone using the cheapest set of edges and show that this greedy approach leads to a 4-competitive solution. The oblivious algorithm for the i.i.d. case is in fact a stepping stone that enables us to further analyze the greedy algorithm in the unknown distribution and different known distributions (prophet) cases.

▶ **Theorem 3.** *The oblivious greedy is 4-competitive algorithm for i.i.d. SNDP.*

[10] An oblivious algorithm connects a demand through a path which is independent of the rest of the demands

A similar result to Theorem 3 is also given in [5].

1.1.2.1 Unknown i.i.d.

Although computing the backbone solution is impossible for the unknown i.i.d. case, we take advantage of our analysis of the oblivious i.i.d. algorithm and show that the classic greedy algorithm is $O(\log n)$-approximation for unknown i.i.d. SNDP. The main idea is simple but tricky: due to the sampling nature of the backbone solution, for every $1 \leq k \leq T/2$ we can think of a solution for the first k demands as a backbone solution for the next k demands. Hence, we can exploit our analysis for the known i.i.d. case. This in conjunction with the submodularity of Steiner networks results in the desired competitive ratio. We note that the factor $O(\log n)$ is almost tight given the $\Omega(\frac{\log n}{\log \log n})$ lower bound of [9] for the 1-connectivity case.

▶ **Theorem 4.** *The classic greedy algorithm is $O(\log n)$-competitive for unknown i.i.d. SNDP.*

1.1.2.2 From Oblivious i.i.d. to Prophet

We show if there exists a competitive oblivious algorithm for an online problem in i.i.d. setting, we can obtain a competitive algorithm for the same problem in prophet setting. Roughly speaking, we show that we can combine different distributions in the prophet setting to obtain a single average distribution. Therefore, the i.i.d. oblivious algorithm for the average distribution does not incur more than a constant factor to the competitive ratio.

▶ **Theorem 5** (restated informally)**.** *Given an oblivious α-competitive online algorithm for problem $\mathcal{P}$ in the i.i.d. setting, there exists an $\alpha\frac{2e}{e-1}(1+o(1))$-competitive online algorithm for $\mathcal{P}$ in prophet setting.*

▶ **Corollary 6.** *There exists a constant competitive algorithm for prophet SNDP.*

Using our framework, we can obtain competitive algorithms for many fundamental and classical problems in prophet setting. For instance, define $D_1, \ldots, D_T$ be T probability distributions over the elements of a set cover instance. Now let i-th demand of a set cover problem be an element randomly and independently drawn from distribution D_i. We call this problem the prophet set cover problem. Similarly one may define prophet facility location the same as the classical facility location problem, with the difference that the i-th demand is randomly drawn from a known distribution D_i. Garg et al. [9] provide oblivious online algorithms for i.i.d. facility location and i.i.d. vertex cover. Applying the above framework directly results in the following corollary.

▶ **Corollary 7.** *There exist constant competitive algorithms for prophet vertex cover and prophet facility location problems.*

Also, Grandoni et al. [12] provide an oblivious online algorithm for i.i.d. set cover. Hence, we have the following corollary.

▶ **Corollary 8.** *There exists an $O(\log n)$-competitive algorithm for prophet set cover.*

1.2 Further Related Work

Over the past decades, SNDP and proper cut functions have been an important testbed for primal-dual and iterative rounding methods. Goemans and Williamson [11] (SIAM'95) were first to consider the case of $\{0, 1\}$-proper functions. They used a primal-dual method to obtain a 2-approximation algorithm for the problem; which later on got generalized to the celebrated moat-growing framework for solving connectivity problems. Klein and Ravi [26] (IPCO'93) considered the two-connectivity problem and the case of $\{0, 2\}$-proper functions. They gave a primal-dual 3-approximation algorithm for the problem. Williamson, Goemans, Mihail, and Vazirani [27] (Combinatorica'95) were first to consider general proper functions. They too developed a primal-dual algorithm with approximation ratio $2k$, where $k = \max_S f(S)$. Subsequently, Goemans, Goldberg, Plotkin, Shmoys, Tardos, and Williamson [10] (SODA'94) presented a primal-dual $2H(k)$-approximation algorithm, where $H(k)$ is the k^{th} harmonic number. Finally, in his seminal work [18] (Combinatorica'01), Jain introduced the *iterative rounding* method by developing a 2-approximation algorithm for network design problems characterized by proper cut functions[11]. We refer the reader to [13] for a survey of results for (offline) network design problems.

2 Uniform SNDP

In this section we consider the uniform-connectivity version of the online survivable Steiner network design problem, in which all connectivity requirements are equal to a given number. For this problem we first give a very simple algorithm and then analyze it using Steiner packing tools. Further in the paper, we explain how to generalize our algorithm to make it work for inputs with non-uniform connectivity requirements.

In the online uniform-connectivity survivable Steiner forest problem we are given an offline graph $G = \langle V(G), E(G)\rangle$, an integer k, and an online stream of demands $S = (s_1, t_1), (s_2, t_2), \ldots$. Every time a demand (s_i, t_i) arrives we have to add some of the edges of G to our current solution H in order to make k edge-disjoint paths between s_i and t_i in H. The online uniform-connectivity survivable Steiner tree problem is a special case of the forest problem in which the second endpoints of all demands are fixed at some vertex *root*. The objective of the problems is to minimize the cost of the selected subgraph H according to a given cost function.

A simple approach to solve these problems is to choose edges based on the following greedy method: for every demand add a minimum-cost subset of edges that satisfies the k-connectivity between its endpoints. In this section we show that this algorithm is not competitive to the optimum offline solution. This is shown by Lemma 14 in which we give an instance graph and a series of demands for which the greedy algorithm gives a solution of cost $\Omega(n)$ times the cost of the optimum offline solution.

However, we show a modified version of the greedy algorithm can be a viable approach for these problems if we lose some factor on the connectivity requirement. This can be done by satisfying half of the required connectivity. In particular, for every demand we add a minimum-cost subset of the edges that makes the current solution $(k/2)$-connected between the endpoints of that demand. Let us call this algorithm *GA*. In this section we show the cost of the edges GA selects is poly-logarithmically competitive to the optimum offline solution that satisfies k-connectivity for every demand.

[11] Indeed, the results in [10] and [18] applies to the more general class of weakly or skew supermodular cut functions.

Algorithm 1: 2-scaled Greedy

Input: A graph G, an integer k, and an online stream of demands $(s_1, t_1), (s_2, t_2), \ldots$.

Output: A set H of edges such that every given demand (s_i, t_i) is connected through k edge-disjoint paths in H.

Offline Process:

1: Initialize $H = \varnothing$.

Online Scheme; assuming a demand (s_i, t_i) is arrived:

1: $P_i =$ A minimum-cost subset of edges, such that s_i is $k/2$-connected to t_i in $H \cup P_i$.

2: Update $H = H \cup P_i$.

▶ **Theorem 9.** *For the online survivable Steiner forest problem, the output of GA satisfies $(k/2)$-connectivity for every demand and its cost is $O(\log^2 n)$-competitive.*

▶ **Theorem 10.** *For the online survivable Steiner tree problem, the output of GA satisfies $(k/2)$-connectivity for every demand and its cost is $O(\log n)$-competitive.*

As a direct consequence of adding edges according to GA, the $(k/2)$-connectivity is guaranteed for every demand. To complete the proof of the theorems, we need to show that the cost of the solution produced by GA is upper bounded by a factor of $O(\log^2 n)$ for forests, and $O(\log n)$ for trees.

Let $c : E(G) \rightarrow \mathbb{R}^{\geq 0}$ be the cost function on the edges. With some abuse of notation, we also use $c(Y)$ for a subset of edges $Y \subseteq E(G)$ as the sum of the cost of the edges in Y. With this notation we can say at every step i GA chooses a subset of edges P_i that satisfies $(k/2)$-connectivity and minimizes $c(P_i)$.

The overall idea of the proofs is as follows. We take an optimum solution and charge every $c(P_i)$ to $c(L_i)$, where L_i is a set of edges chosen from the optimum solution. The way we define L_i's allows them to have overlapping edges, but we show that their total cost is limited by the desired poly-logarithmic factor of the cost of the optimum solution. More specifically, we charge $c(L_i)$ to the cost of a fractional routing Q_i between s_i and t_i. Every Q_i is itself a linear combination of routes on different Steiner forests of the optimum solution. The coefficients of this linear combination are achieved from an Steiner forest packing of the optimum solution. In this fashion, the problem boils down to finding an upper bound for the total cost of routings on each Steiner forest. In the following we formally prove every step in detail.

Let OPT be an optimum offline solution of the survivable Steiner forest problem on graph G, a stream of demands S, and the connectivity requirement k. Now we define L_i for every demand i as a minimum-cost set of edges in OPT that is $(k/2)$-connected between s_i and t_i assuming the endpoints of every previous demand are contracted. In particular, we call a set of edges a *pseudo-path* between s_i and t_i if there is a path between these vertices using those edges and the edges in $\{(s_j, t_j) | \forall j < i\}$. A *pseudo-routing* between s_i and t_i is hence a set of pseudo-paths between s_i and t_i. With these definitions, L_i is a minimum-cost pseudo-routing between s_i and t_i in OPT that consists of $k/2$ pseudo-paths. The following lemma shows the relation between the costs of L_i and P_i.

▶ **Lemma 11.** *For every demand i, $c(P_i) \leq c(L_i)$.*

Proof. Every time a demand i arrives, GA finds a set P_i with the minimum cost and adds it to H in order to satisfy $(k/2)$-connectivity between s_i and t_i. Note that the endpoints of every demand $j < i$ are already connected with $k/2$ disjoint paths in H. Besides, L_i is a pseudo-routing between s_i and t_i which is $(k/2)$-connected between s_i and t_i if we contract the two endpoints of every previous demand. Therefore adding L_i to H makes H $(k/2)$-connected between s_i and t_i. Since GA finds a minimum-cost set of edges that satisfies $(k/2)$-connectivity in H, $c(P_i)$ never exceeds $c(L_i)$. ◀

In the remaining we show how to charge the total cost of L_i's to $c(OPT)$. As a property of an optimum solution, OPT contains k edge-disjoint paths between the endpoints of every demand $(s_i, t_i) \in S$. Therefore, according to Theorem 2 there exists a solution for the fractional Steiner forest packing of OPT and demand set S with value at least $k/2$. Let $\mathbf{z}$ be a Steiner forest packing of OPT with value $k/2$. In the following we use $\mathcal{F}_S(OPT)$ to denote the collection of all Steiner forests of OPT with respect to demand set S. The theorem states there exists a vector $\mathbf{z}$ such that

$$\sum_{F \in \mathcal{F}_S(OPT)} z_F = k/2 \tag{1}$$

$$\sum_{F \in \mathcal{F}_S(OPT): e \in F} z_F \leq 1 \qquad \forall e \in OPT\ . \tag{2}$$

Moreover, the following inequality holds for the summation of the costs of these forests.

▶ **Lemma 12.** $\sum_{F \in \mathcal{F}_S(OPT)} z_F.c(F) \leq c(OPT)$.

Proof. For each forest we replace its cost with the sum of the cost of its edges.

$$\begin{aligned}
\sum_{F \in \mathcal{F}_S(OPT)} z_F.c(F) &= \sum_{F \in \mathcal{F}_S(OPT)} z_F \sum_{e \in F} c(e) \\
&= \sum_{e \in OPT} \sum_{F \in \mathcal{F}_S(OPT): e \in F} z_F c(e) \\
&= \sum_{e \in OPT} c(e) \left(\sum_{F \in \mathcal{F}_S(OPT): e \in F} z_F \right) .
\end{aligned}$$

Now we use the fact that the load on every edge in the fractional Steiner forest packing is no more than 1.

$$\begin{aligned}
\sum_{F \in \mathcal{F}_S(OPT)} z_F.c(F) &\leq \sum_{e \in OPT} c(e) && \textit{Inequality} \quad (2) \\
&= c(OPT) \ .
\end{aligned}$$

◀

Now for every forest $F \in \mathcal{F}_S(OPT)$ and every demand i we define $Q_i(F)$ as a minimum-cost pseudo-path between s_i to t_i in F. This definition allows using an edge $e \in F$ multiple times in $Q_i(F)$ of different demands. Note that $Q_i(F)$ can be considered as a fractional pseudo-routing between s_i and t_i with value z_F. Considering this for all forests in $\mathcal{F}_S(OPT)$, we achieve a fractional pseudo-routing between s_i and t_i that has a value of $k/2$. We use Q_i to refer to this fractional pseudo-routing and $c(Q_i) = \sum_{F \in \mathcal{F}_S(OPT)} z_F.c(Q_i(F))$ to refer to its cost.

For every demand i we have mentioned two different pseudo-routings between s_i and t_i in OPT with value $k/2$: an integral pseudo-routing L_i, and a fractional pseudo-routing Q_i. The following lemma shows the relation between the costs of these two.

▶ **Lemma 13.** *For every L_i and Q_i pseudo-paths defined as above, we have:*

$$c(L_i) \leq c(Q_i)$$

In the interest of space, we defer the proof of Lemma 13 to the full-version of the paper.

Finally for a particular $F \in \mathcal{F}_S(OPT)$ we show an upper bound for the sum of $c(Q_i(F))$ over all demands. First let us take a closer look at every $Q_i(F)$ on a particular F. Every time a new demand (s_i, t_i) arrives $Q_i(F)$ connects its endpoints through a pseudo-path in F. This can be generalized to an algorithm for the online single-connectivity Steiner forest problem that greedily connects the endpoints of every demand by fully buying a minimum-cost pseudo-path between s_i and t_i. This is very similar to the greedy algorithm proposed in [2]. Theorem 2.1 of that paper states that their greedy algorithm is $O(\log^2 n)$-competitive. The statement of that theorem is slightly different than Claim 2.1, but the same proof verifies the correctness of the claim.

▶ **Claim 2.1.** *For the online Steiner forest problem, the algorithm that connects every demand with a minimum-cost pseudo-path is $O(\log^2 n)$-competitive.*

Now we are ready to wrap up the proof of Theorem 9.

Proof of Theorem 9. Let ALG denote the output of GA. The cost of ALG is the sum of the cost of P_i's over all demands. Therefore, by applying lemmas 11 and 13 we have

$$\begin{aligned} c(ALG) &= \sum_{(s_i,t_i)\in S} c(P_i) \\ &\leq \sum_{(s_i,t_i)\in S} c(L_i) && \textit{Lemma} \quad 11 \\ &\leq \sum_{(s_i,t_i)\in S} c(Q_i) && \textit{Lemma} \quad 13 \end{aligned}$$

Now we replace $c(Q_i)$ with the weighted sum of $c(Q_i(F))$'s with respect to $\mathbf{z}$.

$$\begin{aligned} c(ALG) &\leq \sum_{(s_i,t_i)\in S} \sum_{F\in\mathcal{F}_S(OPT)} z_F.c(Q_i(F)) \\ &= \sum_{F\in\mathcal{F}_S(OPT)} z_F \sum_{(s_i,t_i)\in S} c(Q_i(F)) \end{aligned} \tag{3}$$

By applying Claim 2.1 to Inequality (3) we achieve an $O(\log^2 n)$-competitive ratio for GA.

$$\begin{aligned} c(ALG) &\leq \sum_{F\in\mathcal{F}_S(OPT)} z_F \Big(O(\log^2 n) c(F) \Big) && \textit{Claim} \quad 2.1 \\ &\leq O(\log^2 n) \sum_{F\in\mathcal{F}_S(OPT)} z_F.c(F) \\ &\leq O(\log^2 n) c(OPT) \ . && \textit{Lemma} \quad 12 \end{aligned}$$

◀

Finally, for the survivable Steiner tree problem we show that GA is $O(\log n)$-competitive. In other words, if one endpoint of every demand is fixed at the root, then the output of GA is at most $O(\log n)$ times the optimum offline solution. To complete the proof of Theorem 10 we use a result from [22]. In that paper the authors prove a competitive ratio of $O(\log n)$ for the algorithm which satisfies every demand using a minimum-cost pseudo-path. The following claim is a restatement of their result.

▶ **Claim 2.2.** *For the online Steiner tree problem, the algorithm that satisfies each demand with a minimum-cost pseudo-path is $O(\log n)$-competitive.*

of Theorem 10. Note that the tree problem is a special case of the forest problem, hence Inequality (3) also holds for it. By applying Claim 2.2 to that inequality the proof is complete.

$$\begin{aligned} c(ALG) &\leq \sum_{F \in \mathcal{F}_S(OPT)} z_F \left(O(\log n) c(F) \right) && \textit{Claim} \quad 2.2 \\ &\leq O(\log n) \sum_{F \in \mathcal{F}_S(OPT)} z_F . c(F) \\ &\leq O(\log n) c(OPT) \ . && \textit{Lemma} \quad 12 \end{aligned}$$

◀

The following Lemma shows that there exists a graph G and a sequence of demands σ such that Greedy algorithm performs $\Omega(n)$ times worse than the optimal solution.

▶ **Lemma 14.** *The competitive ratio of the greedy algorithm for survivable Steiner network design is $\Omega(n)$, even if every connectivity requirement is exactly 2.*

Proof. First we provide an online instance of the survivable network design problem where every connectivity requirement is exactly 2 and show the greedy algorithm performs poorly in comparison with the optimal solution. We construct a graph G of size n as follows. For each $1 \leq i \leq n-1$, there exist two undirected edges from node i to node $i+1$ of weights 1 and $n - i - \epsilon$ for some small $\epsilon > 0$. There exist two undirected edges from node n to node 1 with weights 1 and $n - \epsilon$. Thus G is the union of two cycles of size n. We construct a set of demands S as follows. For each $1 \leq i \leq n-1$, let $(i, i+1)$ be the i'th demand in S.

Now we analyze the output of the greedy algorithm for the input instance. We claim that after satisfying demand i the greedy algorithm has selected both edges between j and $j+1$ for every $j \leq i$. We prove this claim by induction. For the base case, when the first demand arrives the greedy algorithm chooses both edges between nodes 1 and 2 which costs $n - \epsilon$. Now assume the greedy algorithm has selected every edge between j and $j+1$ for every $j < i$ before the arrival of the i'th demand. When the i'th demand arrives, the set of edges with minimum cost that provides two edge-disjoint paths from i to $i+1$ is the two edges between i and $i+1$ which costs $n - i - \epsilon$. Thus the total cost of the greedy algorithm at the end is $\frac{n(n-1)}{2} - \epsilon n$. However, the optimum offline solution chooses the cycle containing all edges of weight 1. Thus the competitive ratio of the greedy algorithm is $\Omega(n)$. ◀

References

1. Noga Alon, Baruch Awerbuch, Yossi Azar, Niv Buchbinder, and Joseph Naor. The online set cover problem. *SIAM Journal on Computing*, 39(2):361–370, 2009.
2. Baruch Awerbuch, Yossi Azar, and Yair Bartal. On-line generalized steiner problem. *Theoretical Computer Science*, 324(2):313–324, 2004.
3. Piotr Berman and Chris Coulston. On-line algorithms for steiner tree problems. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 344–353. ACM, 1997.
4. Robert Carr and Santosh Vempala. Randomized metarounding. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 58–62. ACM, 2000.
5. Chandra Chekuri and Nitish Korula. A graph reduction step preserving element-connectivity and applications. In *International Colloquium on Automata, Languages, and Programming*, pages 254–265. Springer, 2009.

6 Joseph Cheriyan and Mohammad R Salavatipour. Hardness and approximation results for packing steiner trees. *Algorithmica*, 45(1):21–43, 2006.

7 Julia Chuzhoy and Shi Li. A polylogarithmic approximation algorithm for edge-disjoint paths with congestion 2. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 233–242. IEEE Computer Society, 2012. `doi:10.1109/FOCS.2012.54`.

8 Matt DeVos, Jessica McDonald, and Irene Pivotto. Packing steiner trees. *arXiv preprint arXiv:1307.7621*, 2013.

9 Naveen Garg, Anupam Gupta, Stefano Leonardi, and Piotr Sankowski. Stochastic analyses for online combinatorial optimization problems. In *SODA*, pages 942–951. Society for Industrial and Applied Mathematics, 2008.

10 Michel X Goemans, Andrew V Goldberg, Serge A Plotkin, David B Shmoys, Eva Tardos, and David P Williamson. Improved approximation algorithms for network design problems. In *SODA*, volume 94, pages 223–232, 1994.

11 Michel X Goemans and David P Williamson. A general approximation technique for constrained forest problems. *SIAM Journal on Computing*, 24(2):296–317, 1995.

12 Fabrizio Grandoni, Anupam Gupta, Stefano Leonardi, Pauli Miettinen, Piotr Sankowski, and Mohit Singh. Set covering with our eyes closed. In *Foundations of Computer Science, 2008. FOCS'08. IEEE 49th Annual IEEE Symposium on*, pages 347–356. IEEE, 2008.

13 Anupam Gupta and Jochen Könemann. Approximation algorithms for network design: A survey. *Surveys in Operations Research and Management Science*, 16(1):3–20, 2011.

14 Anupam Gupta, Ravishankar Krishnaswamy, and R Ravi. Online and stochastic survivable network design. *SIAM Journal on Computing*, 41(6):1649–1672, 2012.

15 Mohammad T Hajiaghayi, Vahid Liaghat, and Debmalya Panigrahi. Online node-weighted steiner forest and extensions via disk paintings. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 558–567. IEEE, 2013.

16 MohammadTaghi Hajiaghayi, Vahid Liaghat, and Debmalya Panigrahi. Near-optimal online algorithms for prize-collecting steiner problems. In *Automata, Languages, and Programming*, pages 576–587. Springer, 2014.

17 Makoto Imase and Bernard M Waxman. Dynamic steiner tree problem. *SIAM Journal on Discrete Mathematics*, 4(3):369–384, 1991.

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

19 Kamal Jain, Mohammad Mahdian, and Mohammad R Salavatipour. Packing steiner trees. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 266–274. Society for Industrial and Applied Mathematics, 2003.

20 Matthias Kriesell. Edge-disjoint trees containing some given vertices in a graph. *Journal of Combinatorial Theory, Series B*, 88(1):53–65, 2003.

21 Lap Chi Lau. An approximate max-steiner-tree-packing min-steiner-cut theorem. In *Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on*, pages 61–70. IEEE, 2004.

22 Joseph Naor, Debmalya Panigrahi, and Monika Singh. Online node-weighted steiner tree and related problems. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 210–219. IEEE, 2011.

23 CSJA Nash-Williams. Edge-disjoint spanning trees of finite graphs. *Journal of the London Mathematical Society*, 1(1):445–450, 1961.

24 Louis Petingi and J Rodriguez. Bounds on the maximum number of edge-disjoint steiner trees of a graph. *Congressus Numerantium*, pages 43–52, 2000.

25 Jiawei Qian and David P Williamson. An o (logn)-competitive algorithm for online constrained forest problems. In *Automata, Languages and Programming*, pages 37–48. Springer, 2011.

26 R Ravi and Philip Klein. When cycles collapse: A general approximation technique for constrained two-connectivity problems. In *3rd Conference on Integer Programming and Combinatorial Optimization*, pages 39–56, 1993.

27 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.