Online Network Design with Outliers

Aris Anagnostopoulos[1], Fabrizio Grandoni[2], Stefano Leonardi[3], and Piotr Sankowski[4]

[1] Sapienza University of Rome, Rome, Italy. aris@dis.uniroma1.it
[2] University of Rome Tor Vergata, Rome, Italy. grandoni@disp.uniroma2.it
[3] Sapienza University of Rome, Rome, Italy. leon@dis.uniroma1.it
[4] Sapienza University of Rome, Rome, Italy and University of Warsaw, Warsaw, Poland. sankowski@dis.uniroma1.it

Abstract. In a classical online network design problem, traffic requirements are gradually revealed to an algorithm. Each time a new request arrives, the algorithm has to satisfy it by augmenting the network under construction in a proper way (with no possibility of recovery). In this paper we study a natural generalization of the problems above, where a fraction of the requests (the *outliers*) can be disregarded. Now, each time a request arrives, the algorithm first decides whether to satisfy it or not, and only in the first case it acts accordingly; in the end at least k out of t requests must be selected. We cast three classical network design problems into this framework, the *Online Steiner Tree with Outliers*, the *Online TSP with Outliers*, and the *Online Facility Location with Outliers*.
We focus on the known distribution model, where terminals are independently sampled from a given distribution. For all the above problems, we present bicriteria online algorithms that, for any constant $\epsilon > 0$, select at least $(1-\epsilon)k$ terminals with high probability and pay in expectation $O(\log^2 n)$ times more than the expected cost of the optimal offline solution (selecting k terminals). These upper bounds are complemented by inapproximability results.

1 Introduction

In a classical *online network design* problem, traffic requirements are revealed gradually to an algorithm. Each time a new request arrives, the algorithm has to satisfy it by augmenting the network under construction in a proper way. An online algorithm is α-*competitive* (or α-*approximate*) if the ratio between the solution computed by the algorithm and the optimal (offline) solution is at most α.

For example, in the *Online Steiner Tree* problem (OST), we are given an n-node graph $G = (V, E)$, with edge weights $c : E \to \mathbb{R}^+$, and a root node r. Then t terminal nodes (where t is known to the algorithm) arrive one at a time. Each time a new terminal arrives, we need to connect it to the Steiner tree $\mathcal{S}$ under construction (initially containing the root only), by adding a proper set of edges to the tree. The goal is minimizing the final cost of the tree. The input for the *Online TSP* problem (OTSP) is the same as in OST. The difference is that here the solution is a permutation ϕ of the input terminals. (Initially, $\phi = (r)$). Each time a new terminal arrives, we can insert it into ϕ at an arbitrary point. The goal is to minimize the length of shortest cycle visiting the nodes in ϕ according to their order of appearance in ϕ. In the *Online Facility Location* problem (OFL), we are also given a set of facility nodes $\mathcal{F}$, with associated opening costs $\mathrm{o} : V \to \mathbb{R}^+$. Now, each time a new terminal v arrives, it must be connected to some facility f_v: f_v is opened if not already the case. The goal is to minimize the

facility location cost given as $\sum_{e \in F} f(e) + \sum_{v \in K} dist_G(v, e_v)$, where $F = \cup_{v \in K} f_v$ is the set of open facilities[5].

When the input sequence is chosen by an adversary, $O(\log n)$-approximation algorithms are known for the problems above, and this approximation is tight [16, 26, 28]. Recently, the authors of [13] studied the case where the sequence of terminals is sampled from a given distribution. For these relevant special cases, they provided online algorithms with $O(1)$ expected competitive ratio[6]. This shows a logarithmic approximability gap between worst-case and stochastic variants of online problems.

Stochastic Online Network Design with Outliers. In this paper we study a natural generalization of online network design problems, where a fraction of the requests (the *outliers*) can be disregarded. Now, each time a request arrives, the algorithm first decides whether to satisfy it or not, and only in the first case updates the network under construction accordingly. Problems with outliers have a natural motivation in the applications. For example, mobile phone companies often declare the percentage of the population which is covered by their network of antennas. In order to declare a large percentage (and attract new clients), they sometimes place antennas also in areas where costs exceed profits. However, covering everybody would be too expensive. One option is choosing some percentage of the population (say, 90%), and covering it in the cheapest possible way. This type of problems is well-studied in the offline setting, but it was never addressed before in the online case (to the best of our knowledge).

We restrict our attention to the outlier version of the three classical online network design problems mentioned before: *Online Steiner Tree with Outliers* (outOST), *Online TSP with Outliers* (outOTSP), and *Online Facility Location with Outliers* (outOFL). For each such problem, we assume that only $0 < k < t$ terminals need to be connected in the final solution.

It is easy to show that, for $k \leq t/2$, the problems above are not approximable in the adversarial model. The idea is providing k terminals with connection cost $M \gg k$. If the online algorithm selects at least an element among them, the next elements have connection cost 0. Otherwise, the next elements have connection cost M^2 and the online algorithm is forced to pay a cost of kM^2. Essentially the same example works also if we allow the online algorithm to select only $(1-\epsilon)k \geq 1$ elements. For this reason and following [13], from now on we focus our attention on the *stochastic* setting, where terminals are sampled from a given probability distribution[7]. As we will see, these stochastic online problems have strong relations with classical secretary problems.

There are two models for the stochastic setting: the *known-distribution* and the *unknown-distribution* models. In the former the algorithm knows the distribution from

[5] For a weighted graph G, $dist_G(u, v)$ denotes the distance between nodes u and v in the graph. For the sake of simplicity, we next associate an infinite opening cost to nodes which are not facilities, and let $\mathcal{F} = V$.

[6] Throughout this paper the expected competitive ratio, also called ratio of expectations (RoE), is the ratio between the expected cost of the solution computed by the online algorithm considered and the expected cost of the optimal offline solution. Sometimes in the literature the expectation of ratios EoR is considered instead (which is typically more involving).

[7] For the sake of shortness, we will drop the term stochastic from problem names.

which terminals are sampled. In the latter the algorithm does not have any information about the distribution apart from the incoming online requests.

Our Results and Techniques. First, we give inaproximability results and lower bounds. For the known-distribution model we show that the considered problems are inapproximable if we insist on selecting exactly k elements, for $k = 1$ and for $k = t - 1$. To prove these results we need to carefully select input distributions that force the online algorithm to make mistakes: if it decides to select a terminal then with sufficiently high probability there will be cheap subsequent requests, inducing a large competitive ratio, while if it has not selected enough terminals it will be forced to select the final terminals, which with significant probability will be costly.

Furthermore, we prove an $\Omega(\log n/\log\log n)$ lower bound on the expected competitive ratio even when the online algorithm is allowed to select αk terminals only, for a constant $\alpha \in (0, 1)$. To prove it we use results from urn models.

Finally, for the unknown-distribution model we show a lower bound of $\Omega(\log n)$ for $k = \Theta(t)$ if the online algorithm is required to select $k - O(k^\alpha)$ requests for $0 \leq \alpha < 1$.

Given the inapproximability results for the case that the online algorithm has to select exactly k terminals, we study bicriteria algorithms, which select, for any given $\epsilon > 0$, at least $(1 - \epsilon)k$ terminals with high probability[8], and pay in expectation $O(\log^2 n)$ times more than the expected cost of the optimal offline solution (selecting at least k terminals).

To obtain these results, we are first able to show that very simple algorithms provide a $O(k)$ expected competitive ratio. Henceforth, the main body of the paper is focused on the case $k = \Omega(\log n)$. Our algorithms crucially exploit the probabilistic embeddings of graph metrics into tree metrics developed by Bartal et al. [4, 9]. A Bartal tree of the input graph is used to partition the nodes into a collection of groups of size $\Theta(\frac{n}{t}\log n)$. Note that $\Theta(\log n)$ terminals are sampled in each group with high probability. Next, in the case of the outOST problem, we compute an anticipatory solution formed by a Steiner tree on k out of t terminals sampled beforehand from the known distribution. The anticipatory solution is deployed by the algorithm. When the actual terminals arrive, the algorithm selects all terminals that belong to a group (which we *mark*) that contains at least one terminal selected in the anticipatory solution, and connects the selected terminals to the anticipatory solution itself. Roughly speaking, there are $\Theta(k/\log n)$ marked groups and each such group collects $\Theta(\log n)$ actual terminals: altogether, the number of connected terminals is $\Theta(k)$. A careful charging argument shows that the connection cost to the anticipatory solution is in expectation $O(\log n)$ times the cost of the embedding of the anticipatory solution in the Bartal tree. In expectation, this tree embedding costs at most $O(\log n)$ times more than the anticipatory solution itself, which in turn costs $O(1)$ times more than the optimal solution. Altogether, this gives a $O(\log^2 n)$ competitive ratio.

The results on outOST immediately generalize to the case of outOTSP, modulo constant factors: for this reason we will describe the results for outOST only. The basic idea

[8] Throughout this paper we use the term *with high probability* (abbreviated whp.) to refer to probability that approaches 1 as k, the number of selected terminals, grows. In particular, the probability of failure is polynomially small for $k = \Omega(\log n)$ in the cases considered.

is to construct a Steiner tree using an online algorithm for outOST, and to duplicate its edges. This defines a multi-graph containing an Euler tour spanning $\Theta(k)$ terminals. By shortcutting the Euler tour we obtain the desired permutation ϕ of selected terminals. In each step the Euler tour can be updated preserving the the relative order of the terminals in the permutation. The cost of the optimal Steiner tree is a lower bound on the cost of the optimal TSP tour. Edge duplication introduces only a factor 2 in the approximation. Summarizing the discussion above.

Lemma 1. *Given an online α-approximation algorithm for outOST, there is an online 2α-approximation algorithm for outOTSP.*

The situation for outOFL is more involved, as in addition to the connection cost we need to take care of the facilities' cost. In this case, as well, we deploy an anticipatory solution on k out of t terminals sampled beforehand from the known distribution. In order to be able to apply some charging arguments we create a new virtual metric space, which can also capture the cost of opening the facilities: we connect every vertex of the graph to a virtual root in the tree metric with an edge of cost equal to the corresponding facility opening cost. An additional complication is to decide when to open facilities that are not opened in the anticipatory solution. We open a new facility if a selected vertex is connected to the closest facility in the anticipatory solution through a path that traverses the root in the tree embedding.

To summarize our results:

- We give inapproximability results and lower bounds for the known-distribution model.
- We give $O(\log^2 n)$ approximation algorithms for the outOST (Section 3), the outOTSP, and the outOFL (Section 4) problems for the known distribution model. In the case that $k = \Theta(t)$ we give $O(\log n \log\log n)$ approximations (details will appear in the full version of the paper).
- We extend the upper and lower bounds to the unknown-distribution model (details will appear in the full version).

The problems that we consider in this paper include as a special case *minimization* versions of the *secretary* problem. In the classical *secretary* problem a set of t elements (the *secretaries*), each one with an associated non-negative numerical value, are presented one by one to the algorithm (the *employer*). The algorithm has to decide when to stop and select the current element with the goal of maximizing the value of the selected element. A well-known extension of the problem above is the *multiple-choice secretary* problem, where the algorithm has to select $k < t$ elements of the sequence with the goal of maximizing the sum of the k selected values (or, alternatively, the ranks of the selected elements). While this problem dates back to the fifties, it has recently attracted a growing interest given its connections to selecting winners in online auctions [2, 15].

In the classical secretary problem, it is easy to achieve a constant approximation to the optimal expected value; for example, waiting until seeing half the elements and then selecting the first element that has value higher than the maximum of the first half achieves in expectation a value that is at least $1/4$ of the optimal offline value. Here we show that the minimization version is strictly harder, the reason being that a wrong

choice might be very costly. The hardness arises from the fact that at least k secretaries must be hired: Intuitively, if $k - x$ secretaries have been hired after $t - x$ secretaries have been sampled, the last x secretaries must be hired irrespectively of their values. So, in Theorem 2 we show that even in the simple case that $k = 1$ the cost of the online algorithm can be exponentially larger than the optimal offline cost.

For the same reason (that a wrong choice can be very costly) the online network design problems with outliers are in general strictly harder than the versions without outliers. For example, in [13] the authors show that for the known distribution model the expected ratio of the online Steiner tree problem (without outliers, corresponding to the case that $k = t$) is constant. Instead, in Theorem 1 we show that even if we let $k = t - 1$ the approximation ratio can be arbitrarily large.

Throughout this paper we use *OPT* to denote the optimal offline solution, and *opt* to denote its expected cost. For a set of elements A and a cost function c defined on such elements, $c(A) := \sum_{a \in A} c(a)$. For a graph A, we use $c(A)$ as a shortcut for $c(E(A))$.

Related work. Competitive analysis of online algorithms has a long history (e.g., [5, 10, 29] and the many references therein). Steiner tree, TSP, and facility location can be approximated up to a worst-case $\Theta(\log n)$ competitive factor in the online case [16, 26, 28]. There have been many attempts to relax the notion of competitive analysis for classical list-update, paging and k-server problems (see [5, 10, 17, 18, 24, 27, 30]).

In many of the online problems studied in the literature, and in particular the versions of the online problems we study here without outliers ($k = t$), the case of known distribution was easy. As we mentioned, in this case outOST, outOTSP and outOFL reduce to the online stochastic version of Steiner tree, TSP, and facility location, for which the ratio between the the expected online cost and the expected optimal cost are constant for the known distribution model [13]. In the random permutation model, Meyerson [26] shows for facility location an algorithm with $O(1)$ ratio between the expected online cost and the expected optimal cost. In the Steiner tree problem the $\Omega(\log n)$ lower bound is still retained in the random permutation model [13].

The offline versions of the problems considered here are known as the *Steiner Tree problem with Outliers* (outST), the *TSP problem with Outliers* (outTSP) and the *Facility Location problem with Outliers* (outFL). For these problems, worst-case constant approximation algorithms are known [7, 12][9]. We will exploit such (offline) approximation algorithms as part of our online algorithms.

As we mentioned, the problems that we study in this paper have strong relations with the secretary problem. Secretary problems have been studied under several models. There is a rich body of research on secretary problems and the determination of optimal stopping rules in the random permutation model since the early sixties [8, 11, 14, 25]. In this classical model a set of t arbitrary numerical values is presented to the algorithm in random order. A strategy is known that selects the best secretary with probability $1/e$ [25]. For the multiple-choice secretary problem it has recently proposed [21] a strategy that achieves a $1 - O(\sqrt{1/k})$ fraction of the sum of the k largest elements in the sequence, i.e., a competitive ratio [5] that approaches 1 for $k \to \infty$.

[9] The k-MST problem studied in [12] and the Steiner tree problem with outliers are equivalent approximation-wise, modulo constant factors. The same holds for TSP with outliers.

In the known-distribution model, the numerical values are identical independent samples from a known distribution (e.g., [14, 20]). These problems are also known as house-selling problems (e.g., [19]), and generalizations have appeared under the name of dynamic and stochastic knapsack problems [1, 22]. In this model an online algorithm that maximizes the expected revenue is obtained through dynamic programming even for the multiple-choice version of the problem [14].

Secretary problems with an underlying graph structure have been recently studied in the context of online matching problems and their generalizations [3, 23]. One can define a minimization version of all the problems above, as we do here. Minimization secretary problems are much less studied in the literature, and most studies cover some basic cases. In particular, researchers have studied the problems where the goal is to minimize the expected rank of the selected secretary (as opposed to the actual expected cost) or to minimize the expected cost if the input distribution is uniform in $[0, 1]$ (look, for example, the work of Bruce and Ferguson [6] and the references therein). However, to our knowledge, there has not been any comparison of the online and offline solutions for arbitrary input distributions. In particular, nothing to our knowledge was known on the gap between their costs.

2 Lower Bounds

Let us start by proving inapproximability results for outOST in the known distribution model, when we insist on selecting *exactly* k terminals. Similar lower bounds hold for outOTSP and outOFL. The proof of the following theorem will appear in the full version of the paper.

Theorem 1. *In the known distribution model, the expected competitive ratio for outOST can be arbitrarily large for $k = t - 1$.*

The next theorem considers the somehow opposite case that k is very small (proof omitted). Note that the construction in the proof shows that the minimization version of the secretary problem has an exponential competitive ratio.

Theorem 2. *In the known distribution model, the expected competitive ratio for outOST can be exponentially large in the number n of nodes for $t = 3n/4$ and $k = 1$.*

Next we present an $O(\frac{\log n}{\log \log n})$ lower bound for outOST, outOTSP and outOFL, which applies also to the case that the online algorithm is allowed to connect only $\alpha\, k$ terminals, for a sufficiently large constant $\alpha \in (0, 1)$.

Theorem 3. *Assume that an online algorithm for outOST (resp., outOTSP or outOFL) is allowed to connect αk terminals, for a sufficiently large constant $\alpha \in (0, 1)$. Then the expected competitive ratio is $\Omega\left(\frac{\log n}{\log \log n}\right)$.*

Proof. We give the proof for outOST. The proof for the other two problems is analogous. Consider the star graph with the root r as center, and uniform edge weights 1. Suppose that each leaf is sampled with uniform probability $1/(n-1)$. Let $t = n - 1$ and $k = \frac{\ln n}{c \ln \ln n}$, for a sufficiently large constant c. When a leaf is sampled at least k times, the optimum solution cost is 1, and in any case it is not larger than $n - 1$. By

Figure 1 Algorithm `outost-large` for outOST.

(Preprocessing Phase)

Step 1. Compute a Bartal tree $\mathcal{B}$ for the input graph. Partition the leaves of $\mathcal{B}$ from left to right in groups $V_1, \ldots, V_{n/\sigma}$ of size σ.

Step 2. Sample t nodes $\tilde{T}$ from the input probability distribution. Compute a ρ_{outST}-approximate solution $\tilde{\mathcal{S}}$ to the (offline) outST problem induced by $\tilde{T}$. Let $\tilde{K}$ be the resulting set of k terminals, and $\mathcal{K}$ be the nodes of groups with at least one node in $\tilde{K}$, excluding the leftmost and rightmost such groups. Set $\mathcal{S} = \tilde{\mathcal{S}}$.

(Online Phase)

Step 3. For each input node $v \in T$, if $v \in \mathcal{K}$, add v to K and augment $\mathcal{S}$ with a shortest path to v.

standard balls-and-bins results, the probability that no leaf is sampled at least k times is polynomially small in n. Hence $opt = O(1)$.

Take now any online algorithm. Suppose that at some point this algorithm connects a terminal v for the first time. After this choice, the same terminal v will be sampled $O(1)$ times in expectation. Hence, the expected total number of connected terminals is proportional to the number of distinct leaves which are connected. This implies that the online algorithm is forced to connect $\Omega(k)$ distinct nodes in expectation, with a cost of $\Omega(k)$. Therefore, the competitive ratio is $\Omega(k) = \Omega\left(\frac{\log n}{\log \log n}\right)$. □

We observe that the proof above applies to the case of small values of k. Extending the proof to large values of k (or finding a better algorithm in that case) is an interesting open problem. The next theorem (proof omitted) moves along these lines.

Theorem 4. *In the unknown distribution model, the expected competitive ratio for outOST is $\Omega(\log n)$ if the online algorithm is required to connect $(1-\epsilon)k$ terminals for $\epsilon < \frac{\log n}{\sqrt{n}}$, $t = n$ and $k = \frac{t}{2}$.*

3 Online Steiner Tree with Outliers

In this section we consider the Online Steiner Tree problem with Outliers (outOST). We consider the case that the input distribution is the uniform distribution, while the generalization for any distribution will appear in the full version of the paper.

First, let us assume $k \geq c \log n$ for a large enough constant $c > 0$. We next describe an algorithm `outost-large` with $O(\log^2 n)$ competitive ratio, which connects at least $(1-\epsilon)k$ terminals with high probability, for any given constant parameter $\epsilon > 0$.

A crucial step is constructing a Bartal tree $\mathcal{B}$ over the input graph G using the algorithm in [9]. We recall that $\mathcal{B} = (W, F)$ is a rooted tree, with edge costs $c_{\mathcal{B}} : F \rightarrow \mathbb{R}^+$, whose leaves are the nodes V, and such that the following two properties hold:

1. Edges at the same level in the tree have the same cost and given edges e and f at level i and $i+1$, respectively (the root is at level zero), $c_{\mathcal{B}}(e) = 2c_{\mathcal{B}}(f)$.
2. For any two leaves $u, v \in \mathcal{B}$, $\frac{1}{O(\log n)} E[dist_{\mathcal{B}}(u,v)] \leq dist_G(u,v) \leq dist_{\mathcal{B}}(u,v)$.

Algorithm `outost-large` is described in Figure 1. The algorithm starts with two preprocessing steps. Initially it computes a Bartal tree $\mathcal{B}$ for G, and partitions its leaves from left to right into groups $V_1, V_2, \ldots, V_{n/\sigma}$ of size $\sigma = \alpha \frac{n}{t} \log n$ each, for

a constant α to be fixed later[10]. Then the algorithm samples t nodes $\tilde{T}$, and constructs a Steiner tree $\tilde{\mathcal{S}}$ (*anticipatory solution*) on k such nodes $\tilde{K}$, using a $\rho_{outST} = O(1)$ approximation algorithm for (offline) outST [12][11]. We call *azure* and *blue* the nodes in $\tilde{T}$ and $\tilde{K}$, respectively. We also call *blue* the groups containing at least one blue node, and *boundary* the leftmost and rightmost blue groups. The Steiner tree $\mathcal{S}$ under construction is initially set to $\tilde{\mathcal{S}}$.

In the online part of the algorithm, each time a new terminal $v \in T$ arrives, v is added to the set K of selected terminals if and only if v belongs to a non-boundary blue group. In that case, the algorithm also adds to $\mathcal{S}$ a shortest path from v to $\mathcal{S}$. We call *orange* and *red* the nodes in T and K, respectively. It turns out that the connection of orange nodes in blue groups can be conveniently charged to the cost of the anticipatory solution (boundary blue groups are excluded for technical reasons).

Let us initially bound the number of red nodes, that is, the number of terminals connected by the algorithm.

Lemma 2. *For any $\epsilon > 0$ and $\sigma = \alpha \frac{n}{t} \log n$, there is a choice of $\alpha > 0$ such that the number of red nodes is at least $(1-\epsilon)k$ with high probability.*

Proof. The number N_i of azure (resp., orange) nodes in a given group V_i, counting repetitions, satisfies $\mathrm{E}[N_i] = \frac{t}{n}\frac{n}{t}\alpha \log n = \alpha \log n$. Let $\delta \in (0,1)$ be a sufficiently small constant. By Chernoff's bounds, we know that there is a value of $\alpha > 0$ such that the probability of the event $\{N_i \notin [(1-\delta)\alpha \log n, (1+\delta)\alpha \log n]\}$ is smaller than any given inverse polynomial in n. Hence, from the union bound, with high probability all the groups contain between $(1-\delta)\alpha \log n$ and $(1+\delta)\alpha \log n$ azure (resp., orange) nodes. Let us assume from now on that this event happens. Recall that by assumption $k \geq c \log n$ for a sufficiently large constant $c > 0$.

Each blue group contains at most $(1+\delta)\alpha \log n$ azure (and hence blue) nodes. Therefore, there are at least $\frac{k}{(1+\delta)\alpha \log n}$ blue groups, and so the number of orange nodes in non-boundary blue groups (i.e. the number of red nodes) is at least

$$(1-\delta)\alpha \log n \left(\frac{k}{(1+\delta)\alpha \log n} - 2 \right) \geq \frac{1-\delta}{1+\delta} k - 2\frac{(1-\delta)\alpha}{c} k.$$

The latter quantity is at least $(1-\epsilon)k$ for proper constants c and δ. □

We continue by proving the following basic tool lemma that will be reused for outOFL later on. Refer to Figure 2. Let r_v (resp., ℓ_v) be the first blue node to the right (resp., left) of node $v \in K$ (with respect to the given ordering of leaves from left to right). Note that r_v and ℓ_v are well defined, since the boundary blue groups are not used to define K.

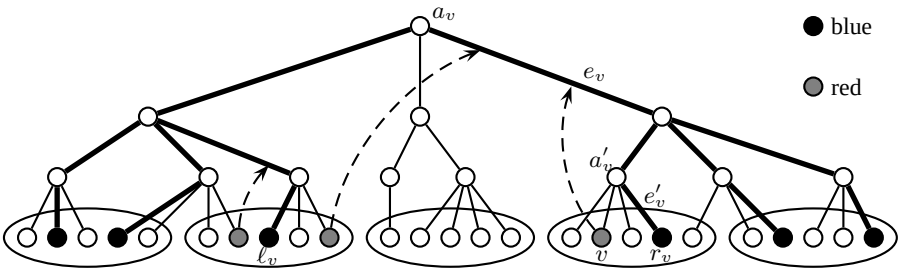
Lemma 3. *Let $\tilde{\mathcal{B}}$ be any subtree in $\mathcal{B}$ spanning nodes in $\tilde{K}$. Then*

$$\mathrm{E}\left[\sum_{v \in K} dist_{\mathcal{B}}(v, r_v)\right] \leq 8\sigma \tfrac{t}{n} \mathrm{E}[c_{\mathcal{B}}(\tilde{\mathcal{B}})].$$

[10] To avoid inessential technicalities, we will always assume that n is a multiple of σ.

[11] Since the cost of an MST spanning a set of vertices W is at most twice the corresponding cost of the best Steiner tree connecting those vertices, we can obtain a constant approximation for the outST problem if we have a constant approximation for the k-MST problem

Figure 2 Charging scheme in the analysis of `outost-large`. Bold edges indicate the subtree $\tilde{\mathcal{B}}$. Groups are enclosed into ellipses. Dashed arcs reflect the charging of red nodes connections to the edges of $\tilde{\mathcal{B}}$.



Proof. The idea of the proof it to charge the distances $dist_{\mathcal{B}}(v, r_v)$ to a proper subset of edges $\tilde{E} \subseteq E(\tilde{B})$, so that each such edge is charged $O(\sigma \frac{t}{n})$ times in expectation. Let a_v (resp., a'_v) be the lowest common ancestor of ℓ_v (resp., v) and r_v. Let moreover e_v (resp., e'_v) be the first edge along the path from a_v (resp., a'_v) to r_v. (See also Figure 2). Since v lies between ℓ_v and r_v, the level of a'_v is not higher than the level of a_v. We can conclude by Property 1 of Bartal trees that $c_{\mathcal{B}}(e'_v) \leq c_{\mathcal{B}}(e_v)$. Property 1 also implies that $dist_{\mathcal{B}}(v, r_v) = dist_{\mathcal{B}}(v, a'_v) + dist_{\mathcal{B}}(a'_v, r_v) \leq 4c_{\mathcal{B}}(e'_v)$. Altogether, we obtain

$$\mathrm{dist}_{\mathcal{B}}(v, r_v) \leq 4c_{\mathcal{B}}(e_v). \tag{1}$$

Let $\tilde{E} := \cup_{v \in K} e_v \subseteq E(\tilde{\mathcal{B}})$. Consider any edge $e = e_w \in \tilde{E}$. Any red node u to the left of ℓ_w or to the right of r_w satisfies $e_u \neq e_w$. We conclude that the set $\tilde{V}_e := \{v \in K : e_v = e\}$ is a subset of the red nodes contained in the groups of r_w and ℓ_w. Then

$$\mathrm{E}\left[\sum_{v \in K} c_{\mathcal{B}}(e_v)\right] = \mathrm{E}\left[\sum_{e \in \tilde{E}} |\tilde{V}_e| \cdot c_{\mathcal{B}}(e)\right] \leq 2\sigma \frac{t}{n} \mathrm{E}\left[\sum_{e \in \tilde{E}} c_{\mathcal{B}}(e)\right] \leq 2\sigma \frac{t}{n} \mathrm{E}\left[c_{\mathcal{B}}(\tilde{\mathcal{B}})\right]. \tag{2}$$

The lemma follows by summing up over v the expectation of (1) and combining it with (2). □

We are now ready to bound the competitive ratio of the algorithm.

Lemma 4. *The expected cost of the solution computed by algorithm* `outost-large` *is $O(\sigma \frac{t}{n} \log n)$ times the expected cost of the optimum offline solution.*

Proof. The anticipatory problem instance is sampled from the same distribution as the real problem instance, so $\mathrm{E}[c(\tilde{\mathcal{S}})] \leq \rho_{outST} \cdot opt = O(opt)$.

Let us bound the cost C_{on} paid by the algorithm during the online phase. Consider the minimal subtree $\tilde{\mathcal{B}}$ of $\mathcal{B}$ spanning $\tilde{K} \cup \{r\}$. Of course, $\tilde{\mathcal{B}}$ is an optimal Steiner tree over $\tilde{K} \cup \{r\}$ with respect to graph $\mathcal{B}$. It follows from Property 2 and the fact that the cost of a minimum spanning tree is twice the cost of a Steiner tree that connects the same vertices that

$$\mathrm{E}[c_{\mathcal{B}}(\tilde{\mathcal{B}})] \leq \mathrm{E}[2\, O(\log n) c(\tilde{\mathcal{S}})] = O(\log n) \cdot opt. \tag{3}$$

We have

$$C_{on} \leq \sum_{v \in K} dist_G(v, \tilde{K}) \overset{\text{Prop. 2}}{\leq} \sum_{v \in K} dist_{\mathcal{B}}(v, \tilde{K}) \leq \sum_{v \in K} dist_{\mathcal{B}}(v, r_v). \quad (4)$$

$\tilde{\mathcal{B}}$ satisfies the conditions of Lemma 3, hence by putting everything together we obtain

$$\mathrm{E}[C_{on}] \overset{(4)}{\leq} \mathrm{E}\left[\sum_{v \in K} dist_{\mathcal{B}}(v, r_v)\right] \overset{\text{Lem. 3}}{\leq} 8\sigma \frac{t}{n} \mathrm{E}[c_{\mathcal{B}}(\tilde{\mathcal{B}})] \overset{(3)}{=} O\left(\sigma \frac{t}{n} \log n\right) \cdot \mathit{opt}. \qquad \square$$

Note that up to now we have assumed that $k = \Omega(\log n)$. The following simple algorithm, `outost-small`, has competitive ratio $O(k)$ (proof omitted), so it can be applied in the case that $k = O(\log n)$.

Let W be the set of the $(1-\delta)\frac{n}{t}k$ nodes which are closest to the root (breaking ties arbitrarily). Here $\delta \in (0,1)$ is a proper constant. Whenever a new node $v \in T$ arrives, `outost-small` adds it to the set K of selected nodes iff $v \in W$. In that case, the algorithm connects v to the current tree $\mathcal{S}$ via a shortest path.

Let `outost` be the (polynomial-time) algorithm for outOST which either runs `outost-small` for $k < c \log n$, or `outost-large` with $\sigma = \alpha \frac{n}{t} \log n$ otherwise. The following theorem easily follows from Lemmas 2 and 4.

Theorem 5. *For any given $\epsilon > 0$ and for $\sigma = \alpha \frac{n}{t} \log n$, Algorithm* `outost` *connects at least $(1-\epsilon)k$ terminals with high probability. The expected cost of the solution is $O\left(\sigma \frac{t}{n} \log n\right) = O(\log^2 n)$ times the expected cost of the optimum offline solution.*

4 Online Facility Location with Outliers

In this section we consider the Online Facility Location problem with Outliers (outOFL). Like in the case of outOST, let us assume that $k \geq c \log n$ for a sufficiently large constant $c > 0$, while again a simple algorithm can handle the case that $k \leq c \log n$.

Our algorithm `outofl-large` is described in Figure 3. Let $G_r = (V \cup r, E')$ be a graph obtained from G by adding a new vertex r and connecting it to all other vertices v with edges of cost $\mathrm{o}(v)$. We denote by c_{G_r} the edge weights of G_r. Note that every facility location solution $\mathcal{F} = (F, K)$ in G can be mapped to a Steiner tree $T_{\mathcal{F}}$ in G_r spanning $K \cup \{r\}$ with the same cost: it is sufficient to augment the connection paths in $\mathcal{F}$ with the edges between open facilities and r. Unfortunately, solving a outOST problem on G_r is not sufficient to solve the original outOFL problem. This is because not every tree in G_r corresponds to a valid facility location solution. Nevertheless, the graph G_r is very useful in our case as it allows to introduce a convenient metric into the facility location problem. (See Figure 4 for an example of graph G_r, and a corresponding implementation of Steps 3.1 and 3.2).

First of all note that the set of nodes that are selected by the algorithm are defined in the same way as in Algorithm `outost-large`, that is, by a constant approximation to the (offline) outFL problem on a set of sampled terminals $\tilde{T}$, using, for example, the algorithm of Charikar et al. [7]. Hence, Lemma 2 holds here as well. Therefore, we only need to show that the cost of the online solution is small. The proof of the following theorem will appear in the full version of the paper.

Figure 3 Algorithm `outofl-large` for outOFL.

(Preprocessing Phase)

Step 1. Construct the graph G_r and compute a Bartal tree $\mathcal{B}$ for G_r. Partition the leaves of $\mathcal{B}$ from left to right in groups $V_1, \ldots, V_{n/\sigma}$ of size σ.

Step 2. Sample t nodes $\tilde{T}$ from the input probability distribution. Compute a ρ_{outFL}-approximate solution $\tilde{\mathcal{F}} = (\tilde{F}, \tilde{K})$ to the (offline) facility location problem with outliers induced by $\tilde{T}$, where $\tilde{F}$ and $\tilde{K}$ are the open facilities and the selected set of k terminals, respectively. Let $\mathcal{K}$ be the nodes of groups with at least one node in $\tilde{K}$, excluding the leftmost and rightmost such groups. Open the facilities in $\tilde{F}$.

(Online Phase)

Step 3. For each input node $v \in T$, if $v \in \mathcal{K}$, add v to K. Let r_v be the first node from $\tilde{K}$ to the right of v. Consider the shortest path π from v to r_v in G_r:

- **Step 3.1.** If π goes through r, then let (f_v, u) be the first edge on π such that $u = r$. Open facility f_v, if not already open, and connect v to f_v.
- **Step 3.2.** Otherwise connect v to the facility f_v to which node r_v is connected in $\tilde{\mathcal{F}}$.

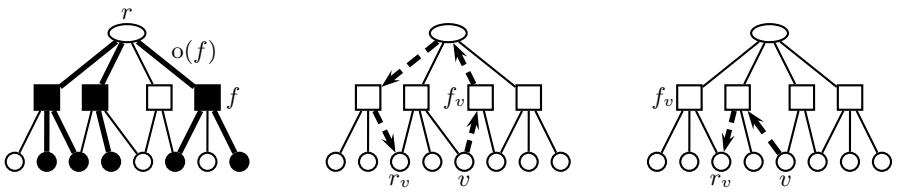
Theorem 6. *For any given $\epsilon > 0$, Algorithm* `outofl` *connects at least $(1-\epsilon)k$ terminals with high probability. The expected cost of the solution is $O(\sigma \frac{t}{n} \log n) = O(\log^2 n)$ times the expected cost of the optimum offline solution.*

Acknowledgements We would like to thank Anupam Gupta for fruitful discussions. This work was partially supported by the Polish Ministry of Science grant N206 355636.

References

1. M. Babaioff, N. Immorlica, D. Kempe, and R. Kleinberg. A knapsack secretary problem with applications. In *APPROX '07: Proceedings of the 10th International Workshop on Approximation*, pages 16–28, 2007.
2. M. Babaioff, N. Immorlica, D. Kempe, and R. Kleinberg. Online auctions and generalized secretary problems. *SIGecom Exch.*, 7(2):1–11, 2008.
3. M. Babaioff, N. Immorlica, and R. Kleinberg. Matroids, secretary problems, and online mechanisms. In *SODA '07*, pages 434–443, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.
4. Y. Bartal. On approximating arbitrary metrics by tree metrics. In *STOC'98: Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 161–168, 1998.
5. A. Borodin and R. El-Yaniv. *Online computation and competitive analysis*. Cambridge University Press, New York, NY, USA, 1998.
6. F. T. Bruce and T. S. Ferguson. Minimizing the expected rank with full information. *Journal of Applied Probability*, 30(3):616–626, 1993.
7. M. Charikar, S. Khuller, D. M. Mount, and G. Narasimhan. Algorithms for facility location problems with outliers. In *SODA '01: Proceedings of the 12th annual ACM-SIAM symposium on Discrete algorithms*, pages 642–651, 2001.
8. E. B. Dynkin. The optimum choice of the instant for stopping a markov process. *Sov. Math. Dokl.*, 4, 1963.
9. J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *Journal of Computer and System Sciences*, 69(4):485–497, 2004.
10. A. Fiat and G. J. Woeginger, editors. *Online algorithms*, volume 1442 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 1998.
11. P. Freeman. The secretary problem and its extensions: a review. *Internat. Statist. Rev.*, 51(2):189–206, 1983.

Figure 4 An example of graph G_r is given in (a). For clarity of illustration, we distinguished between terminals (circles) and facilities (squares). The oval node is the root. Terminals $\tilde{K}$ and the open facilities in the corresponding anticipatory solution are drawn in bold, as well as the associated Steiner tree $T_{\mathcal{F}}$. Examples of Steps 3.1 and 3.2 are given in (b) and (c), respectively (bold edges indicate one possible shortest path from v to r_v)



(a) (b) (c)

12. N. Garg. Saving an epsilon: a 2-approximation for the k-mst problem in graphs. In *STOC '05: Proceedings of the 37th annual ACM symposium on Theory of computing*, pages 396–402, 2005.
13. N. Garg, A. Gupta, S. Leonardi, and P. Sankowski. Stochastic analyses for online combinatorial optimization problems. In *SODA 2008*, pages 942–951, 2008.
14. J. P. Gilbert and F. Mosteller. Recognizing the maximum of a sequence. *Journal of the American Statistical Association*, 61(313):35–73, 1966.
15. M. T. Hajiaghayi, R. Kleinberg, and D. C. Parkes. Adaptive limited-supply online auctions. In *EC '04: Proceedings of the 5th ACM conference on Electronic commerce*, pages 71–80, 2004.
16. M. Imase and B. M. Waxman. Dynamic Steiner tree problem. *SIAM J. Discrete Math.*, 4(3):369–384, 1991.
17. S. Irani and A. R. Karlin. On online computation. In D. Hochbaum, editor, *Approximation Algorithms for NP Hard Problems*. PWS publishing Co, 1996.
18. A. R. Karlin, S. J. Phillips, and P. Raghavan. Markov paging. *SIAM J. Comput.*, 30(3):906–922, 2000.
19. S. Karlin. Stochastic models and optimal policy for selling an asset. *Studies in applied probability and management science*, pages 148–158, 1962.
20. D. Kennedy. Prophet-type inequalities for multichoice optimal stopping. *Stoch. Proc. Appl.*, 24(1):77–88, 1987.
21. R. Kleinberg. A multiple-choice secretary algorithm with applications to online auctions. In *SODA '05*, pages 630–631, 2005.
22. A. J. Kleywegt and J. D. Papastavrou. The dynamic and stochastic knapsack problem. *Oper. Res.*, 46(1):17–35, 1998.
23. N. Korula and M. Pal. Algorithms for secretary problems on graphs and hypergraphs. *CoRR*, abs/0807.1139, 2008.
24. E. Koutsoupias and C. H. Papadimitriou. Beyond competitive analysis. *SIAM J. Comput.*, 30(1):300–317, 2000.
25. D. V. Lindley. Dynamic programming and decision theory. *Applied Statistics*, 10:39–51, March 1961.
26. A. Meyerson. Online facility location. In *FOCS'01*, pages 426–431, 2001.
27. P. Raghavan. A statistical adversary for on-line algorithms. In *Online Algorithms*, volume 53 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 79–83. 1991.
28. D. J. Rosenkrantz, R. E. Stearns, and P. M. Lewis, II. An analysis of several heuristics for the traveling salesman problem. *SIAM J. Comput.*, 6(3):563–581, 1977.
29. D. D. Sleator and R. E. Tarjan. Amortized efficiency of list update and paging rules. *Comm. ACM*, 28(2):202–208, 1985.
30. N. E. Young. On-line paging against adversarially biased random inputs. *J. Algorithms*, 37(1):218–235, 2000.