Noname manuscript No.
(will be inserted by the editor)

Online Network Design with Outliers

Aris Anagnostopoulos · Fabrizio Grandoni · Stefano Leonardi · Piotr Sankowski

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 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 it acts accordingly. We cast three classical network design problems into this framework:

- *Online Steiner Tree with Outliers*. In this case a set of t terminals that belong to an n-node graph is presented, one at a time, to an algorithm. Each time a new terminal arrives, the algorithm can either discard or select it. In the latter case, the algorithm connects it to the Steiner tree under construction (initially consisting of a given root node). At the end of the process, at least k terminals must be selected.
- *Online TSP with Outliers*. This is the same problem as above, but with the Steiner tree replaced by a TSP tour.
- *Online Facility Location with Outliers*. In this case, we are also given a set of facility nodes, each one with an opening cost. Each time a terminal is selected, we have to connect it to some facility (and open that facility, if it is not already open).

A preliminary version of this work was presented in the Proceedings of the 37th International Colloquium on Automata, Languages and Programming (ICALP 2010) [1]

Partially supported by the EU FET projects MULTIPLEX 317532 and SIMPOL 610704, the ERC Starting Grants NEWNET 279352 and PAAl 259515, and the Google Focused Research Award Algorithms for Large-Scale Data Analysis.

A. Anagnostopoulos and S. Leonardi
Sapienza University of Rome, Rome, Italy. E-mail: {aris, leonardi}@dis.uniroma1.it

F. Grandoni
DSIA, University of Lugano. E-mail: fabrizio@idsia.ch

P. Sankowski
Sapienza Univerity of Rome, Rome, Italy and University of Warsaw, Warsaw, Poland. E-mail: sankowski@dis.uniroma1.it

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 for the case that one insists on selecting exactly k terminals, and by lower bounds including an $\Omega(\log n/\log\log n)$ lower bound for the case that the online algorithm is allowed to select $\alpha\, k$ terminals only, for a sufficiently large constant $\alpha \in (0,1)$.

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.

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 it 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 that 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 to choose some percentage of the population (say, 90%), and to cover 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 three classical online network design problems, which we define later: *Online Steiner Tree with Outliers* (outOST), *Online TSP with Outliers* (outOTSP), and *Online Facility Location with Outliers* (outOFL).

In more detail, 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 to minimize the final cost of the tree. The input for the *Online Traveling Salesman Problem* (OTSP) is the same as in OST. The difference is that here the solution is a permutation ϕ of the input terminals. (Initially, we have $\phi = (r)$). 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 $o : V \to \mathbb{R}^+$. 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 $\sum_{e \in F} f(e) + \sum_{v \in K} dist_G(v, f_v)$, where $F = \cup_{v \in K} f_v$ is the set of open facilities[1].

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 [19, 29, 32]. Garg et al. [15] 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[2]. This shows a logarithmic approximability gap between worst-case and stochastic variants of online problems.

In the generalization of the above problems with outliers, 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 to present k terminals with connection cost $M \gg k$. If the online algorithm selects some element among them, the next elements have connection cost 0. Otherwise, the next elements have connection cost M^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 [15], from now on we focus our attention on the *stochastic* setting, where terminals are sampled from a given probability distribution[3]. When a request for a node v arrives according to the input distribution, we say that *node v is sampled*. 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 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, whereas 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 a $\Omega(\log n / \log \log n)$ 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 this result 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 satisfy $k - O(k^\alpha)$ requests for $0 \leq \alpha < 1$.

[1] 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 that are not facilities, and let $\mathcal{F} = V$.

[2] 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 complicated).

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

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[4], 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 an $O(k)$ 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 and subsequent researchers [6,11]. A Bartal tree of the input graph is used to partition the nodes into a collection of groups of size $\Theta(\frac{n}{t}\log n)$ each. Note that because t requests arrive overall, $\Theta(\log n)$ terminals are sampled in each group whp. 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 optimum solution. Altogether, this gives a $O(\log^2 n)$ competitive ratio. In this ratio, the factor $O(\log n)$ originating from tree embeddings seems inherent in our algorithmic approach, and therefore hard to remove. The extra factor $O(\log n)$ is due to the use of probability concentration bounds in the analysis: here a refined analysis (and slight adaptation of the algorithm) might give an improvement.

The results on outOST immediately generalize to the case of outOTSP, modulo constant factors. The idea 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 one obtains the desired permutation ϕ of selected terminals. The Euler tour can be updated in each step such that the new permutation obtained preserves the relative order of the terminals in the starting 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. The following lemma summarizes 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 facility costs. In this case, as well, we deploy an anticipatory solution on k out of t terminals sampled beforehand from the known

[4] 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 considered cases.

distribution. 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 facility of the graph to a virtual root in the tree metric with an edge of cost equal to the 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 and the unknown distribution models (Section 2).
- 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. First we present the results for the uniform input distribution and then we extend them to any distribution.
- We extend our algorithms to the unknown-distribution model (Section 5).

Online network design problems with outliers are in general strictly harder than their counterpart without outliers. Intuitively, this is because of the fact that a wrong choice can be very costly. Indeed, in [15] 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, $c(A) \stackrel{\triangle}{=} \sum_{a \in A} c(a)$. For a graph A, we use $c(A)$ as a shortcut for $c(E(A))$.

1.1 Related Work

The problems in this paper are generalizations of various online problems where irrevocable decisions are being made. We elaborate on the most related areas, namely the problems of online network design and the secretary problems.

1.1.1 Online Network Design

Competitive analysis of online algorithms has a long history (see, for example, [7, 12, 33] 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 [19, 29, 32]. There have been many attempts to relax the notion of competitive analysis for classical list-update, paging and k-server problems (see [7, 12, 20, 21, 27, 31, 35]).

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 expected online cost and the expected optimal cost are constant for the known distribution model [15]. In the random permutation model, Meyerson [29] 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 [15].

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 [9, 14][5]. We will exploit such (offline) approximation algorithms as part of our online algorithms.

We remark that none of the above mentioned approximation algorithms exploits tree embeddings at an algorithmic level: this is a novelty of our approach. On the other hand, the idea of constructing an anticipatory solution is not new. For example, it is used frequently in the design of approximation algorithms for 2-stage stochastic optimization problems (see e.g. [34] and references therein).

1.1.2 Secretary Problems and Prophet Inequalities

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). This problem dates back to the fifties and it has recently attracted a growing interest given its connections to selecting winners in online auctions [4, 17].

In the classical secretary problem, it is easy to achieve a constant approximation to the optimal expected value; for example, one can observe that waiting until seeing half of the elements and then selecting the first element that has value higher than the maximum of the first half will select the best element with probability at least $1/4$ and thus 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.

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 [10, 13, 16, 28]. In this classical model a set of t arbitrary numerical values is presented to the algorithm in

[5] The k-MST problem studied in [14] and the Steiner tree problem with outliers are equivalent approximation-wise, modulo constant factors. The same holds for TSP with outliers.

random order. A strategy is known that selects the best secretary with probability $1/e$ [28]. For the multiple-choice secretary problem it has been recently proposed [24] a strategy that achieves a $1 - O(\sqrt{1/k})$ fraction of the sum of the k largest elements in the sequence, that is, a competitive ratio [7] that approaches 1 for $k \to \infty$.

In the known-distribution model the numerical values are identical independent samples from a known distribution. The known-distribution model has been considered in several works (e.g., [16,23]). These problems are also known as house-selling problems (e.g., [22]), and generalizations have appeared under the name of dynamic and stochastic knapsack problems [3,25]. 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 [16]. The more general setting with nonidentical distributions is referred to as *prophet inequalities* in the theory of optimal stopping (see, e.g., [2,18]).

In the unknown-distribution model each element of the sequence is drawn from an unknown distribution. The unknown-distribution model is more general than the random permutation model if we disregard repetitions. The latter can be simulated by sampling from an unknown distribution that is uniform on the secretaries that will arrive, and zero otherwise.

Secretary problems with an underlying graph structure have been recently studied in the context of online matching problems and their generalizations [5, 26].

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 [8] and the references therein). However, to the best of our knowledge, there has not been any comparison of the online and offline solutions for arbitrary input distributions.

2 Lower Bounds

In this section we provide some lower bounds for online network design with outliers. We focus on outOST, but similar counterexamples can work for outOTSP and outOFL. First we deal with the known-distribution model. We first prove, in Theorem 1, that for $k = t - 1$ the competitive ratio can be arbitrariliy bad. Contrast this example with the results of Garg et al. [15] who showed that for $k = t$ the competitive ratio is constant. Instead, here we show that allowing for just one request to be dropped can make the competitive ratio arbitrarily bad. Next we look into the other extreme, $k = 1$ (Theorem 2). We show that in that case the competitive ratio is exponentially bad in n. As we mentioned in the introduction, this setting can be seen as a minimization version of the classical secretatry problem, and our result shows that (not surprisingly) minimization is much harder than maximization. Our final result on lower bounds (Theorem 3) addresses the case in which we allow the online algorithm to select only αk requests, for some constant $\alpha < 1$, and comparing it with the offline algorithm that selects k requests. In that

case we prove a lower bound of $\Omega(\log n/\log\log n)$. This theorem complements our upper bounds, in which we show the existence of an algorithm with competitive ratio of $O(\log^2 n)$ in the case that the online algorithm is required to select $(1-\epsilon)k$ requests, for ϵ being an arbitrarily small constant.

Let us start by proving inapproximability results for outOST in the known distribution model, when we insist on selecting at least k terminals.

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

Proof Consider the star graph whose center is the root and with three terminals v, u, and w (which can be repeatedly sampled), connected to the root with edges of weight $c(v) = 0$, $c(u) = C$, and $c(w) = C^3$. Here $C \gg t$. The sampling probabilities are $p(v) = 1 - 1/C - 1/C^2$, $p(u) = 1/C$ and $p(w) = 1/C^2$. OPT selects one copy of w only if w is sampled at least twice. Otherwise, it selects one copy of u only if u is sampled at least twice or u and w are sampled exactly once each. The probability of the first and second event is at most $\Theta(\frac{t^2}{C^4})$ and $\Theta(\frac{t^2}{C^3} + \frac{t^2}{C^2}) = \Theta(\frac{t^2}{C^2})$, respectively. Hence

$$opt \le C^3\Theta\left(\frac{t^2}{C^4}\right) + C\Theta\left(\frac{t^2}{C^2}\right) = \Theta\left(\frac{t^2}{C}\right).$$

Consider now any given online algorithm selecting a multiset K of k terminals. The probability that within $t/2$ samplings, node u and w are sampled at least once is $\Theta(\frac{t}{2C})$ and $\Theta(\frac{t}{2C^2})$, respectively. If the first event happens within the first $t/2$ samples, the algorithm can either select u or select all the following $t/2$ sampled terminals. In the first case the algorithm pays at least C. In the second case it pays at least C^3 with probability $\Theta(\frac{t}{2C^2})$. Hence

$$\mathrm{E}[c(K)] \ge \Theta\left(\frac{t}{2C}\right) \cdot \min\left\{C, C^3\Theta\left(\frac{t}{2C^2}\right)\right\} = \Theta(t).$$

The overall competitive ratio is $\Omega\left(\frac{C}{t}\right)$, which becomes arbitrarily large as C increases.

The next theorem considers the somehow opposite case that k is very small. 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 at least 1.46^n, where n is the number of nodes, for $t = 3n/4$ and $k = 1$.*

Proof Consider the star graph with root r and with n nodes connected $v_1, v_2, \ldots, v_n$, with the edge $\{r, v_i\}$ having weight $d_i = 2^i$. Another view of the problem is the following. Let $d_i = 2^i$ for $i = 1, 2, \ldots, n$, X be a random variable distributed uniformly at random in $\{d_i\}$, and assume that t mutually independent copies of X are drawn. The optimal offline solution V_t^{off} is the minimum out of t draws from the distribution. Then

$$\begin{aligned}\mathrm{E}[V_t^{\text{off}}] &= \sum_{x=1}^{\infty} \Pr(V_t^{\text{off}} \ge x) \overset{(a)}{=} \Pr(V_t^{\text{off}} \ge 1) + \sum_{i=1}^{n} 2^{i-1}\left(\frac{n-i+1}{n}\right)^t \\ &= 1 + \sum_{i=0}^{n-1} 2^i\left(1 - \frac{i}{n}\right)^t \le 1 + \sum_{i=0}^{n-1} 2^i \cdot e^{-\frac{it}{n}} = 1 + \sum_{i=0}^{n-1} e^{i\ln 2 - \frac{it}{n}},\end{aligned}$$

where (a) follows from the fact that, to have $V_t^{\text{off}} \geq x \geq 2$, for all the t copies of X it must hold $X \geq 2^{\lfloor \log x \rfloor}$. In particular, for $t = 3n/4$,

$$\mathrm{E}[V_t^{\text{off}}] = 1 + \sum_{i=0}^{n-1} e^{\left(\ln 2 - \frac{3}{4}\right) i} = O(1).$$

Now let us compute the expected value V_t^{on} of the optimal online solution. Note that the problem can be modeled as a finite-horizon Markov decision process [30], and we can compute the expected value of the optimal online strategy as follows. Let $v_t = \mathrm{E}[V_t^{\text{on}}]$, then we have that $v_1 = \mathrm{E}[X]$ and

$$v_{t+1} = v_t \cdot \Pr(X \geq v_t) + \mathrm{E}[X \mid X < v_t] \cdot \Pr(X < v_t).$$

In fact, the optimal strategy when $(t+1)$ terminals have still to be sampled is to look at the first terminal, discard it if its value is larger than v_t, and selects it otherwise. The first event happens with probability $\Pr(X \geq v_t)$, and in that case the expected future value is v_t. The second event happens with probability $\Pr(X < v_t)$, and in that case the expected value is $\mathrm{E}[X \mid X < v_t]$. Thus the expected value v_{t+1} is given by the above expression. From the theory of Markov decision processes [30] it follows that this is the optimal online stategy.

To conclude the proof we only have to show that $v_{3n/4}$ is exponential in n. Note that v_t is a decreasing function of t and define t_0 to be the smallest t for which $v_t \leq 2^{n/4}$. Define also $\tilde{v}_t = v_t$ for $t \leq t_0$ and

$$\tilde{v}_{t+1} = \frac{3}{4}\tilde{v}_t,$$

for $t > t_0$. Then we have that $\tilde{v}_t \leq 2^{n/4}$ for $t \geq t_0$. Note also that for $t \geq t_0$ we have

$$\Pr(X \geq \tilde{v}_t) = 1 - \frac{\lceil \log \tilde{v}_t \rceil - 1}{n} \geq \frac{3}{4}, \tag{1}$$

where the equality follows from the definition of the distribution, and the inequality follows from the fact that $\tilde{v}_t \leq 2^{n/4}$ for $t \geq t_0$, and thus

$$\frac{\lceil \log \tilde{v}_t \rceil - 1}{n} \leq \frac{1}{4}.$$

Now we show that $v_t \geq \tilde{v}_t$ for all $t \geq 1$. By definition it is true for $t \leq t_0$, and we use induction to prove it for $t \geq t_0 + 1$. So, assuming that it holds for $t \geq t_0$ we have

$$\begin{aligned}
v_{t+1} &= v_t \cdot \Pr(X \geq v_t) + \mathrm{E}[X \mid X < v_t] \cdot \Pr(X < v_t) \\
&\overset{(a)}{\geq} v_t \cdot \Pr(X \geq v_t) \\
&\overset{(b)}{\geq} \tilde{v}_t \cdot \Pr(X \geq \tilde{v}_t) \\
&\overset{(c)}{\geq} \frac{3}{4}\tilde{v}_t \\
&= \tilde{v}_{t+1},
\end{aligned}$$

where (a) follows from the fact that $\mathrm{E}[X \mid X < v_t] \cdot \Pr(X < v_t)$ is nonnegative, (b) follows from the fact that $v \cdot \Pr(X \geq v)$ is an increasing function of v for our distribution, and (c) follows from Equation (1). Finally we have that

$$\tilde{v}_{t_0} = v_{t_0} \geq v_{t_0-1}\left(1 - \frac{\lceil \log v_{t_0-1} \rceil - 1}{n}\right) \geq 2^{\frac{n}{4}}\left(1 - \frac{n-1}{n}\right) = \frac{2^{\frac{n}{4}}}{n},$$

since $2^{n/4} < v_{t_0-1} \leq v_1 = \mathrm{E}[X] < 2^n$. Therefore, for sufficiently large n,

$$v_t \geq \tilde{v}_t = \left(\frac{3}{4}\right)^{t-t_0} \tilde{v}_{t_0} \geq \left(\frac{3}{4}\right)^{t} \frac{2^{\frac{n}{4}}}{n} = 2^{\frac{n}{4} - t\log\frac{3}{4} - \log n},$$

and so we obtain

$$v_{3n/4} \geq 2^{\left(\frac{1}{4} - \frac{3}{4}\log\frac{3}{4}\right)n - \log n} \geq 1.46^n.$$

We can conclude that the expected online value is exponentially higher (in n) than the expected offline value.

Finally, we present an $\Omega(\frac{\log n}{\log\log n})$ lower bound for outOST, outOTSP and outOFL, which applies also to the case that the online algorithm is allowed to connect only αk 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 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 compatitive ratio is $\Omega(k) = \Omega\left(\frac{\log n}{\log\log n}\right)$ in the case considered.

3 Online Steiner Tree with Outliers

In this section we consider the Online Steiner Tree problem with Outliers (outOST). We first consider the case that the input distribution is uniform, and in Section 3.1 we generalize the results to any distribution.

The algorithm considers two cases, one for small values of k ($k = O(\log n)$) and one for large ones ($k \geq c \log n$ for a suitable constant c).

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.

First we consider the case of $k = O(\log n)$, which is handled by the following algorithm `outost-small`: Let W be the set of the $(1-\delta)\frac{n\,k}{t}$ 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.

Lemma 2 *For any $\epsilon > 0$, there is a choice of $\delta \in (0,1)$ such that Algorithm* `outost-small` *connects at least $(1-\epsilon)k$ nodes whp. The expected cost of the solution computed is $O(k)$ times the expected cost of the optimum offline solution.*

Proof Let $K = T \cap W$ be the set of connected nodes. Trivially $\mathrm{E}[|K|] = \frac{t}{n}(1-\delta)\frac{n\,k}{t} = (1-\delta)k$. By Chernoff's bounds we have that

$$\Pr(|K| < (1-\delta)\cdot \mathrm{E}[|K|]) \leq e^{-\delta^2(1-\delta)k/2}$$

and

$$\Pr(|K| > (1+\delta)\cdot \mathrm{E}[|K|]) \leq e^{-\delta^2(1-\delta)k/3}.$$

Assuming the event $\{|K| \geq (1-\delta)\cdot \mathrm{E}[|K|]\}$, which happens whp, the number of connected terminals is at least $(1-\delta)^2 k$, which is at least $(1-\epsilon)k$ for a proper choice of $\delta \in (0,1)$. Moreover, whp the sampled nodes in W are at most $(1+\delta)(1-\delta)k = (1-\delta^2)k < k$. When that happens (let us call it event A), the optimal solution must select at least one terminal not belonging in W, and therefore it has cost at least $D \stackrel{\triangle}{=} \max_{v\in W} dist_G(v,r)$. This implies that $opt \geq D\cdot\Pr(A) \geq D\cdot(1-e^{-\delta^2(1-\delta)k/3}) = \Theta(D)$. Altogether, we get

$$E[c(\mathcal{S})] \leq \mathrm{E}\left[\sum_{v\in K} dist_G(v,r)\right] \leq \mathrm{E}\left[\sum_{v\in K} D\right] = (1-\delta)kD = O(k\cdot opt) = O(\log n)\cdot opt.$$

Now we consider the more interesting case with $k \geq c\log n$ 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 whp, for any given constant parameter $\epsilon > 0$.

A crucial step in our algorithms is constructing a Bartal tree $\mathcal{B}$ over the input graph G using the algorithm in [11]. We recall that $\mathcal{B} = (W, F)$ is a randomly created rooted tree, with edge costs $c_{\mathcal{B}} : F \to \mathbb{R}^+$, 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[6]. 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 [14][7]. 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 lower bound the number of red nodes, that is, the number of terminals connected by the algorithm.

Lemma 3 *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 $(1-\epsilon)k$ whp.*

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, whp 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

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

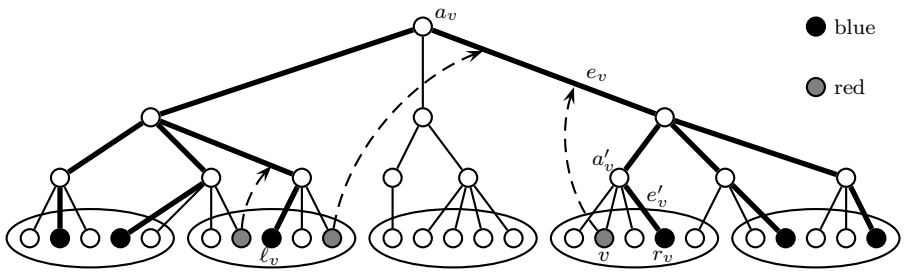
The latter quantity is at least $(1-\epsilon)k$ for proper constants c and δ.

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

[7] 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 rooted k-MST problem. Indeed, by apending a large number N (say $N = n^2$) of copies of each terminal with edges of cost 0 and multiplying k by a factor N, we can achieve for outST the same approximation factor as for k-MST, that is 2 [14].

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, because the boundary blue groups are not used to define K.

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 $\tilde{\mathcal{B}}$.



Lemma 4 *Let $\tilde{\mathcal{B}}$ be any subtree in $\mathcal{B}$ spanning the root r and the nodes in $\tilde{K}$. Then*

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

Proof The idea of the proof is 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). Because 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 $c_{\mathcal{B}}(e'_v) \le c_{\mathcal{B}}(e_v)$. Property 1 also implies that $dist_{\mathcal{B}}(v, r_v) = dist_{\mathcal{B}}(v, a'_v) + dist_{\mathcal{B}}(a'_v, r_v) \le 4c_{\mathcal{B}}(e'_v)$. Altogether, we obtain

$$\mathrm{dist}_{\mathcal{B}}(v, r_v) \le 4c_{\mathcal{B}}(e_v). \tag{2}$$

Let $\tilde{E} \triangleq \cup_{v\in K} e_v \subseteq E(\tilde{\mathcal{B}})$. 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 can conclude that the set $\tilde{V}_e \triangleq \{v \in K : e_v = e\}$ is a subset of the red nodes contained in the groups of r_w and ℓ_w. Conditioned on the constructed Bartal tree, the expected number of red nodes in those two groups is $2\sigma t/n$. Then

$$\mathrm{E}\left[\sum_{v\in K} c_{\mathcal{B}}(e_v)\right] = \mathrm{E}\left[\sum_{e\in\tilde{E}} |\tilde{V}_e| \cdot c_{\mathcal{B}}(e)\right] \le 2\sigma\frac{t}{n}\,\mathrm{E}\left[\sum_{e\in\tilde{E}} c_{\mathcal{B}}(e)\right] \le 2\sigma\frac{t}{n}\mathrm{E}\left[c_{\mathcal{B}}(\tilde{\mathcal{B}})\right]. \tag{3}$$

The lemma follows by summing up over v the expectation of (2) and combining it with (3).

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

Lemma 5 *The expected cost of the solution computed by* `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 $\mathcal{B}$. From Property 2 it follows that

$$\mathrm{E}[c_{\mathcal{B}}(\tilde{\mathcal{B}})] \leq \mathrm{E}[O(\log n) c(\tilde{\mathcal{S}})] = O(\log n) \cdot opt. \tag{4}$$

We have

$$C_{on} \leq \sum_{v \in K} dist_G(v, \tilde{K}) \overset{\text{Prop. 2}}{\leq} \sum_{v \in K} dist_{\mathcal{B}}(v, \tilde{K}) \leq \sum_{v \in K} dist_{\mathcal{B}}(v, r_v). \tag{5}$$

Note that $\tilde{\mathcal{B}}$ satisfies the conditions of Lemma 4, hence by putting everything together we obtain

$$\mathrm{E}[C_{on}] \overset{(5)}{\leq} \mathrm{E}\left[\sum_{v \in K} dist_{\mathcal{B}}(v, r_v)\right] \overset{\text{Lem. 4}}{\leq} 8\sigma \frac{t}{n} \mathrm{E}[c_{\mathcal{B}}(\tilde{\mathcal{B}})] \overset{(4)}{=} O\left(\sigma \frac{t}{n} \log n\right) \cdot opt.$$

Let `outost` be the (polynomial-time) algorithm for outOST that runs `outost-small` for $k < c \log n$, and `outost-large` with $\sigma = \alpha \frac{n}{t} \log n$ otherwise, for a sufficiently large constant $\alpha > 0$. The following theorem easily follows from Lemmas 2, 3 and 5.

Theorem 4 *Algorithm* `outost` *connects at least* $(1-\epsilon)k$ *terminals whp. The expected cost of the solution is* $O(\log^2 n)$ *times the expected cost of the optimum offline solution.*

3.1 Nonuniform Probability Distribution

In this section we show how to deal with a non-uniform probability distribution $p : V \to [0, 1]$.

Consider first the case $k = O(\log n)$. Let $v_1, v_2, \ldots, v_n$ be nodes in increasing order $d_1, d_2, \ldots, d_n$ of distance from the root (breaking ties arbitrarily), and let $p_i := p(v_i)$. The algorithm computes the largest index j such that $P_j := \sum_{i \leq j} p_i \leq P' := \frac{k}{t}(1-\delta)$, and adds nodes $v_1, v_2, \ldots, v_j$ to a set W. Moreover, it replaces node v_{j+1} with two nodes v'_{j+1} and v''_{j+1}, with sampling probability $p'_{j+1} = P' - P_j$ and $p''_{j+1} = p_{j+1} - p'_{j+1}$. Node v'_{j+1} is added to W as well. The rest of the algorithm is as in `outost-small`: all the sampled nodes in W are connected, up to k nodes. The expected number of connected nodes is $t \cdot (P_j + p'_{j+1}) = t \cdot P' = k(1-\delta)$. It follows from Chernoff's bounds that whp (in k) the number of connected terminals is at least $(1-\delta)^2 k = (1-\epsilon)k$. Moreover, whp no more than $(1+\delta)(1-\delta)k = (1-\delta^2)k < k$ terminals are connected. It follows (similarly to the uniform case)

that the maximum expected distance of a node in W from the root is $O(opt)$, opt being the expected cost of the optimal offline solution. Altogether, the connection cost paid by the algorithm is $O(k) \cdot opt$ in expectation.

The algorithm for $k = \Omega(\log n)$ is similar to `outost-large`, the main difference being the way groups are formed. Let $v_1, v_2, \ldots, v_n$ be the nodes sorted from left to right in the Bartal tree $\mathcal{B}$, and let $p_i = p(v_i)$. The first groups V_1 is formed in the following way. Let j be the largest index such that $P_j := \sum_{i \le j} p_i \le \frac{\alpha \log n}{t}$. Node v_{j+1} is split in nodes v'_{j+1} and v''_{j+1} is a similar way as above, so that $P_j + p'_{j+1} = \frac{\alpha \log n}{t}$. Then $V_1 = \{v_1, \ldots, v_j, v'_{j+1}\}$. Group V_2 is formed in a similar fashion starting from node v''_{j+1}, and the other groups are constructed analogously. By an argument analogous to the uniform case, $k(1-\epsilon)$ terminals are connected whp and the expected cost of the solution computed is $O(\log^2 n) \cdot opt$.

4 Online Facility Location with Outliers

In this section we consider the Online Facility Location problem with Outliers (outOFL). Similarly to the case of outOST, we consider two cases depending on the value of k, one for $k = O(\log n)$ and one for $k = \Omega(\log n)$.

First we consider the following algorithm `outofl-small`, which handles the case of $k = O(\log n)$. Let W be the set of the $(1-\delta)\frac{n\,k}{t}$ nodes v for which $\min_{f \in V} (\mathrm{o}(f) + \mathrm{dist}(v, f))$ is minimum. Here $\delta \in (0,1)$ is a proper constant. Whenever a new terminal $v \in T$ arrives, `outofl-small` adds it to the set K of selected nodes iff $v \in W$. In that case, the algorithm opens facility

$$f_v = \mathrm{argmin}_{f \in V} \left(\mathrm{o}(f) + \mathrm{dist}(v, f)\right),$$

if not already open, and connects v to f_v. The proof of the following lemma is analogous to the proof of Lemma 2, and is omitted.

Lemma 6 *For any given $\epsilon > 0$, there is a choice of $\delta \in (0,1)$ such that Algorithm* `outofl-small` *connects at least $(1-\epsilon)k$ nodes whp. The expected cost of the solution computed is $O(k) = O(\log n)$ times the expected cost of the optimum offline solution.*

Now we consider the case that $k \ge c \log n$ for a sufficiently large constant $c > 0$. 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 vertices f on which facilities can be opened with edges of cost $\mathrm{o}(f)$. 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).

The following two lemmas can be proved similarly to Lemmas 3 and 4, and we omit their proof.

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.

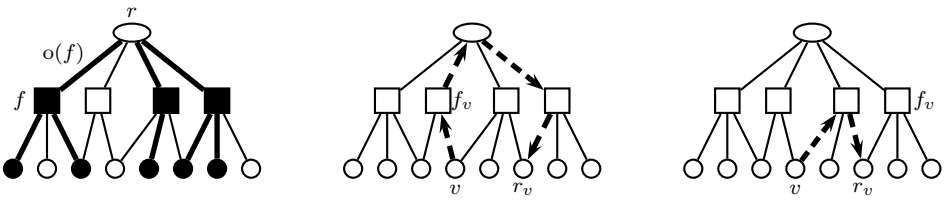
Step 2. Sample t nodes $\tilde{T}$ from the input probability distribution. Compute a ρ_{outFL}-approximate solution $\tilde{\mathcal{F}} = (\tilde{F}, \tilde{K})$ to the (offline) facility location problem with outliers induced by $\tilde{T}$, where $\tilde{F}$ and $\tilde{K}$ are the open facilities 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 $\tilde{\mathcal{F}}$.

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 full, 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)

Lemma 7 *For any $\epsilon > 0$ and $\sigma = \alpha \frac{n}{t} \log n$, there is a choice of $\alpha > 0$ such that the number of nodes in K at the end of the algorithm is at least $(1-\epsilon)k$ whp.*

Lemma 8 *Let $\tilde{\mathcal{B}}$ be any subtree in $\mathcal{B}$ spanning the root r and the nodes in $\tilde{K}$. Then*

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

Given Lemma 8, we can bound the expected cost of the solution.

Lemma 9 *The expected cost of the solution computed by* `outofl-large` *is $O\left(\sigma \frac{t}{n} \log n\right)$ times the expected cost of the optimum offline solution.*

Proof The expected cost of the anticipatory solution $\tilde{\mathcal{F}}$ is at most $\rho_{outFL} \cdot opt = O(opt)$.

Consider the solution $\tilde{\mathcal{F}}$ and the corresponding tree $T_{\tilde{\mathcal{F}}}$ in G_r. Define π_e for each edge $e = (u, v) \in E$ to be a path in $\mathcal{B}$ from u to v. Let us define:

$$\tilde{\mathcal{B}} = \bigcup_{e \in T_{\tilde{\mathcal{F}}}} \pi_e.$$

From Property 2 of Bartal trees, it follows that

$$\mathrm{E}[c_{\mathcal{B}}(\tilde{\mathcal{B}})] \leq O(\log n) \cdot \mathrm{E}[c_{G_r}(T_{\tilde{\mathcal{F}}})] \leq O(\log n) \cdot opt. \tag{6}$$

Note that $\tilde{\mathcal{B}}$ is a subtree of $\mathcal{B}$ that spans r and the nodes in $\tilde{K}$. Hence we can apply Lemma 8 to get:

$$\begin{aligned} \mathrm{E}\left[\sum_{v \in K} \mathrm{dist}_{G_r}(v, r_v)\right] &\overset{\text{Prop. 2}}{\leq} \mathrm{E}\left[\sum_{v \in K} \mathrm{dist}_{\mathcal{B}}(v, r_v)\right] \\ &\overset{\text{Lem. 8}}{\leq} 8\sigma \frac{t}{n} \mathrm{E}[c_{\mathcal{B}}(\tilde{\mathcal{B}})] \\ &\overset{(6)}{\leq} O\left(\sigma \frac{t}{n} \log n\right) \cdot opt. \end{aligned} \tag{7}$$

The cost of selecting v in Step 3.1 is no more than the length of π, which in turn is equal to $\mathrm{dist}_{G_r}(v, r_v)$. The cost of selecting v in Step 3.2 is no more than the length of π plus the distance from r_v to f_v. Hence, the cost C_{on} of the algorithm paid during the online phase is:

$$C_{on} \leq \sum_{v \in K} \left(\mathrm{dist}_{G_r}(v, r_v) + \mathrm{dist}_G(r_v, f_v)\right). \tag{8}$$

Note that each $r_v \in \tilde{K}$ is the first node to the right for nodes in at most two groups. Hence, the distance $\mathrm{dist}_G(r_v, f_v)$ is charged in expectation at most $2\sigma\frac{t}{n}$ times, conditioned on the requests arrived in the first round. Then

$$\mathrm{E}\left[\sum_{v \in K} \mathrm{dist}_G(r_v, f_v)\right] \leq 2\sigma \frac{t}{n} \mathrm{E}\left[\sum_{v \in \tilde{K}} \mathrm{dist}_G(r_v, f_v)\right] \leq 2\sigma \frac{t}{n} \rho_{outFL} \cdot opt. \tag{9}$$

Taking the expectation of (8) and combining it with (7) and (9), we get

$$\mathrm{E}[C_{on}] \leq E\left[\sum_{v \in K} \mathrm{dist}_{G_r}(v, r_v)\right] + \mathrm{E}\left[\sum_{v \in K} \mathrm{dist}_G(r_v, f_v)\right] \leq O\left(\sigma \frac{t}{n} \log n\right) \cdot opt.$$

Let `outofl` be the (polynomial-time) algorithm for outOFL that runs `outofl-small` for $k < c \log n$, and `outofl-large` with $\sigma = \alpha \frac{n}{t} \log n$ otherwise, for a sufficiently large constant $\alpha > 0$. The following theorem is an easy consequence of Lemmas 6, 7 and 9.

Theorem 5 *For any given $\epsilon > 0$, Algorithm* `outofl` *connects at least $(1 - \epsilon)k$ terminals whp. The expected cost of the solution is $O(\log^2 n)$ times the expected cost of the optimum offline solution.*

Figure 5 Algorithm `outost-large-unk` for outOST with unknown probability distributions. Here, β and δ are small constants, while a is a large constant. We assume that $k > c \log(n\Delta_r)$ for a large enough constant c.

(Online Observe Phase)

Step 1. Observe the first $\tilde{t} = \beta t$ terminals $\tilde{T}$ from the input sequence.

(Limiting Phase)

Step 2. Compute a Bartal tree $\mathcal{B}$ for the input graph. Partition the leaves of $\mathcal{B}$ from left to right in groups $V_1, \ldots, V_p$ each containing exactly $(1-\delta)\beta\, a \log n$ terminals from $\tilde{T}$.

Step 3. Compute a ρ_{OST}-approximate solution $\tilde{\mathcal{S}}$ to the OST problem induced by $\tilde{T}$ with threshold $\tilde{k} = (1-\delta)\beta k$. Let $\tilde{K}$ be the resulting set of $\tilde{k}$ terminals, and $\mathcal{K}$ be the nodes of groups with at least one element in $\tilde{K}$, excluding the leftmost and rightmost such groups. Set $\mathcal{S} = \tilde{\mathcal{S}}$.

(Online Cutoff Phase)

Step 4. For the remaining $(1-\beta)t$ terminals T', if $v \in \mathcal{K}$, add v to K and augment $\mathcal{S}$ with a shortest path to v.

5 Algorithm for Unknown Probability Distributions

In this section we present an approximation algorithm for the case of unknown probability distribution. Also in this case we focus on outOST: analogous arguments work for the other problems.

In the case of standard secretary problems, this scenario can be addressed by observing a small fraction of the input. Here we use a similar approach. However, the whole argument is more subtle in our case. For this reason, we make an additional assumption on the skewness of the metric space: we assume that Δ_r, the ratio of the distances of the farthest and closest terminal to the root r, is not too large. In more detail, we assume $k > c \log(n\Delta_r)$ for a sufficiently large constant $c > 0$.

Our algorithm `outost-large-unk` is described in Figure 5. The algorithm initially (Step 1) observes the first $\tilde{t} = \beta\, t$ input terminals $\tilde{T}$ (*azure* terminals), for a small constant $\beta > 0$. Then (Step 2) it computes a Bartal tree, and partitions its leaves in groups containing exactly $(1-\delta)\beta\, a \log n$ azure terminals each, where $\delta > 0$ is a small constant and $a > 0$ a large one. Note that formerly (see Section 3.1), the partition in groups was based on the probability distribution, which is unknown here. In Step 3, the algorithm constructs an anticipatory solution $\tilde{\mathcal{S}}$ on a set $\tilde{K}$ of $\tilde{k} = (1-\delta)\beta\, t$ azure terminals (*blue* terminals), using a ρ_{OST} approximation algorithm for OST. A crucial fact is that the expected cost of $\tilde{\mathcal{S}}$ is of the same order of the cost of the optimum solution (Lemma 10). We call *blue* the groups containing at least one blue terminal, excluding the boundary such groups. The set of nodes in blue groups is denoted by $\mathcal{K}$. Eventually (Step 4), the algorithm, among the remaining $(1-\beta)t$ terminals T' (*orange* terminals), select the ones (*red* terminals) falling in a blue group (i.e., belonging to $\mathcal{K}$). Whp, the number of red terminals in each group is close to $(1-\delta)(1-\beta)\, a \log n$ (Lemma 11). The claim

follows (Theorem 6). The values of constants β, δ, a and c depend on ϵ, and are fixed later on.

Without loss of generality, we assume that βt and $(1-\delta)\beta k$ are integral. Let us denote by $e_{k,t}$ the expected cost of the solution for threshold k over t nodes. In particular, $e_{(1-\delta)\beta k,\beta t}$ is the expected cost of the anticipatory solution $\tilde{\mathcal{S}}$.

Lemma 10 *For any $\beta, \delta \in [0,1]$, and for $t \leq n$, there exists $c > 0$ such that for $k > c\log(n\Delta_r)$ we have $e_{(1-\delta)\beta k,\beta t} \leq 2e_{k,t}$.*

Proof Consider a sequence T of t random terminals and let $\mathcal{S}$ be a solution to T. Moreover, let K be the set of k selected terminals. Note that by taking a random subset of βt nodes from T we obtain a random sequence $\tilde{T}$ of βt nodes. From Chernoff's bounds the probability that $\tilde{T}$ contains fewer than $(1-\delta)\beta k$ nodes from K_T is bounded by

$$\Pr(|\tilde{T} \cap K| < (1-\delta)\beta k) \leq e^{-\beta k\delta^2/2} \leq (n\Delta_r)^{-c\beta\delta^2} \leq \frac{1}{n\Delta_r}, \tag{10}$$

for $c \geq \frac{1}{\beta\delta^2}$. Hence, with probability $1-\frac{1}{n\Delta_r}$ the OST instance defined by $\tilde{T}$ can be solved using a solution for T whose expected cost is bounded by $e_{k,t}$. Otherwise, the cost of the solution is no higher than $\Delta_r t e_{k,t}$. Therefore, the expected cost of the anticipatory solution is bounded by

$$e_{(1-\delta)\beta k,\beta t} \leq e_{k,t} + \frac{1}{n\Delta_r}\Delta_r t e_{k,t} \leq 2e_{k,t}.$$

Lemma 11 *For every $\delta, \beta, \sigma \in [0,1]$, there exist constants $a, c > 0$ such that for $t \geq c\log n$ each group contains at least $(1-2\sigma)a(1-\delta)(1-\beta)\log n$ and no more than $(1+2\sigma)a(1-\delta)(1-\beta)\log n$ orange nodes whp.*

Proof Let π be the input probability distribution, which is hidden from the algorithm. Virtually partition the leaves of $\mathcal{B}$ from left to right into *quanta* $Q_1, \ldots, Q_q$, each containing nodes with probability summing up to $\frac{1}{\beta t}$. As in Section 3.1, we allow the subdivision of nodes to obtain these quanta. Observe that quanta are defined in such a way that the expected number of azure nodes in each of them is 1.

From Chernoff's bounds we get that for each σ there exist sufficiently large values of a, c such that for $t \geq c\log n$ whp all sets of $(1+\sigma)(1-\delta)\beta\, a\log n$ consecutive quanta contain at least $(1-\delta)\beta\, a\log n$ azure nodes and no set of $(1-\sigma)(1-\delta)\beta\, a\log n$ consecutive quanta contains more than $(1-\delta)\beta\, a\log n$ azure nodes. Hence, whp, every group V_i consists of at least $(1-\sigma)(1-\delta)\beta\, a\log n$ quanta and of at most $(1+\sigma)(1-\delta)\beta\, a\log n$ quanta. In expectation we get $\frac{1-\beta}{\beta}$ orange nodes in each quantum. Again from Chernoff's bounds we obtain that each group contains at least $(1-2\sigma)(1-\delta)(1-\beta)\, a\log n$ and no more than $(1+2\sigma)(1-\delta)(1-\beta)\, a\log n$ orange nodes whp.

Theorem 6 *For every $\epsilon > 0$ there is a choice of $\delta, \beta, \sigma \in [0,1]$ and $a, c > 0$ such that, for $k \geq c\log n$, Algorithm* `outost-large-unk` *connects at least $(1-\epsilon)k$ nodes whp. The expected cost of the solution is $O(\log^2 n)$ times the expected cost of the optimum offline solution.*

Proof We next assume that δ, β and σ are sufficiently small, and that a and c are sufficiently large. There are $(1-\delta)\beta k$ blue terminals and each group contains exactly $(1-\delta)a\beta \log n$ blue terminals, so the number of blue groups is $\frac{(1-\delta)\beta k}{a(1-\delta)\beta \log n} - 2 = \frac{k}{a \log n} - 2$. By Lemma 11, the numer of red nodes is whp at least

$$(1-2\sigma)(1-\delta)(1-\beta)\, a \log n \left(\frac{k}{a \log n} - 2 \right) \geq (1-2\sigma)(1-\delta)(1-\beta)(1-\frac{2a}{c})k \geq (1-\epsilon)k.$$

When the number of orange terminals in each group is less than $(1+2\sigma)(1-\delta)(1-\beta)\, a \log n = O(\log n)$, we can bound the cost of the algorithm as in Lemma 5 by $O(\log^2 n)$ times the expected cost of the optimum solution. Otherwise, we can bound the worst case cost of the solution by $\Delta_r t e_{k,t}$. Making the probability of the latter event less than $\frac{1}{\Delta_r n}$, we can bound the total expected cost by $O(\log^2 n)$ times the expected cost of the optimal solution.

6 Acknowledgments

We would like to thank the anonymous reviewers of both this as and the conference version, for their constructive comments.

References

1. Anagnostopoulos, A., Grandoni, F., Leonardi, S., Sankowski, P.: Online network design with outliers. In: ICALP, pp. 114–126 (2010)
2. Azar, P.D., Kleinberg, R., Weinberg, S.M.: Prophet inequalities with limited information. In: SODA, pp. 1358–1377 (2014)
3. Babaioff, M., Immorlica, N., Kempe, D., Kleinberg, R.: A knapsack secretary problem with applications. In: APPROX '07/RANDOM '07, pp. 16–28 (2007)
4. Babaioff, M., Immorlica, N., Kempe, D., Kleinberg, R.: Online auctions and generalized secretary problems. SIGecom Exchanges **7**(2), 1–11 (2008). DOI http://doi.acm.org/10.1145/1399589.1399596
5. Babaioff, M., Immorlica, N., Kleinberg, R.: Matroids, secretary problems, and online mechanisms. In: SODA, pp. 434–443 (2007)
6. Bartal, Y.: On approximating arbitrary metrics by tree metrics. In: STOC, pp. 161–168 (1998)
7. Borodin, A., El-Yaniv, R.: Online computation and competitive analysis. Cambridge University Press, New York, NY, USA (1998)
8. Bruce, F.T., Ferguson, T.S.: Minimizing the expected rank with full information. Journal of Applied Probability **30**(3), 616–626 (1993)
9. Charikar, M., Khuller, S., Mount, D.M., Narasimhan, G.: Algorithms for facility location problems with outliers. In: SODA '01, pp. 642–651 (2001)
10. Dynkin, E.B.: The optimum choice of the instant for stopping a markov process. Sov. Math. Dokl. **4** (1963)
11. Fakcharoenphol, J., Rao, S., Talwar, K.: A tight bound on approximating arbitrary metrics by tree metrics. Journal of Computer and System Sciences **69**(4), 485–497 (2004). URL http://dx.doi.org/10.1016/j.jcss.2004.04.011
12. Fiat, A., Woeginger, G.J. (eds.): Online algorithms, *Lecture Notes in Computer Science*, vol. 1442. Springer-Verlag, Berlin (1998). The state of the art, Papers from the Workshop on the Competitive Analysis of On-line Algorithms held in Schloss Dagstuhl, June 1996
13. Freeman, P.: The secretary problem and its extensions: a review. International Statistical Review **51**(2), 189–206 (1983)
14. Garg, N.: Saving an epsilon: a 2-approximation for the k-MST problem in graphs. In: STOC, pp. 396–402 (2005)

15. Garg, N., Gupta, A., Leonardi, S., Sankowski, P.: Stochastic analyses for online combinatorial optimization problems. In: SODA, pp. 942–951 (2008)
16. Gilbert, J.P., Mosteller, F.: Recognizing the maximum of a sequence. Journal of the American Statistical Association **61**(313), 35–73 (1966). DOI 10.2307/2283044. URL http://dx.doi.org/10.2307/2283044
17. Hajiaghayi, M.T., Kleinberg, R., Parkes, D.C.: Adaptive limited-supply online auctions. In: EC, pp. 71–80 (2004)
18. Hill, T.P., Kertz, R.P.: A survey of prophet inequalities in optimal stopping theory. Contemporary Mathematics **125**, 191–207 (1992)
19. Imase, M., Waxman, B.M.: Dynamic Steiner tree problem. SIAM Journal on Discrete Mathematics **4**(3), 369–384 (1991)
20. Irani, S., Karlin, A.R.: On online computation. In: D. Hochbaum (ed.) Approximation Algorithms for NP Hard Problems. PWS publishing Co (1996)
21. Karlin, A.R., Phillips, S.J., Raghavan, P.: Markov paging. SIAM Journal on Computing **30**(3), 906–922 (2000)
22. Karlin, S.: Stochastic models and optimal policy for selling an asset. Studies in applied probability and management science pp. 148–158 (1962)
23. Kennedy, D.: Prophet-type inequalities for multichoice optimal stopping. Stochastic Processes and their Applications **24**(1), 77–88 (1987)
24. Kleinberg, R.: A multiple-choice secretary algorithm with applications to online auctions. In: SODA, pp. 630–631 (2005)
25. Kleywegt, A.J., Papastavrou, J.D.: The dynamic and stochastic knapsack problem. Operations Research **46**(1), 17–35 (1998)
26. Korula, N., Pal, M.: Algorithms for secretary problems on graphs and hypergraphs. CoRR **abs/0807.1139** (2008)
27. Koutsoupias, E., Papadimitriou, C.H.: Beyond competitive analysis. SIAM Journal on Computing **30**(1), 300–317 (2000)
28. Lindley, D.V.: Dynamic programming and decision theory. Applied Statistics **10**, 39–51 (1961)
29. Meyerson, A.: Online facility location. In: FOCS, pp. 426–431 (2001)
30. Puterman, M.L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons, Inc., New York, NY (1994)
31. Raghavan, P.: A statistical adversary for on-line algorithms. In: Online Algorithms, *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, vol. 53, pp. 79–83 (1991)
32. Rosenkrantz, D.J., Stearns, R.E., Lewis II, P.M.: An analysis of several heuristics for the traveling salesman problem. SIAM Journal on Computing **6**(3), 563–581 (1977)
33. Sleator, D.D., Tarjan, R.E.: Amortized efficiency of list update and paging rules. Communications of the ACM **28**(2), 202–208 (1985)
34. Swamy, C., Shmoys, D.B.: Approximation algorithms for 2-stage stochastic optimization problems. In: FSTTCS, pp. 5–19 (2006)
35. Young, N.E.: On-line paging against adversarially biased random inputs. Journal of Algorithms **37**(1), 218–235 (2000)