A Simple and Deterministic Competitive Algorithm for Online Facility Location

Aris Anagnostopoulos, Russell Bent,
Eli Upfal, and Pascal Van Hentenryck

Department of Computer Science
Brown University
Providence, Rhode Island 02912

CS-03-16
September 2003

A Simple and Deterministic Competitive Algorithm for Online Facility Location

Aris Anagnostopoulos, Russell Bent, Eli Upfal, and Pascal Van Hentenryck

Department of Computer Science
Brown University
Box 1910
Providence, RI 02912

Abstract

This paper presents a deterministic and efficient algorithm for online facility location. The algorithm is based on a simple hierarchical partitioning and is extremely simple to implement. It also applies to a variety of models, i.e., models where the facilities can be placed anywhere in the region, or only at customer sites, or only at fixed locations. The paper shows that the algorithm is $O(\log n)$-competitive under these various models. It also shows that the algorithm is $O(1)$-competitive with high probability and for any arrival order when customers are uniformly distributed or when they follow a distribution satisfying a smoothness property. Experimental results for a variety of scenarios indicate that the algorithm behaves extremely well in practice.

1 Introduction

Online facility location problems arise in a variety of telecommunication, networking, and mobile computing applications. They consist of choosing when and where to open facilities in order to minimize the associated cost of opening a facility and the (transportation) cost of servicing customers. The offline version of the problem is a well-known and well-studied combinatorial optimization for which effective mathematical programming, local search, and approximation algorithms are known. The online version, however, has received much less attention. Meyerson [15] presented the first randomized online algorithm for facility location and proves that it was $O(\log n)$-competitive in the worst-case and $O(1)$-competitive when customers arrive in random order.[1] Very recently, Fotakis [7] presented the first deterministic online algorithm which achieves the optimal competitive ratio of $O(\frac{\log n}{\log \log n})$. Unfortunately, the resulting algorithm is hard to implement and very demanding computationally.

This paper presents a simple and deterministic competitive algorithm for online facility location. The algorithm, whose key idea is a hierarchical partition of the region of interest, is $O(\log n)$-competitive and runs in $O(n \log n)$ time in the worst case. The algorithm, developed independently of [7], is very simple to implement, and applies to a variety of models. Despite its simplicity, the algorithm behaves

[1] We use $\log n$ to denote the base-2 logarithm of n in this paper.

very well in practice under a variety of models and distributions. More precisely, the main contributions of this paper are as follows:

- The paper presents a simple and deterministic $O(\log n)$-competitive algorithm for online facility location. The algorithm applies to a variety of models (the region model, Meyerson's model, and the fixed location model). It is extremely simple to implement and is much more efficient than the other deterministic algorithm developed independently in [7]. It is also the first competitive algorithm for the fixed location model.

- The paper presents the first probabilistic analysis of an online facility location algorithm. The analysis shows that our algorithm is $O(1)$-competitive for any arrival order when customers are uniformly distributed. It also shows that our algorithm remains $O(1)$-competitive for any arrival order as long as the distribution satisfies a smoothness property.

- The paper presents the first experimental results comparing the various algorithms under a variety of hypotheses. They show that our algorithm compares well, and almost always outperforms, other competitive algorithms. It can also bring some significant benefits compared to Meyerson's randomized algorithm.

The rest of this paper is organized as follows. Section 2 specifies the problem and describes related work. Section 3 presents the novel competitive algorithm for the region model and proves the $O(\log n)$-competitive ratio. Sections 4 and 5 generalize the result to Meyerson's model and to the fixed location model. Section 6 shows that the quality of the online algorithm is independent (asymptotically) of the arrival order of the customer. Section 7 describes the probabilistic analysis of the algorithm. Section 8 reports the experimental results and Section 9 concludes the paper and describes future work.

2 Problem Description and Prior Work

This section describes the online facility problems and gives an overview of prior work.

Problem Description This paper considers a class of online facility location problems, whose corresponding offline problem is essentially uncapacitated facility location. More precisely, given a set of facility and customer locations, the objective is to minimize the fixed costs of opening facilities and travel costs to serve customers by choosing which facility to open. The fixed cost of opening a facility is f, while the travel cost from a facility ℓ to customer c is given by the metric distance between ℓ and c and denoted by $t_{\ell,c}$. Hence the objective function can be specified as

$$f|Open| + \sum_{c \in Customers} \min_{\ell \in Open} t_{\ell,c}$$

where *Open* is the set of open facilities.

In the online problem, the customer locations are not known a priori but are revealed over time. The goal is thus to decide dynamically when and where to open facilities. Several online models are studied in this paper. In the region model, the facilities can be placed anywhere in a region. In Meyerson's model, the facilities can only be placed at existing customer locations. In the fixed location model, the facility locations are given a priori and the objective is to decide whether, when, and where to open a given a facility.

Related Work Most of the work on facility location is concerned with the offline case, where the locations of all the customers are known in advance. See, for instance, [4, 9, 10, 13, 19, 20] for a variety of approximation algorithms. See also references [5, 6, 8, 12, 16] for some interesting mathematical programming and local search algorithms.

The online uncapacitated facility location problem was first studied by Meyerson in [15]. He presents a randomized algorithm with an $O(\log n)$ *expected* competitive ratio for the model where the facilities are placed at customer sites. The algorithm is simple and elegant; when a new customer arrives, a facility is opened at the new customer site with probability proportional to the distance between the location of the new customer and the closest opened facility. In addition, Meyerson shows that, whenever the location of the incoming points is adversarial but the arrival order is random, the *expected* competitive ratio of the algorithm is constant. Fotakis [7] continued the study of the problem and showed that no randomized algorithm can achieve a competitive ratio better than $\Omega\left(\frac{\log n}{\log\log n}\right)$ against an oblivious adversary, even if the metric space is a line segment. He also presented a deterministic algorithm for any metric space that achieves the optimal competitive ratio of $O\left(\frac{\log n}{\log\log n}\right)$. At the conceptual level, the algorithm can be thought of as a derandomized version of Meyerson's algorithm. Fotakis' results, which are very elegant technically, represent a fundamental theoretical advance. However, from a practical standpoint, his algorithm appears very difficult to implement efficiently and to apply in practice. For each arriving customer, the algorithm first finds the nearest facility, say at a distance d, in a manner similar to the Meyerson algorithm. It then defines a cluster of "unsatisfied" customers that are within a radius d/x (for some $x \geq 10$) of the new customer. The "potential" of the cluster is defined by the travel cost of the customers to their nearest facility. If the potential is greater than the cost of opening a facility, then a new facility is opened within this cluster and the cluster is removed from the set of possible unsatisfied customers. The location of the new facility is chosen very carefully. If the distance d is greater than the cost f of opening a facility, the facility is created at the location of the new customer. Otherwise, the algorithm systematically creates subclusters around each customer in the cluster. Initially, the radius of a subcluster is the radius of the entire cluster

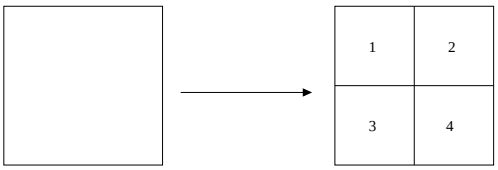

Figure 1: Partitioning a Square

and it is repeatedly halved until there exists one subcluster whose potential is half of the total potential or until the potentials of all the subclusters contain less than half of the total potential. In the former case, the new facility is opened at the center of the pertinent subcluster while, in the latter case, the location of an arbitrary customer is chosen as the location of the new facility.

The time complexity for Meyerson's algorithm is at least $\Omega(\log n)$ when the n-th customer arrives, even for a Euclidean space, since the algorithm must find the location of the nearest facility. The time complexity of Fotakis' algorithm is not addressed in [7]; however, a crude analysis indicates that it may take up to $O(n^2 + \log d)$ time to process the n-th customer, where d is the maximum distance. It may be possible to improve this bound by using advanced data structures to perform some of the queries. For the Euclidean space, see, for instance, [2] and, for a general metric space, refer to the survey in [3]. (A nice discussion also appears in [11]). Nevertheless, all these data structures—especially their dynamic versions—are quite sophisticated and complicated, and it seems unlikely that his algorithm could approach the simplicity, practical efficiency, and time-complexity bound of the algorithm presented here. This is a significant drawback of Fotakis' algorithm because of the online nature of the problem.

Finally, the work of Mettu and Plaxton [14] on the online median problem is also related to online facility location. Here the location of the customers is known in advance and the number of facilities increases in an online fashion.

3 The Region Model

We present the algorithm for the region model which assumes that the facilities can be located anywhere in a given region at a cost f. To ease the presentation, we first specify and analyze the algorithm for the case where the region is a square with diagonal of length f. We then show how to generalize it to arbitrary regions.

3.1 The Online Algorithm

The key idea behind the online algorithm is to partition the initial square into smaller and smaller squares as customers arrive. More precisely, its basic operation, depicted in Figure 1, consists of partioning a square into four squares of the same size, called quadrants. Once a quadrant q is created and not (yet) partitioned,

the online algorithm keeps track of the customers arriving in q. These customers are called *support customers* of quadrant q in the rest of this paper and the travel cost of these support customers (to a facility to be specified) is called the *support cost* of q. Once the support cost of quadrant q exceeds a threshold, the algorithm opens a facility in q and partitions q. In the following, we use *support(q)* to denote the support customers of q and *cost(q)* to denote its support cost. We also use *facility(q)* to denote the facility associated with a quadrant q when q is partitioned. A quadrant q is *open* if it has an associated facility (and thus is partitioned); it is *recruiting* otherwise.

It is also important to introduce a few additional concepts before presenting the algorithm. If a quadrant q is partitioned into squares q_i $(1 \leq i \leq 4)$, quadrant q is said to be the parent of q_i $(1 \leq i \leq 4)$. All quadrants have a parent, except the root square. The ancestors of a quadrant q are its parent p and the ancestors of p. A *corner ancestor* of quadrant q is an ancestor whose center lies on a corner of q. It can be shown that the partitioning into quadrants guarantees that a quadrant q has at most two corner ancestors, one of which being of course its parent. Moreover, if two exist, they must lie on diagonally opposite corners of q. The *local facilities* of a quadrant are simply the facilities associated with its corner ancestors and we use *local(q)* to denote them. Local facilities and corner ancestors are important concepts for some of the competitive ratios presented later in the paper.

We are now ready to present algorithm $\mathcal{A}$ which is depicted in Figure 2. Algorithm $\mathcal{A}$ first initializes the root quadrant by partitioning it (procedure PARTITION) and then serves the customers as they come (procedure SERVECUSTOMER). Procedure PARTITION partitions a square q by selecting a facility location (procedure SELECTLOCATION) and by constructing its four subquadrants. PARTITION also replaces q by the four new quadrants in the set *Quadrants*, which contains all the recruiting quadrants. For each incoming customer c, procedure SERVECUSTOMER inserts c in a recruiting quadrant (procedure ADDTOQUADRANT) and assigns c to its closest facility (procedure ASSIGNCLOSESTFACILITY). To add a customer c to a recruiting quadrant, procedure ADDTOQUADRANT first locates the recruiting quadrant q containing c (procedure FINDQUADRANT in line 1) and adds c to the support of q (line 2). It computes the distance from c to the closest local facility of q (line 3), and updates the support cost accordingly (line 4). If the support cost exceeds the threshold af (line 5), where a is a parameter of the implementation, the quadrant q is partitioned and becomes open (line 5). In the region model, procedure SELECTLOCATION simply chooses the center of the quadrant to locate the facility. (This implementation is reconsidered in other models).

Observe that the cost of a quadrant q is the cost of its associated facility (if any) and the support cost of its support customers $cost(q)$. Since all the facilities and all the customers are associated with quadrants, the cost of the solution is the sum of the costs of all quadrants.

It is important to emphasize that the support cost of a customer, i.e., the travel

```
A(Square r)
1  INIT(r);
2  for each arriving customer c
3  do SERVECUSTOMER(c);

INIT(Square q)
1  parent(q) ← {};
2  Quadrants ← {};
3  PARTITION(q);

PARTITION(Square q)
1  facility(q) ← SELECTLOCATION(q);
2  quadrants(q) ←  4 Quadrants of q ;
3  for r ∈  quadrants(q)
4  do cost(r) ← 0;
5       support(r) ← {};
6       parent(r) ← {q};
7  Quadrants ← Quadrants ∪ quadrants(q) \ {q};

SERVECUSTOMER(Customer c)
1  ADDTOQUADRANT(c);
2  ASSIGNCLOSESTFACILITY(c);

ADDTOQUADRANT(Customer c)
1  q ← FINDQUADRANT(Quadrants, c);
2  support(q) ← support(q) ∪ {c};
3  tc ← min(ℓ in local(q)) t_{ℓ,c};
4  cost(q) ← cost(q) + tc;
5  if cost(q) > af
6     then PARTITION(q);
```

Figure 2: Algorithm $\mathcal{A}$

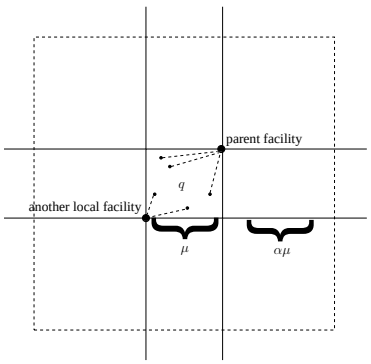

Figure 3: Visualization of Optimal Facility Placement

cost to its closest local facility, is greater or equal to the actual travel cost to its closest location. The use of the support cost in thresholding simplifies the analysis and makes it possible to prove robustness results with respect to the customer ordering. Note also that most (but not all) results in this paper hold when line 3 in procedure ADDTOQUADRANT is replaced by $tc \leftarrow t_{parent(q),c}$.

3.2 Worst-Case Competitive Analysis

This section analyzes the worst case performance of the algorithm assuming that the maximum length (i.e., the diagonal) of the square is f, where f is the cost of opening a facility. This assumption is relaxed in Section 3.3. The first lemma bounds the maximum depth of a partition produced by Algorithm $\mathcal{A}$. It does not depend on the location of the facilities inside the quadrants.

Lemma 1. *The maximum partition depth produced by Algorithm $\mathcal{A}$ when serving n customers is $O(\log n)$.*

Proof. Observe that there must be at least $a2^{i-1}$ customers in the support of a quadrant at depth i in order to partition it, since the maximum distance in a quadrant at depth i is not greater than $\frac{f}{2^{i-1}}$. Hence, the maximum depth for n customers is $\log n - \log a$, which is $O(\log n)$. □

The second lemma is central to the rest of the paper and will be adapted to other models subsequently. Informally speaking, it specifies that every quadrant created by algorithm $\mathcal{A}$ is close to a facility in an optimal solution, the distance depending on the size of the quadrant.

Lemma 2. *Let q be an open quadrant with sides of length μ produced by algorithm $\mathcal{A}$. In an optimal solution, there must exist at least one facility within distance $\alpha\mu$ of q, where $\alpha = \frac{(a+2)}{a}$.*

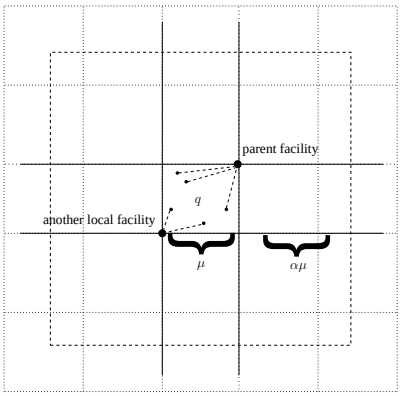

Figure 4: Illustrating Corollary 1. There must exist an optimal facility in the square whose sides have length $(2\alpha + 1)\mu$.

Proof. The proof is by contradiction. Assume that there exists no facility within distance $\alpha\mu$ of an open quadrant q in an optimal solution O. Since q is open, the cost s of its support is greater than af. Moreover, there are at least $\frac{s}{\mu}$ support customers in q, since the maximum distance to a local facility is μ (see Figure 3). Now the cost to connect one of these support customers to a facility in O is at least $\alpha\mu$ by assumption. It follows that the travel cost of the support customers in the optimal solution is given by

$$\frac{s}{\mu} \cdot \frac{(a+2)}{a}\mu = \frac{a+2}{a}s$$

which is greater than

$$s + 2f$$

whenever $s > af$. As a consequence, we could improve O by opening the local facilities of q, which contradicts the fact that O is optimal. □

The following corollary, which is illustrated graphically in Figure 4, is a direct consequence of Lemma 2.

Corollary 1. *Let q be an open quadrant with sides of length μ produced by algorithm $\mathcal{A}$. An optimal solution must have at least one facility in a square of size $(2\alpha + 1)\mu$ by $(2\alpha + 1)\mu$ whose center is q, where $\alpha = \frac{(a+2)}{a}$.*

As mentioned, the lemma and corollary will be adapted slightly for the other models by changing the value of α to take account higher travel costs in the other model but the proof will remain similar. We are now in position to present the proof of the competitive ratio.

Theorem 1. *Algorithm $\mathcal{A}$ is $O(\log n)$-competitive for a square whose diagonal has length f.*

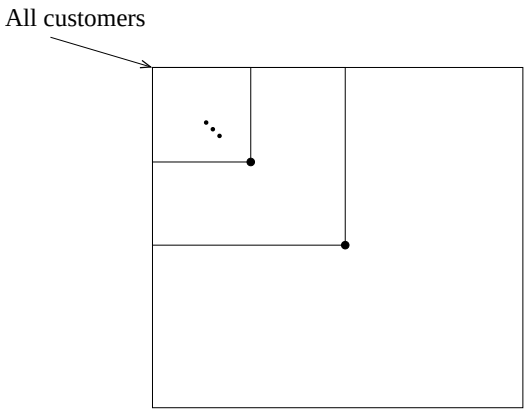

Figure 5: Visualization of a Worst Case Instance for the Lower Bound.

Proof. Let ℓ be an optimal facility and consider a depth d. Consider all quadrants at depth d which have some customers allocated to ℓ. By Corollary 1, there are at most $(\lceil 2\alpha \rceil + 1)^2$ such quadrants. Moreover, the cost of a quadrant is at most $2af$ (travel cost) plus f (facility cost) if it is open and at most af otherwise. Hence the total cost of the quadrants at depth d is thus at most $(\lceil 2\alpha \rceil + 1)^2 \cdot 2af$. By Lemma 1, there are $O(\log n)$ such depths d and the total cost of all quadrants with customers allocated to ℓ in the optimal solution is thus $O(f \times \log n)$. The result follows since the above reasoning can be applied to all facilities in the optimal solution. □

The above competitive ratio is tight as shown by the instance depicted in Figure 5. In this instance, all customers are placed in the corner of the region. The optimal solution opens a facility in the corner and the total cost of the optimal solution will be f. Algorithm $\mathcal{A}$ produces $\log n$ quadrants. The result follows, since the cost of each such quadrant is at most $O(f)$ (as discussed in the proof of Theorem 1).

3.3 Arbitrary Regions

Algorithm $\mathcal{A}$ naturally generalizes to arbitrary regions. The result holds if the region can be enclosed by the square whose diagonal is of length f, since no assumptions on the customer locations were used.[2] Consider now the case where the region is enclosed by a square whose diagonal is greater than f. The key idea behind the generalized algorithm $\mathcal{A}'$ is to cover this square by non-overlapping squares whose diagonals are of length f. No facilities are created in these squares initially. When a customer arrives in one of these squares, algorithm $\mathcal{A}$ is applied to that particular square. More precisely, when the first customer in such a square s arrives, the data structures for s are initialized using procedure INIT and SERVECUSTOMER. Only

[2]The result also holds when the facilities must be placed inside the initial region, as will become clear when other models will be presented.

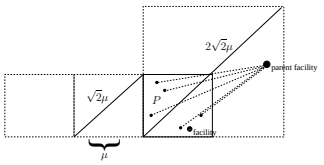

Figure 6: Visualization of Optimal Facility Placement

procedure serveCustomer is called for subsequent customers in s. Observe that Lemmas 1 and 2 still hold for all the created quadrants. In addition, Lemma 2 can also be adapted to apply to the squares used in the partition covering the initial region and the result follows.

Theorem 2. *Algorithm $\mathcal{A}'$ is $O(\log n)$-competitive for online facility location with no restriction on the locations of the facilities.*

Note that it is not necessary for the region to be known ahead of time. When the first customer arrives, a square with maximum length f can be placed around the customer. If subsequent customers fall outside that square, then squares of maximum length f can be created around the initial square until the customers are covered. In the rest of the paper, we assume for simplicity that the region is a square whose diagonal has length f.

4 Meyerson's Model

We now show that algorithm $\mathcal{A}$ naturally generalizes to Meyerson's model where facilities are only opened at customer locations. The only modification in the algorithm is in procedure selectLocation(q) which now places a facility at the support customer of q which is closest to its center (instead of at the center of q).[3]

Under this model, Lemma 1 still holds, since its proof only relies on the sizes of the quadrant. Lemma 2 also holds under the new model by making $\alpha = \frac{2(a+2)\sqrt{2}}{a}$, since the maximum distance of a support customer to a local facility of a quadrant q is $2\sqrt{2}\mu$, since the parent facility of q can be anywhere in its own quadrant (See Figure 6). As a consequence, algorithm $\mathcal{A}$ is also $O(\log n)$-competitive under this model.

Theorem 3. *Algorithm $\mathcal{A}$ is $O(\log n)$-competitive for online facility location when facilities must be located at customer sites.*

Note that the result also holds if only the parent facility is used, since the proof of (the equivalent of) Lemma 2 only relies on the parent facility. This makes it possible to reduce α to $\frac{2\sqrt{2}(a+1)}{a}\mu$, since only one facility is used in the proof.

[3]The facility can be placed anywhere in the quadrant, a property used in the experimental results.

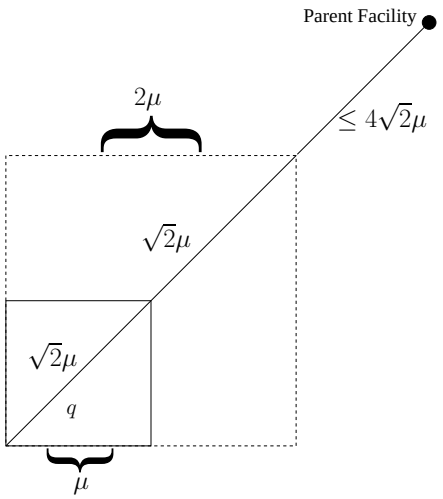

Figure 7: Opening Facilities in the Fixed Location Model.

5 The Fixed Facility Location Model

We now consider the fixed facility location model, where facilities can only be located at a fixed set of locations. Once again, we assume that the region is a square whose diagonal is of size f for simplicity. Clearly, the $O(\log n)$ lower bound still applies, since the fixed locations can be precisely located at the center of the quadrant as in Figure 5. We now prove that algorithm $\mathcal{A}$ can be naturally adapted to remain $O(\log n)$-competitive under this model.

The main change in the algorithm is to restrict further when a facility can be opened for a quadrant. More precisely, a facility can be opened for a quadrant q with sides of length μ when the cost of its support customers exceeds af and when there exists a facility (inside or outside the quadrant) within distance $2\sqrt{2}\mu$ of q. Note that this facility may be opened already, implying that the same facility may be associated with several quadrants. This amounts to replacing line 5 of procedure serveCustomer by

if $cost(q) > af$ & $existsFacility(q)$

where $existsFacility(q)$ returns true if there exists a facility location within distance $2\sqrt{2}\mu$ of q. Procedure selectLocation(q) is also updated slightly in order to choose the location closest to q within distance $2\sqrt{2}\mu$.

We now show that the modified algorithm is $O(\log n)$-competitive. The key idea behind the proof is to separate the quadrants in two sets: the *traditional* quadrants which are open or have a cost lower than af and the *isolated* quadrants which have a cost greater than af but no facility within distance $2\sqrt{2}\mu$. The traditional quadrants can be analyzed in a way similar to earlier proofs. Moreover, the travel costs of the

isolated quadrants is shown to be within a constant factor of their travel costs in the optimal solution. Figure 7 depicts the intuition behind the proof visually.

Theorem 4. *Algorithm $\mathcal{A}$ is $O(\log n)$-competitive for online facility location with fixed locations for facilities.*

Proof. Let σ be the solution produced by algorithm $\mathcal{A}$. Consider first an isolated quadrant q with sides of length μ in σ. Its parent p has sides of length 2μ and has a facility at distance $4\sqrt{2}\mu$, since it has been partitioned. Consequently, each customer in the support of q has a travel cost of at most $6\sqrt{2}\mu$. Moreover, since there are no facilities within distance $2\sqrt{2}\mu$ of q, each such support customer must pay a cost of at least $2\sqrt{2}\mu$ in the optimal solution. Hence the travel cost of the support customers of q in σ is at most 3 times their travel cost in the optimal solution.

Consider now a traditional quadrant q with sides of length μ in σ. Its parent has a facility within distance $4\sqrt{2}\mu$ which means that each support customer has a travel cost of at most $6\sqrt{2}\mu$ again. We can thus prove a result similar to Lemma 2 by choosing $\alpha = \frac{6\sqrt{2}(a+2)}{a}$. Moreover, the maximum distance to a facility for a quadrant q at depth i is $4\frac{f}{2^{i-1}}$. Hence the maximum partition depth is $O(\log n)$ when serving n customers, providing the counterpart to Lemma 1. The rest of the proof for the traditional quadrants can then proceed as in Theorem 1 and the result follows. □

6 Robustness of the Algorithm

We now show that algorithm $\mathcal{A}$ is robust with respect to the order in which customers arrive. More precisely, we show that the cost of algorithm $\mathcal{A}$ for any customer ordering is only a constant factor worse than the cost for the best customer ordering. This result indicates that algorithm $\mathcal{A}$ depends essentially on the customer locations, not their arrival order. It makes it possible to analyze the performance of algorithm $\mathcal{A}$ by assuming that the customers arrive in random order. This property is used in Section 7 where the performance of the algorithm is analyzed under a uniform distribution of the customers. We show the result for the basic region model.

Theorem 5 (Worst Case Ordering). *Let $c_1, \ldots, c_n$ be n customer locations and let Σ be the set of sequences of these n locations. Let*

$$\sigma^* = \arg\min_{\sigma \in \Sigma} cost(\sigma)$$

where $cost(\sigma)$ is the cost of the solution produced by Algorithm $\mathcal{A}$ on the sequence σ. Then, there exists a constant $\kappa \geq 1$ such that

$$\forall \sigma \in \Sigma : \ cost(\sigma) \leq \kappa \ cost(\sigma^*).$$

Proof. Let q be a quadrant which is not partitioned in the best-case ordering o^* and which is partitioned in the worst-case ordering o. Since the same customers

belong to q in both orderings, this happens when some of the customers are in the support of the ancestors of q in o^*, but not in o. Observe that the travel cost of any such *unaccounted* customer in ordering o is not greater than its travel cost in ordering o^*, since its local facilities include its closest ancestor. Clearly, the set of unaccounted customers is not greater than the set of all customers whose travel cost is bounded by $2afw$, where w is the number of facilities for ordering o^*. As a consequence, since ordering o requires at least af in travel cost to open a facility, the unaccounted customers can only open $2w$ facilities and the result follows. □

Note that this result can be generalized to the fixed location model provided that the choice of the facility location for a quadrant be deterministic (e.g., ties are broken deterministically in procedure SELECTLOCATION). Such a choice guarantees that the travel cost of the unaccounted customers is not greater in the worst-case ordering than in the best ordering. The result does not generalize to Meyerson's model where facility location choices critically depend on the customer order.

7 Probabilistic Analysis

The previous sections showed that the competitive ratio of the algorithm $\mathcal{A}$ is $\Theta(\log n)$ but, in practice, algorithm $\mathcal{A}$ should behave much better. It is thus interesting to analyze algorithm $\mathcal{A}$, not under an adversarial model, but under various distributions of the customers. This section follows this approach and analyzes algorithm $\mathcal{A}$ for a uniform distribution of the customers, as well as for distributions which have smooth neighborhoods. In both cases, we show that algorithm $\mathcal{A}$ produces a solution whose cost is within a constant of the optimal offline solution with high probability (whp).[4] We start by proving the result for the uniform distribution. The result is then generalized to distributions with smooth neighborhoods.

7.1 Uniform Distribution of Customers

The intuition behind the proof for the uniform distribution is the following. First recall that the partitioning threshold guarantees that the travel cost is proportional to the cost of the facilities, so that the proof can focus on the number of open facilities or, equivalently, on the number of quadrants. Moreover, by Lemma 2, algorithm $\mathcal{A}$ opens at most a constant times more facilities than the optimal solution at a given depth d. Hence, to avoid the logarithmic factor in the ratio, it suffices that most quadrants up to a depth $\overline{d}$ are partitioned/opened and that quadrants deeper than $\overline{d}$ can only increase the cost by a constant factor. More precisely, the quadrants define a tree where nodes have at most 4 children and the proof shows that the tree grows exponentially up to depth $\overline{d}$ (i.e., each node has at least 3 children whp), while the quadrants deeper than $\overline{d}$ only increase the cost by a constant factor. Besides

[4]Recall that an event E holds whp if there is a constant $c > 0$ such that $\mathbf{Pr}(E) > 1 - \frac{1}{n^c}$.

various Chernoff and Markov bounds, the proof relies on two main results. The first main result is Theorem 5 which makes it possible to assume that the customers arrive in random order. In other words, the proof assumes that the location of a new customer is uniformly distributed in the whole region and independent of all previous and subsequent customer locations. The second result is the following theorem expressed in terms of balls and bins [1, 17].

Theorem 6. *Consider two balls and bins experiments. In the first one, m balls are thrown to n bins, independently and uniformly at random. In the second experiment, each of the n bins accepts a number of balls distributed according to the Poisson distribution with mean m/n, all the bins being mutually independent. Consider an event E whose probability is increasing with the number of balls thrown. Then the probability of E in the first experiment is at most 4 times higher than the probability of E in the second experiment.*

This theorem makes it possible to apply a Chernoff bound on the quadrants by removing the dependencies between the quadrants, i.e., the fact that a customer only lies in one quadrant. In the proof, bins are half quadrants (to be specified) and balls are customers. We are now ready to present and prove the main result of this section.

Theorem 7. *Consider a square region and assume that the customers are uniformly distributed in the region and mutually independent. Then the cost of the solution produced by algorithm $\mathcal{A}$ is only a constant times higher than that of the optimal offline algorithm, with probability at least $1 - 1/n$.*

Proof. For simplicity, we assume that the region is a square with sides of length 1 ($f = \sqrt{2}$) and that the parameter $a = 1/\sqrt{2}$ so that the threshold is 1. Let $d^- = \log \ln n$ and $\overline{d} = (\log n)/3 - 1$. The proof is divided in three main steps:

1. It shows that all 4^i quadrants are opened whp up to depth d^-.
2. It shows 3^i quadrants are opened whp at each depth $i \leq \overline{d}$.
3. It shows that no quadrants are opened whp at depth $\overline{d} + 2$.

The first step is in fact necessary to obtain the second step. More precisely, the n customers are divided into groups of size m_i (to be defined later), each group m_i being allocated to a depth i ($i = 1, 2, \ldots, d^-, d^- + 1, \ldots, \overline{d}$). The proof first shows that the first m_1 customers suffice to create all 4 quadrants at level 1 whp, the next m_2 customers create all 16 quadrants at level 2 whp, and so on until level d^-. Then, for levels $i = d^- + 1$ to $\overline{d}$, the proof shows that the corresponding m_i customers create at least 3^i quadrants whp. Therefore, at level $\overline{d}$, there are $3^{\overline{d}} \simeq n^{0.528}$ quadrants.[5] Finally, the proof shows that no quadrants are partitioned at

[5]This argument gives a lower bound on the number of quadrants per level. There are more quadrants in practice and the tree is thus bushier.

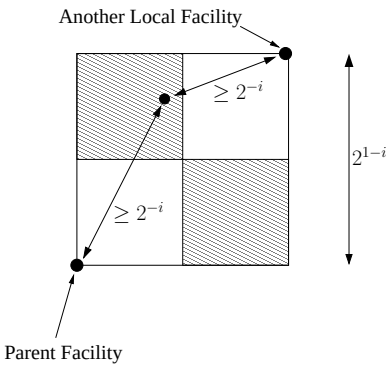

Figure 8: A partition at the i-th level

level $\overline{d}+2$ whp. This implies that if x partitions are opened at level $\overline{d}$, then the total number of facilities opened is $\Theta(x)$ which allows us to conclude by making use of Lemma 2 that the solution produced by algorithm $\mathcal{A}$ is within a constant factor of the optimal one.

Consider a quadrant at level i with side 2^{1-i} (See Figure 8). If a customer falls on the gray half, it induces a travel cost of at least 2^{-i}. As a consequence, if 2^i customers fall in this gray half, the quadrant will be partitioned. The probability for a customer to fall in the gray area equals its area and is 2^{1-2i}.

[Step 1]: Let $i \leq d^-$ and let $m_i = 4(\ln n)^3$ and consider a quadrant q. Then, by conditioning on the partitioning of all ancestors of q, the probability not to partition q is bounded by the probability that the gray area in Figure 8 receives less than 2^i customers. The expected number of the m_i customers in the gray area is

$$4(\ln n)^3 \cdot 2^{1-2i} = (\ln n)^3 \cdot 2^{3-2i}$$

and, by applying a standard Chernoff bound,[6] the probability that fewer than 2^i

[6]The standard Chernoff bound is specified as follows. Assume that the X_i are mutually independent $0-1$ random variables such that $\mathbf{Pr}(X_i = 1) = p$, and let $Y = \sum_{i=1}^{n} X_i$ and $\mu = \mathbf{E}[Y] = np$. Then, for any $\epsilon > 0$,

$$\mathbf{Pr}(Y \geq (1+\epsilon)\mu) \leq e^{-\frac{1}{3}\mu\epsilon^2},$$

and for any $\epsilon \in (0,1)$,

$$\mathbf{Pr}(Y \leq (1-\epsilon)\mu) \leq e^{-\frac{1}{2}\mu\epsilon^2}.$$

fall in the gray area is ($\epsilon = 1 - \frac{2^{3i-3}}{(\ln n)^3}$)

$$\mathbf{Pr}\left(< \left[1 - \left(1 - \frac{2^{3i-3}}{(\ln n)^3}\right)\right] \cdot (\ln n)^3 \cdot 2^{3-2i} \text{ fall in the gray area}\right)$$

$$\leq e^{-\frac{1}{2}(\ln n)^3 \cdot 2^{3-2i} \cdot \left(1 - \frac{2^{3i-3}}{(\ln n)^3}\right)^2} = e^{-(\ln n)^3 \cdot 2^{2-2i} \cdot \left(1 + \frac{2^{6i-6}}{(\ln n)^6} - \frac{2^{3i-2}}{(\ln n)^3}\right)}$$

$$\leq e^{-(\ln n)^3 \cdot 2^{2-2i} + 2^i}$$

$$\leq e^{-(\ln n)^3 \cdot 2^{2-2\log\ln n} + 2^{\log\ln n}} \qquad \text{since } i \leq d^- = \log\ln n$$

$$= e^{-4(\ln n)^3 (\ln n)^{-2} + \ln n} = e^{-3\ln n}$$

$$= \frac{1}{n^3}.$$

Since there cannot be more than n quadrants, all the 4^i quadrants ($i \leq d^-$) are partitioned with probability at least $1 - n^2$.

[Step 2]: For $i = d^- + 1, \ldots, \overline{d}$, the proof sets $m_i = 2^{3i}$ and assumes that there are at least 3^{i-1} parents are present at depth i. Hence there are at most $4 \cdot 3^{i-1}$ new quadrants and the proof shows that at least 3^i are opened whp (using the m_i customers). The situation in this step is more complicated however, since it is difficult to prove that each quadrant is opened whp without using too many customers. As a consequence, the proof needs to reason about the number of open quadrants globally. Since these quadrants are not independent (since a customer can only lie in one quadrant), the proof uses Theorem 6 to circumvent this difficulty. In applying Theorem 6, the gray and white half quadrants obtained as shown in Figure 8 play the role of bins and the customers play the role of balls. The number of balls (customers) that arrive in a bin (half quadrant) follows a Poisson distribution with parameter $m_i \cdot 2^{1-2i} = 2^{i+1}$ and is independent of that of any other bin (half quadrant). We talk about a gray (resp. white) bin to denote a bin which corresponds to a gray (resp. white) half quadrant. With this transformation at hand, the Chernoff bound for the Poisson distribution[7] implies that the probability that fewer

[7]This Chernoff bound is given as follows. Assume that the random variable X is distributed according to a Poisson distribution with parameter λ. Then (see for example [18, Pages 415-6]) for $i < \lambda$

$$\mathbf{Pr}(X < i) \leq \frac{e^{-\lambda}(e\lambda)^i}{i^i}.$$

than 2^i balls (customers) fall in a gray bin (gray half quadrant) is no more than

$$\begin{aligned}
&\frac{e^{-2^{i+1}}(e\cdot 2^{i+1})^{2^i}}{2^{i\cdot 2^i}} \\
&= e^{-2^{i+1}+2^i+(i+1)2^i\ln 2-i\cdot 2^i\ln 2} \\
&= e^{-2\cdot 2^i+2^i+2^i\ln 2} \\
&= e^{-2^i(2-1-\ln 2)} \\
&< e^{-0.3\cdot 2^i} \\
&\le e^{-0.3\cdot 2^{\log\ln n}} && \text{since } i > d^- = \log\ln n \\
&= e^{-0.3\ln n} \\
&= \frac{1}{n^{0.3}},
\end{aligned}$$

If $\tilde{p}$ denotes the probability that a gray bin (gray half-quadrant) receives at least 2^i balls (customers), we have that $\tilde{p} > 1 - 1/n^{0.3}$ and hence $\tilde{p} > 31/32$ for sufficiently large n. By considering the bins independent, the number of open quadrants then stochastically dominates a Binomially distributed random variable $Y \sim \mathit{Binomial}(4\cdot 3^{i-1}, \tilde{p})$. The expected number of gray bins (gray half-quadrants) receiving more than 2^i balls (customers) equals $4\cdot 3^{i-1}\tilde{p}$. By applying a Chernoff bound, the probability that fewer than 3^i gray bins (gray half-quadrants) receive at least 2^i balls (customers) at depth i is bounded by

$$\begin{aligned}
&4\cdot\mathbf{Pr}(<3^i \text{ gray bins get } \ge 2^i \text{ balls}) \\
&\quad\le 4\cdot\mathbf{Pr}(Y<3^i) \\
&\quad\le 4\cdot\mathbf{Pr}\left(Y<\left[1-\left(1-\frac{3}{4\tilde{p}}\right)\right]\cdot 4\cdot 3^{i-1}\tilde{p}\right) \\
&\quad\le 4\cdot e^{-\frac{1}{2}4\cdot 3^{i-1}\tilde{p}\left(1-\frac{3}{4\tilde{p}}\right)^2} \\
&\quad= 4\cdot e^{-2\cdot 3^{i-1}\tilde{p}\left(1+\frac{9}{16\tilde{p}^2}-\frac{3}{2\tilde{p}}\right)} \\
&\quad= 4\cdot e^{-\frac{2}{3}3^i\tilde{p}-\frac{3}{8\tilde{p}}3^i+3^i} \\
&\quad\le 4\cdot e^{-\frac{2}{3}3^i\frac{31}{32}-\frac{3}{8}3^i+3^i} && \text{since } 31/32<\tilde{p}\le 1 \\
&\quad= 4\cdot e^{-\frac{1}{48}3^i} \\
&\quad\le 4\cdot e^{-\frac{1}{48}3^{\log\ln n}} && \text{since } i > d^- = \log\ln n \\
&\quad= 4\cdot e^{-\frac{1}{48}(\ln n)^{\log 3}},
\end{aligned}$$

which is smaller than n^{-c_1} for any constant c_1 for sufficiently large n. Hence, by Theorem 6, 3^i quadrants are opened whp at each depth $i \le \overline{d}$.

[Step 3]: We now show that, at level $\overline{d}+2$, no quadrants are created whp. Consider a quadrant q at level i. The maximum distance (the diagonal) within q is $2^{\frac{3}{2}-i}$ (since there may not be any local facility other than the parent) and hence at least $2^{i-\frac{3}{2}}$ customers are needed to open q. Hence the probability to open quadrant q is bounded by the probability that at least $2^{i-\frac{3}{2}}$ customers fall in q. Recall also that the probability that a customer falls in the quadrant equals its area which is 2^{2-2i}. Since we are interested in an upper bound, we can assume that all n customers are available and, under this assumption, the expected number of customers to lie in q is $2^{\log n-2i+2}$. By using the Chernoff bound, the probability that more than $2^{i-\frac{3}{2}}$ customers lie in q, for $i > \log n/3 + 7/6$, is no more than

$$\begin{aligned}
&\mathbf{Pr}\left(> \left[1+\left(2^{3i-\log n-\frac{7}{2}}-1\right)\right]\cdot 2^{\log n-2i+2} \text{ customers fall in a given quadrant}\right)\\
&\leq e^{-\frac{1}{3}2^{\log n-2i+2}\left(2^{3i-\log n-\frac{7}{2}}-1\right)^2}\\
&\leq e^{-2^{\log n-2i}\left(2^{6i-2\log n-7}+1-2^{3i-\log n-\frac{5}{2}}\right)}\\
&= e^{-2^{4i-\log n-7}-2^{\log n-2i}+2^{i-\frac{5}{2}}},
\end{aligned}$$

giving us for $i = \overline{d}+2 = (\log n)/3+1$

$$\begin{aligned}
&= e^{-2^{\frac{4}{3}\log n+4-\log n-7}-2^{\log n-\frac{2}{3}\log n-2}+2^{\frac{1}{3}\log n+1-\frac{5}{2}}}\\
&= e^{-2^{\frac{1}{3}\log n}\left(\frac{1}{8}+\frac{1}{4}-\frac{1}{\sqrt{8}}\right)}\\
&\leq e^{-0.21n^{\frac{1}{3}}}.
\end{aligned}$$

The total number of potential quadrants at level i is 4^i. Hence, for $i = \overline{d}+2$, the expected number of open quadrants is bounded by

$$4^{\frac{1}{3}\log n}e^{-0.21n^{\frac{1}{3}}} = n^{\frac{2}{3}}e^{-0.21n^{\frac{1}{3}}}$$

which, asymptotically, is smaller than n^{-c_2} for any constant $c_2 > 0$. Let Z be the number of open quadrants at level $\overline{d}+2$. We have that $\mathbf{E}[Z] \leq n^{-c_2}$ and, by Markov's inequality, it follows that

$$\mathbf{Pr}(Z \geq 1) \leq \mathbf{E}[Z] \leq n^{-c_2}$$

which means that no quadrants are opened at level $\overline{d}+2$ whp.

[Synthesis]: Let us now conclude the proof by computing the failing probability, that is, the probability that the algorithm is not within a constant factor of the optimal offline solution. Let B be the event that the algorithm fails to achieve a solution within a constant factor of the optimal and let $B_1, B_2, \ldots, B_k$ be events, corresponding to all the bad events that can take place during the execution of the

algorithm, so that $B = \bigcup_{i=1}^{k} B_i$. Then, by using inductively the property that ($\bar{F}$ denotes the complement event of F)

$$\mathbf{Pr}(E) \leq \mathbf{Pr}(F) + \mathbf{Pr}(E|\bar{F}),$$

the probability of the event B is bounded by

$$\mathbf{Pr}(B) \leq \mathbf{Pr}(B_1) + \mathbf{Pr}(B_2|\bar{B}_1) + \mathbf{Pr}(B_3|\bar{B}_1, \bar{B}_2) + \cdots + \mathbf{Pr}(B_k|\bar{B}_1, \bar{B}_2, \ldots, \bar{B}_{k-1}).$$

In other words, the probability that the tree is not dense (and hence that the algorithm is not $O(1)$-competitive) is bounded by the probability that some partition at depth $1, 2, \ldots, d^-$ is not created, plus the probability that some level $i = d^- + 1, d^- + 2, \ldots, \overline{d}$ has fewer than 3^i quadrants, plus the probability that some quadrant is created in level $\overline{d} + 2$. From Steps 1 to 3, it follows that

$$\mathbf{Pr}(B) \leq \frac{1}{n^2} + (\overline{d} - d^-)\frac{1}{n^{c_1}} + \frac{1}{n^{c_2}} < \frac{1}{n},$$

and the results follows. □

7.2 Smooth Neighborhoods

We now consider a large class of distributions for which Algorithm $\mathcal{A}$ is $O(1)$-competitive whp. These distributions satisfy the smooth neighborhood property which we now define.

Definition 1 (Distribution with Smooth Neighborhoods). *Let I be the unit square I and ν be a probability distribution on I. ν has a smooth neighborhood if there exists a constant K such that*

$$\nu(Q_1) \leq K \cdot \nu(Q_2).$$

for any neighboring quadrant Q_1 and Q_2, i.e., any quadrant Q_1 and Q_2 in I satisfying $|Q_1| = |Q_2|$ and $d(Q_1, Q_2) = 0$ where

$$d(Q_1, Q_2) = \min\{|x_1 - x_2| : x_1 \in Q_1, x_2 \in Q_2\}.$$

Distributions with smooth neighborhoods have the following useful property.

Lemma 3. *Consider a quadrant q with four subquadrants q_i, $i = 1, 2, 3, 4$. For any distribution ν with smooth neighborhoods, we have*

$$\nu(q_i) \geq p \cdot \nu(q),$$

where $p = (3K + 1)^{-1}$.

Proof. Since $d(q_1, q_i) = 0$ $(i = 2, 3, 4)$, it follows that

$$\nu(q_1) \leq K \cdot \nu(q_i).$$

Hence

$$\begin{aligned}
\frac{\nu(q_1)}{\nu(q)} &= \frac{\nu(q_1)}{\nu(q_1) + \nu(q_2) + \nu(q_3) + \nu(q_4)} \\
&\geq \frac{\nu(q_1)}{\nu(q_1) + K \cdot \nu(q_1) + K \cdot \nu(q_1) + K \cdot \nu(q_1)}. \\
&= \frac{1}{3K + 1} \\
&= p
\end{aligned}$$

The proof is similar for the other three subquadrants. □

The following definitions are used in the probabilistic analysis. A *bush* is a quadrant that has more than M descendants, where

$$M = \sqrt{2}\left(\frac{1}{p} + \frac{4}{p^2} + \frac{16}{p^3} + \frac{64}{p^4} + \frac{256}{p^5}\right),$$

and p is defined as in Lemma 3. The *(i)-ancestor* of q is q if $i = 0$, the parent of q if $i = 1$, and the $(i-1)$-ancestor of the parent of q otherwise $(i > 1)$. A *bush child* is a quadrant q with an (i)-ancestor $(i = 0, \ldots, 5)$ which is a bush. A *disjoint bush* is a bush q whose (i)-ancestors $(i = 1, \ldots, 4)$ are not disjoint bushes. We also say that a bush child belongs to a bush q if q is an (i)-ancestor $(i = 0, 1, \ldots, 5)$ of the bush child. We are now ready to prove that Algorithm $\mathcal{A}$ is $O(1)$-competitive with high probability. The intuition behind the proof is as follows.

1. All quadrants up to level $d^- = 2 \log \ln n$ are partitioned whp. (Lemma 4).
2. If a quadrant q at a level $i > d^-$ has more than M descendants, then q is fully partitioned 5 levels down whp (Lemma 5).
3. Every disjoint bush q has a facility in each of its subquadrants in the optimal offline solution (Lemma 6).
4. Only disjoint bushes must be considered to show the $O(1)$-competitive ratio.

Lemma 4. *Consider the region I and a distribution ν with smooth neighborhoods. Algorithm $\mathcal{A}$ partitions all quadrants up to level $d^- = 2 \log \ln n$ with probability at least $1 - n^{-\kappa_1}$ for sufficiently large n.*

Proof. We show that, whenever all the quadrants up to level $(i-1)$ are partitioned, the i-th set of $n/2\log\ln n$ customers partitions all the quadrants of the i-th level whp. (if they have not already been partitioned). Consider Figure 8 which depicts a quadrant at the i-th level. The quadrant will be partitioned if 2^i customers fall on the gray area. By Lemma 3, the probability that a customer falls on the gray area is at least $2p^i$. The expected number of the $n/(2\log\ln n)$ customers falling on the gray is at least

$$\frac{n}{2\log\ln n}2p^i = \frac{n}{2\log\ln n}\cdot\frac{2}{(3K+1)^i}.$$

By applying a Chernoff bound, the probability that fewer than 2^i customers fall on the gray area is bounded by

$$e^{-\frac{1}{2}\frac{n}{2\log\ln n}2p^i\epsilon^2}$$

where

$$\epsilon = 1 - \frac{(2\log\ln n)(3K+1)^i 2^{i-1}}{n} = 1 - O\left(\frac{(\ln n)^2\ln\ln n}{n}\right),$$

when $i \le d^- = 2\log\ln n$. It follows that, for a sufficiently large n, $\epsilon \ge 1/2$ and probability that fewer than 2^i customers fall on the gray area is no more than

$$e^{-\frac{1}{8}\frac{n}{2\log\ln n}p^i} < \frac{1}{\kappa_1+2},$$

for any constant κ_1. The total number of quadrants at level i is $4^{i-1} < n$ and hence all the quadrants of level i are partitioned with probability at least $1 - \frac{1}{n^{\kappa_1+1}}$. By performing this analysis until level d^-, it follows that all the quadrants up to d^- are partitioned with probability at least $1 - \frac{1}{n^{\kappa_1}}$. □

Lemma 5. *Consider any quadrant q at level $i > d^-$ and let*

$$M = \sqrt{2}\left(\frac{1}{p} + \frac{4}{p^2} + \frac{16}{p^3} + \frac{64}{p^4} + \frac{256}{p^5}\right).$$

If q has at least M descendants, q is fully subpartitioned 5 levels down with probability at least $1 - n^{\kappa_2}$,

Proof. Assume that, during the execution, quadrant q has more than M descendants, which means that there are more than M facilities in q. Since the maximum distance from any point to some facility inside q is at most $2^{\frac{3}{2}-i}$, this means that at least

$$\frac{M}{2^{\frac{3}{2}-i}} = M\cdot 2^{i-\frac{3}{2}}$$

customers fell in q. The proof shows that, with so many customers, the probability that q has not been fully partitioned 5 level down is at most $n^{-\kappa_2}$. It proceeds level by level, starting from level i.

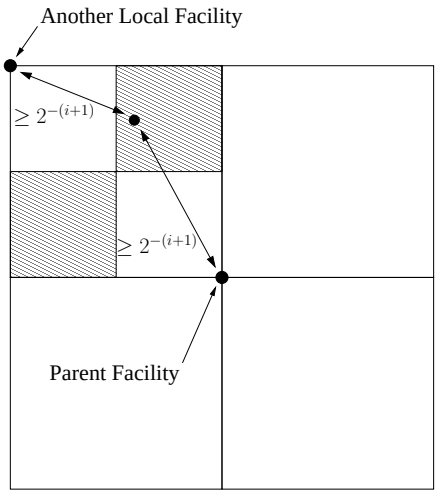

Figure 9: A partition at the $(i+1)$-th level

Consider again Figure 8. Quadrant q is surely partitioned if at least 2^i customers fall on the gray area. By Lemma 3, the probability that a customer falls in the gray area if it falls in q is $2p$. Let $c_1 = 1/(2p)$ and consider the first $c_1 2^i$ of the $\geq M2^{i-\frac{3}{2}}$ customers that have fallen on q. The expected number of these customers falling in the gray area is at least

$$c_1 \cdot 2^i \cdot 2p = 2^{i+1} c_1 \cdot p.$$

Applying a Chernoff bound ($\epsilon = 1 - 1/(c_1 p) \geq 1/2$), the probability that fewer than 2^i of those customers fall in the gray area is bounded by

$$\begin{aligned} e^{-\frac{1}{2}2^{i+1}c_1 p\epsilon^2} &\leq e^{-2^{i-2}c_1 p} \\ &\leq e^{-2^{2\log\ln n - 2}c_1 p} \\ &= e^{-(\ln n)^2 \frac{1}{4} c_1 p} \\ &\leq \frac{1}{n^c}, \end{aligned}$$

for any constant c and for a sufficiently large n.

We now condition on quadrant q having been partitioned and we consider level $i+1$. Quadrant q has 4 subquadrants and we first focus attention on one of them. This quadrant is partitioned if the pertinent gray area receives at least 2^{i+1} customers (Figure 9). By applying Lemma 3 twice, the probability that a customer falls on the gray area, conditioning that it fell in q, is at least $2p^2$. Let $c_2 = 1/(2p^2)$ and consider the next $c_2 2^i$ of the $\geq M2^{i-\frac{3}{2}}$ customers which fell in q (after the $c_1 2^i$ that we just accounted for). The expected number of those customers falling in the gray area is at least

$$c_2 \cdot 2^i \cdot 2p^2 = 2^{i+1} c_2 \cdot p^2.$$

By applying a Chernoff bound ($\epsilon = 1 - 1/(c_1 p^2) \geq 1/2$), the probability that fewer than 2^{i+1} of those customers fall in the gray area is bounded by

$$\begin{aligned} e^{-\frac{1}{2}2^{i+1}c_1 p^2 \epsilon^2} &\leq e^{-2^{i-2}c_1 p^2} \\ &\leq e^{-2^{2\log\ln n - 2}c_1 p^2} \\ &= e^{-(\ln n)^2 \frac{1}{4} c_1 p^2} \\ &\leq \frac{1}{n^c}, \end{aligned}$$

again for any constant c. Since q has 4 subquadrants, it follows that all the 4 subquadrants are partitioned with probability at least $1 - 4n^{-c}$ with the first $4c_2 2^i$ of the $\geq M 2^{i-\frac{3}{2}}$ customers who fell in q.

A similar result holds for the 16 subsubquadrants with $c_3 = 1/(2p^3)$, for the subsequent 64 subquadrants with $c_4 = 1/(2p^4)$, and for the next 256 subquadrants with $c_5 = 1/(2p^5)$. Since

$$M \cdot 2^{i-\frac{3}{2}} = 2^i(c_1 + 4c_2 + 16c_3 + 64c_4 + 256c_5)$$

it follows that, if at least $M 2^{i-\frac{3}{2}}$ customers fall in q, then, for a sufficiently large n, all the 5 levels from i downwards are fully partitioned with probability at least $1 - n^{-\kappa_2}$ for any constant κ_2. □

Lemma 6. *Every disjoint bush has a facility in each of its quadrants in the optimal, offline solution.*

Proof. Consider Figure 10. Since the black quadrant is open, by Lemma 2 ($a = 1/\sqrt{2}$ so $\alpha < 4$), the optimal solution must have a facility in the gray box. □

We are now ready to prove the main result on distributions with smooth neighborhoods.

Theorem 8. *Consider the region I and assume that the customers are mutually independent and obey a distribution ν with smooth neighborhoods. Then, for sufficiently large n, the cost of the solution produced by algorithm $\mathcal{A}$ is only a constant times higher than that of the optimal offline algorithm with probability at least $1 - n^{-1}$,*

Proof. Consider the solution of algorithm $\mathcal{A}$ and the induced partition. As shown earlier, it suffices to bound the number of facilities and we may assume that the customers arrive in random order. By Lemma 4, all quadrants up to level $d^- = 2\log\ln n$ are partitioned whp. By Lemma 5, if a quadrant q at a level $i > d^-$ has more than M descendants, then q is fully partitioned 5 levels down whp. By Lemma 6, every disjoint bush q has a facility in each of its subquadrants in the optimal offline solution. It remains to show that

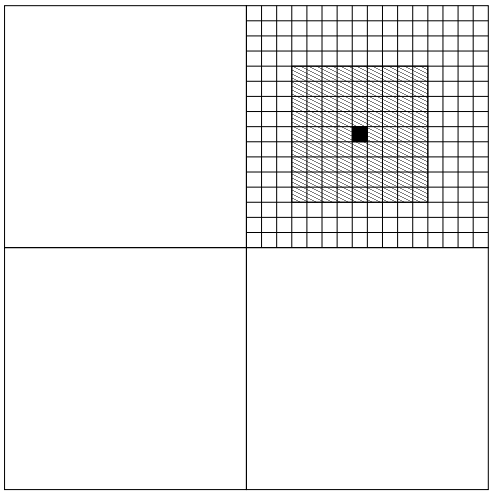

Figure 10: A quadrant of a disjoint bush

1. the cost of the solution is only a constant higher than the cost of the disjoint bushes.

2. the number of disjoint bushes is proportional to the number of facilities in the optimal offline solution.

By Lemma 3, we may eliminate all quadrants that have fewer than M descendants and focus attention on bush children. Indeed, each bush child that is not a bush can have at most $M - 1$ descendants and the number of ignored quadrants is proportional to the number of bush children. Now each bush is either a disjoint bush or it belongs to a disjoint bush (or both). Since the number of bush children is a constant, we can ignore all bushes that are not disjoint bushes, as well as all their children. This proves point (1) above.

Consider now the tree where every node is a disjoint bush and the parent of a disjoint bush q is the closest (i)-ancestor of q which is a disjoint bush. The total number of nodes in the disjoint-bush tree is proportional to the number of leaves and nodes with a single child. By Lemma 6, the optimal solution has at least 4 facilities for every leaf. Similarly, for every node with a single child, it also follows from Lemma 10 that there are at least three facilities in the optimal solution not in the subquadrant of the child, proving point (2) above and concluding the proof. □

The final result of this section identifies a large class of distributions with smooth neighborhoods.

Lemma 7. *Consider a distribution ν on I with probability density function f. If f is continuous on I and uniformly bounded from 0, then ν has smooth neighborhoods.*

Proof. Since f is uniformly bounded from 0, there exists some $\epsilon > 0$ such that $\forall x \in I : f(x) > \epsilon$. Also, since f is continuous on the set I and I is compact, we have that f is bounded on I, i.e., there exists a constant M such that $f(x) < M$ for all $x \in I$.

We now prove that ν satisfies the smooth neighborhoods property for $K = M/\epsilon$. More precisely, we show that

$$\nu(Q_1) \leq \frac{M}{\epsilon} \cdot \nu(Q_2)$$

for two neighboring squares Q_1 and Q_2. Rewrite this relation in terms of the probability density function

$$\int_{Q_1} f(x)\mathrm{d}x \leq \frac{M}{\epsilon} \cdot \int_{Q_2} f(x)\mathrm{d}x.$$

By applying the mean-value theorem (for integrals on $\mathbb{R}^2$), there exist $x_1 \in Q_1$ and $x_2 \in Q_2$ such that

$$f(x_1) = \int_{Q_1} f(x)\mathrm{d}x$$

and

$$f(x_2) = \int_{Q_2} f(x)\mathrm{d}x,$$

The smooth neighborhood property becomes

$$f(x_1) \leq \frac{M}{\epsilon} \cdot f(x_2)$$

and it holds since $f(x_1) < M$ and $f(x_2) > \epsilon$. □

8 Empirical Results

This section describes empirical results on a variety of online facility location problems. To the best of the authors' knowledge, this is the first empirical evaluation of algorithms for this problem. In Tables 1, 2, 3, 4, 5, and 6, the acronym M refers to Meyerson's algorithm of [15] and F to Fotakis' algorithm [7]. The remaining acronyms refer to various ways of choosing facility locations for the partitioning algorithm: C refers to choosing the center point, LC the last customer, and P the average position of customers in the quadrant. All the tables are divided in three parts. The top part of the tables shows results for the region model, while the middle part depicts the results for Meyerson's model. The bottom part gives the performance of Fotakis' algorithm with a parameter around 2 which does not satisfy the competitive ratio criterion given in [7]. This algorithm, denoted by $\tilde{F}$, is however excellent in quality in general and it is interesting to include its results, although

it is very expensive computationally. In particular, Fotakis' algorithm is about 500 times slower than the partitioning algorithm to execute all the benchmarks used in this section.

For each distribution, we give the results when the points come uniformly at random, as well as when the points come in sorted order by their x coordinates. Each column section labels the number of customers generated and summarizes the reported results as an average of 10 problems. The C columns are the total cost and W refers to number of open facilities. Boldface indicates the best results in Meyerson's model. The best result in the region model is boldfaced when it is better than the best result in Meyerson's model. We do not include the last part of the table in computing the best results, since this version of F does not (yet) have any performance guarantee. All algorithms were tested on a variety of parameters and the best parameter over all customer sizes for each problem was chosen. It is important to note that it is possible to do substantially better in some cases using different thresholds on a specific customer size. For the partitioning algorithms, the subscript is the thresholding parameter. In F (resp. $\tilde{F}$), the subscript refers to the x value. It is interesting to see for values of $x \geq 10$, F has a performance guarantee but performs quite poorly. However, from an experimental standpoint, it performs similarly to the algorithms presented here when the constant is lowered and the competitive ratio guarantee is not known to hold. The algorithm by Meyerson does not involve any parameters, however, it is easy to add a parameter that is similar to the ones described here. In the original algorithm, as presented in [15], a new facility is opened at a customer with probability d/f where d is the distance to the nearest facility. It is easy to see that this can be modified to d/af. If a is constant, then the same competitive ratio results hold, albeit with a different constant for the $O(1)$ result that concerns random order of the points. All the problem instances described below exist in the two dimensional unit square.

Uniform Distributions Tables 1 and 2 assess how well the algorithms perform when the customers are distributed in the space uniformly for facility costs of .1 and 1. It is interesting to see here that Meyerson's models perform better than the regions models until high numbers of customers are reached. This may be due, in part, to savings (travel cost of 0) gained by placing a facility with the most recent arriving customer. Overall, $\tilde{F}$ and LC perform the best here, with $\tilde{F}$ having a slight edge in most cases. However, it is interesting to see that whenever the threshold is selected according to the competitive ratio guarantee, F is much worse.

Gaussian Distribution Tables 3 and 4 show the results of the algorithms when the points are drawn from a Gaussian distribution located at the center of the unit square with a parameter of .25 for both the x and y directions with facility costs of .1 and 1. What is clear from these results is that the $\tilde{F}$ method performs the best on the random ordered points. However, its performance degrades more than

Random

	1000		5000		10000		100000	
	C	W	C	W	C	W	C	W
$C_{3.2}$	73.8	278.3	198.7	511.5	312.7	559.7	1465.7	1854.2
$P_{2.7}$	**46.5**	257.1	190.6	1015.2	266.1	1129.4	**1258.6**	4623.6
$M_{1.9}$	**52.6**	234.0	171.8	690.7	281.6	1095.5	1390.9	5066.6
F_{10}	54.6	91.0	196.0	215.0	329.0	288.6	1807.7	1011.2
C_1	69.0	636.0	172.1	1191.7	286.7	1792.5	1315.7	6139.3
$P_{1.6}$	69.0	635.7	**158.2**	1035.4	**255.1**	1385.7	1383.6	5724.6
$LC_{1.1}$	69.0	635.7	166.5	1144	274.7	1723.5	**1300.5**	6170.5
$\tilde{F}_{2.1}$	50.1	183.3	161.4	511.8	264.1	812.7	1288.3	3664.4

Sorted

	1000		5000		10000		100000	
	C	W	C	W	C	W	C	W
$C_{3.6}$	61.9	133.0	183.4	380.6	278.3	407.1	**1297.9**	1607.2
$P_{3.3}$	**50.8**	255.0	200.6	982.8	296.0	1144.9	1457.8	4934.6
M_2	54.5	225.8	172.1	643.2	279.1	1014.7	1353.7	4685.1
F_{10}	60.5	98.9	199.3	241.9	335.4	332.6	1785.6	1132.7
$C_{1.7}$	69.9	641.8	173.1	1047.5	296.5	1491.0	1400.1	5052.6
$P_{1.7}$	69.9	641.8	173.0	1047.5	298.4	1491.0	1510.7	5318.6
$LC_{1.8}$	**51.7**	292.1	**166.9**	680.2	**272.1**	1107.2	**1298.2**	4918.0
$\tilde{F}_{2.1}$	51.1	204.3	162.6	581.3	265.5	923.8	1306.4	4366.6

Table 1: Uniform Distribution, Facility Cost = .1

Random

	1000		5000		10000		100000	
	C	W	C	W	C	W	C	W
$C_{2.4}$	**136.0**	24.1	**410.6**	88.0	628.6	88.7	2966.4	371.6
$P_{2.1}$	141.9	69.7	492.9	278.5	**626.0**	278.3	**2804.3**	1140.4
$M_{1.9}$	145.5	53.5	423.2	152.5	674.0	241.4	3136.6	1095.5
F_{10}	179.8	12.9	571.7	28.5	932.5	43.2	4667.7	165.5
$C_{2.8}$	**133.0**	24.3	**412.4**	88.0	**633.3**	88.7	**2999.5**	355.1
$P_{2.8}$	133.6	24.1	412.9	88.0	638.3	89.7	3016.7	365.8
LC_1	147.2	82.2	444.5	227.5	678.7	332.1	3064.3	1371.7
$\tilde{F}_{2.1}$	132.4	39.1	387.4	110.0	618.0	174.5	2875.6	809.1

Sorted

	1000		5000		10000		100000	
	C	W	C	W	C	W	C	W
$C_{2.5}$	**127.1**	24.1	381.7	88.0	**572.5**	89.7	**2676.1**	428.5
$P_{2.6}$	154.8	70.7	513.4	266.8	707.9	293.8	3228.9	1201.0
$M_{1.9}$	140.7	51.0	411.3	149.5	655.3	230.5	3027.1	1058.2
F_{10}	180.1	14.0	561.4	32.5	911.0	47.3	4505.6	181.5
$C_{1.7}$	151.8	55.0	426.6	121.9	683.8	236.0	3115.2	1093.0
$P_{1.6}$	158.7	58.2	472.5	157.9	727.2	253.5	3393.8	1206.7
$LC_{1.6}$	**139.2**	57.2	**416.7**	145.9	**639.1**	260.3	**2920.5**	1171.7
$\tilde{F}_{2.25}$	133.9	42.8	399.5	126.0	634.8	197.9	2982.2	939.8

Table 2: Uniform Distribution, Facility Cost = 1

Random

	1000		5000		10000		100000	
	C	W	C	W	C	W	C	W
$C_{1.8}$	56.9	262.0	158.2	572.0	246.9	808.8	1114.8	3012.0
$P_{3.5}$	**40.9**	231.1	**136.5**	605.8	226.3	935.8	1130.6	4400.1
$M_{1.9}$	**42.4**	187.5	138.5	557.6	228.2	887.2	1119.2	4101.5
F_{10}	45.1	74.0	160.5	161.3	271.4	230.1	1475.9	803.2
$C_{1.7}$	49.1	413.3	138.1	888.7	217.1	1194.4	**1044.7**	3757.5
$P_{1.7}$	49.1	412.8	138.0	887.7	217.7	1199.7	1048.3	3788.4
$LC_{1.7}$	49.1	413.6	**137.4**	898.1	**216.6**	1234.0	1052.1	4188.8
$\tilde{F}_{2.1}$	40.1	145.0	129.8	412.3	212.0	644.3	1037.6	2950.3

Sorted

	1000		5000		10000		100000	
	C	W	C	W	C	W	C	W
$C_{1.8}$	54.4	258.6	147.3	567.6	227.0	804.8	**1001.3**	2986.3
$P_{3.5}$	45.2	238.6	152.4	652.2	253.4	1046.5	1266.8	5079.6
M_2	42.9	174.6	137.8	520.8	224.0	822.8	1090.2	3753.6
F_{10}	48.4	82.4	165.3	180.0	276.4	259.1	1455.5	903.3
$C_{1.5}$	50.7	413.5	151.3	942.3	240.8	1305.5	1141.4	4463.6
$P_{1.7}$	50.7	411.0	154.1	927.2	249.3	1282.4	1232.4	4389.6
$LC_{1.8}$	**42.1**	241.5	**134.5**	584.2	**219.2**	885.0	**1071.8**	3856.5
$\tilde{F}_{2.1}$	40.6	157.5	30.9	466.3	214.1	747.3	1052.9	3529.5

Table 3: Gaussian Distribution, Facility Cost = .1

the other methods when the points are sorted, indicating it is less able to adapt to problems with more structured orderings. The second best method is LC which has only slightly worse performance but is much faster to compute. It is also important to note that in this problem, algorithm F is unable to match the performance of the other methods when 10 is used as the parameter value in order to insure the competitive ratio guarantee.

New England Population Distributions Table 5 uses the 2000 United States census [21] to determine the populations of towns in New England. By letting P_{ne} be the total population of New England and P_t the population of a town t in New England, a customer is generated by first choosing a town t with probability $\frac{P_t}{P_{ne}}$ and then drawing from a Gaussian distribution with parameter .1 in the x and y direction. Figure 11 gives a picture of how the towns are distributed in the space. In this case, the facility cost is 10. With a higher facility cost, much fewer facilities are placed and it is critical that they be placed correctly. Once again $\tilde{F}$ performs slightly better then the partitioning algorithms, but LC surpasses $\tilde{F}$ when the points come in sorted order.

Random

	1000		5000		10000		100000	
	C	W	C	W	C	W	C	W
$C_{2.4}$	114.2	27.8	326.8	66.7	531.3	109.2	2510.8	517.8
$P_{4.6}$	108.0	35.3	335.7	115.5	526.1	164.5	2508.8	794.0
$M_{1.9}$	110.0	38.8	331.4	122.7	533.9	193.6	2519.4	890.2
F_{10}	142.5	7.4	463.4	19.8	759.5	29.8	3809.4	125.9
$C_{1.4}$	104.0	46.3	**313.2**	117.9	**503.6**	181.1	**2379.4**	782.6
$P_{1.7}$	110.1	38.6	321.5	100.0	513.3	155.9	2407.0	683.1
$LC_{1.4}$	**103.9**	49.0	323.0	137.6	516.1	206.3	2451.4	915.7
$\tilde{F}_{2.1}$	102.1	28.3	310.7	88.2	494.8	138.3	2312.7	648.2

Sorted

	1000		5000		10000		100000	
	C	W	C	W	C	W	C	W
$C_{1.5}$	**101.0**	33.8	**290.9**	91.9	**466.5**	152.5	**2143.5**	684.7
$P_{3.6}$	122.0	47.2	371.1	143.0	601.3	240.6	2834.5	1110.7
$M_{2.1}$	108.8	36.8	323.7	111.7	518.2	172.0	2424.6	789.3
F_{10}	145.2	8.8	452.4	22.8	737.9	34.5	3652.4	139.3
$C_{1.5}$	111.6	40.7	333.3	121.2	534.2	193.5	2486.4	869.3
$P_{1.4}$	119.7	49.0	366.1	139.1	583.4	216.3	2736.3	974.1
$LC_{1.8}$	**106.0**	37.5	**317.9**	114.5	**508.4**	178.3	**2393.4**	815.2
$\tilde{F}_{2.3}$	103.7	31.3	315.6	95.4	505.9	152.9	2401.8	738.2

Table 4: Gaussian Distribution, Facility Cost = 1

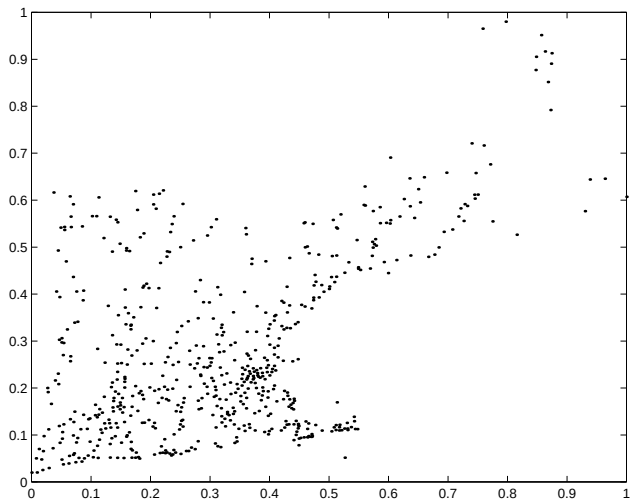

Figure 11: New England Town Locations

Random

	1000		5000		10000		100000	
	C	W	C	W	C	W	C	W
$C_{1.4}$	206.4	8.0	611.2	22.1	925.8	29.8	4274.3	130.1
$P_{4.6}$	**186.7**	7.5	**572.1**	20.3	949.2	31.8	4555.2	157.3
$M_{2.1}$	194.3	7.2	594.9	21.1	959.0	32.2	4534.4	145.5
F_{10}	216.6	1.8	734.3	3.1	1291.8	5.2	6856.0	20.1
$C_{1.7}$	187.6	7.9	578.3	21.0	908.4	29.6	4271.1	126.0
$P_{1.7}$	**187.0**	8.0	577.2	21.0	**880.6**	27.0	**4166.2**	117.0
LC_2	190.4	7.7	**575.5**	19.8	918.6	29.3	4398.8	129.6
$\tilde{F}_{2.1}$	171.0	4.2	551.0	15.6	862.1	23.5	4132.0	118.0

Sorted

	1000		5000		10000		100000	
	C	W	C	W	C	W	C	W
$C_{1.7}$	209.4	8.0	590.8	19.6	**891.0**	28.0	**3864.9**	107.5
$P_{4.8}$	244.8	10.8	613.7	17.6	1079.4	38.1	5162.7	179.1
$CP_{3.5}$	210.8	10.0	644.5	33.0	1123.6	63.5	4923.9	279.0
$M_{2.1}$	**199.2**	7.0	602.1	19.6	951.1	33.1	4376.8	142.0
F_{10}	228.8	2.0	881.4	3.6	1401.3	5.2	6879.7	20.8
$C_{1.5}$	208.7	8.6	606.1	23.6	926.7	32.7	4424.9	157.0
P_2	215.3	7.4	656.1	20.6	1021.4	28.0	4877.0	133.5
LC_2	201.8	7.9	**573.1**	21.1	**893.5**	28.6	**4258.5**	133.5
$F_{2.4}$	189.1	4.9	566.7	15.3	909.3	24.3	4362.4	125.7

Table 5: New England Population Distribution, Facility Cost = 10

Random

	1000		5000		10000		100000	
	C	W	C	W	C	W	C	W
$C_{1.3}$	764.2	5.0	1781.0	8.5	2479.1	10.0	10686.8	36.1
P_5	572.5	4.0	1448.8	6.7	**2067.6**	7.7	10584.4	36.7
$M_{1.9}$	458.8	2.2	1452.9	4.5	2417.6	9.0	10861.8	40.6
F_{10}	**391.8**	1.0	1376.4	1.3	2521.0	2.7	12710.3	8.5
$C_{1.3}$	605.0	4.4	1463.6	7.7	**2078.2**	9.4	10523.4	41.3
$P_{1.4}$	594.3	4.2	1373.1	6.7	2108.6	9.3	**9658.9**	33.6
LC_2	580.5	4.0	**1305.7**	5.7	2099.6	8.3	10101.0	33.0
$\tilde{F}_{2.25}$	389.9	1.3	1381.6	4.2	2180.9	6.5	9337.3	23.8

Sorted

	1000		5000		10000		100000	
	C	W	C	W	C	W	C	W
$C_{1.6}$	767.8	5.0	1668.8	7.0	2587.5	10.0	10320.4	33.2
P_5	**658.4**	4.0	1808.9	7.4	2783.1	7.7	12416.0	40.2
$M_{1.9}$	707.2	2.6	1632.0	5.9	2507.0	8.5	10541.1	36.8
F_{10}	684.8	1.8	2231.7	2.2	2866.3	3.0	13547.9	10.0
$C_{1.3}$	725.2	5.0	1684.0	8.0	2546.2	11.4	11116.8	47.2
$P_{1.2}$	729.8	5.0	1797.5	9.0	2732.9	12.0	11874.1	50.0
$LC_{1.7}$	**681.4**	4.4	**1630.4**	7.2	**2438.9**	10.4	**10442.1**	39.6
$\tilde{F}_{2.4}$	570.1	2.2	1472.3	5.0	2374.8	8.0	9898.0	26.8

Table 6: Rhode Island Population Distribution, Facility Cost = 100

Rhode Island Population Distribution Finally, a model similar to the New England population distribution is used for the cities of Rhode Island. The Gaussian parameter remains .1, but the facility cost is changed to 100. The results are shown in Table 6. Figure 12 shows the cities mapped onto a 2 dimensional plane. The results here are more muddled, though it appears that LC generally performs the best in the random ordered case.

Summary Overall the experimental results are very favorable to the partitioning algorithm. The partitioning algorithm LC is generally the best partitioning version (although C is also very robust) and it is only slightly outperformed as far as quality is concerned by $\tilde{F}$. Algorithm $\tilde{F}$ however is much more demanding computationally and much more complicated to implement. It also does not have a performance guarantee of now and its performance degrades considerably when its parameter is chosen to obtain such a guarantee. The deterministic partitioning algorithm almost always outperforms Meyerson's randomized algorithm and the benefits can be quite significant sometimes.

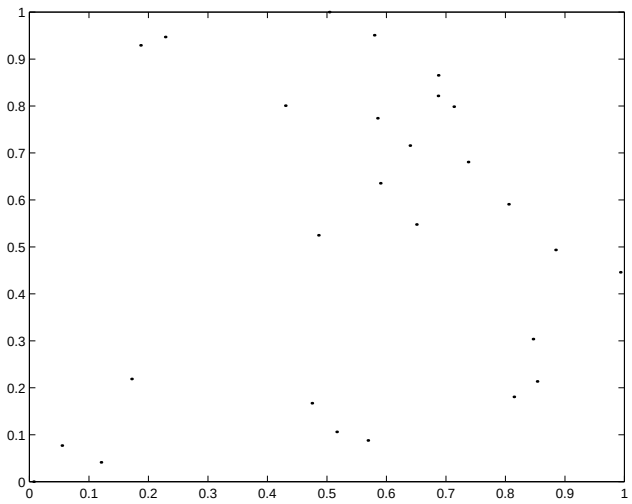

Figure 12: Rhode Island Town Locations

9 Conclusion and Future Work

This paper reconsidered online facility location and presented a simple and deterministic competitive algorithm for this problem. The algorithm, whose key idea is a hierarchical partitioning based on thresholding, is very simple to implement and runs in $O(n \log n)$, where n is the number of customers. The paper showed that the algorithm is $O(\log n)$-competitive for a variety of models, including the region model, Meyerson's model where facilities must co-exist with existing customers, and the fixed location model. The paper also presented the first probabilistic analysis of online facility location, showing that the partitioning algorithm is $O(1)$-competitive for any arrival order whenever the customers are uniformly distributed in the region. Experimental results have shown that the algorithm behaves very well in practice under a variety of hypotheses. It is only slightly outperformed by a version of Fotakis' algorithm that is not guaranteed to be competitive and is much more demanding computationally. The experimental results also show that our algorithm can bring significant benefits compared to Meyerson's algorithm.

There are still various open issues for future research. First, it is important to extend the partitioning idea to other, and perhaps all, metric spaces. It is also interesting to generalize online facility location algorithms to account for non-uniform facility costs. This would probably requiring changing the sizes of the partitioning dynamically. It is also important to develop a model for online facility location that allows for capacitated facilities and the closing and re-opening of facilities as featured in a variety of networking and mobile computing applications. The partitioning scheme described here would naturally extend to such models. On the practical side, it may be interesting to evaluate empirically adaptive versions of the

algorithms where thresholds are refined on the fly. Indeed, early experimental results indicate it is better to have higher thresholds for smaller numbers of customers. On the theoretical side, there are many issues left open with the probabilistic analysis. These include identifying which distributions have smooth neighborhoods, and generalizing the proof to weaker properties, since we believe that the algorithm would behave well in many other contexts.

Acknowledgments

This research is partially supported by NSF ITR Award DMI-0121495 and by an NDSEG fellowship from ASEE.

References

[1] M. Adler, S. Chakrabarti, M. Mitzenmacher, and L. Rasmussen. Parallel randomized load balancing. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on the Theory of Computing*, pages 238–247, Las Vegas, Nevada, 29 May–1 June 1995.

[2] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, Sept. 1975.

[3] E. Chávez, G. Navarro, R. Baeza-Yates, and J. L. Marroquín. Searching in metric spaces. *ACM Computing Surveys*, 33(3):273–321, Sept. 2001.

[4] F. A. Chudak and D. B. Shmoys. Improved approximation algorithms for a capacitated facility location problem. In *Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '99)*, pages 875–876, N.Y., Jan. 17–19 1999. ACM-SIAM.

[5] G. Cornuéjols, G. Nemhauser, and L. Wolsey. *Discrete Location Theory*, chapter The Uncapacitated Facility Location Problem, pages 119–171. Lecture Note in Artificial Intelligence (LNAI 1865). Wiley, 1990.

[6] D. Erlenkotter. A Dual-Based Procedure for Uncapacitated Facility Location: General Solution Procedures and Computational Experience. *Operations Research*, 26:992–1009, 1978.

[7] D. Fotakis. Online facility location. In *Proceedings of the 13th International Colloquium on Automata, Languages and Programming (ICALP '03)*, pages 637–652, 30 June–4 July 2003.

[8] L. Gao and E. Robinson. Uncapacitated Facility Location: General Solution Procedures and Computational Experience. *European Journal of Operations Research*, 76:410–427, 1994.

[9] S. Guha and S. Khuller. Greedy strikes back: Improved facility location algorithms. In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '98)*, pages 649–657, San Francisco, California, 25–27 Jan. 1998.

[10] K. Jain and V. V. Vazirani. Primal-dual approximation algorithms for metric facility location and k-median problems. In IEEE, editor, *Proceedings of the 40th Annual Symposium on Foundations of Computer Science: October 17–19, 1999, New York City, New York,*, pages 2–13, 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA, 1999. IEEE Computer Society Press.

[11] D. Karger and M. Ruhl. Finding nearest neighbors in growth-restricted metrics. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC '02)*, pages 741–750, New York, May 19–21 2002. ACM Press.

[12] J. Kratica, D. Tosic, V. Filipovic, and I. Ljubic. Solving the Simple Plant Location Problems by Genetic Algorithm. *RAIRO Operations Research*, 35:127–142, 2001.

[13] M. Mahdian, Y. Ye, and J. Zhang. Improved approximation algorithms for metric facility location problems. In *Proceedings of Fifth International Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX '02)*, volume 2462 of *Lecture Notes in Computer Science*, pages 229–242, 2002.

[14] R. R. Mettu and C. G. Plaxton. The online median problem. In IEEE, editor, *Proceedings of the 41st Annual Symposium on Foundations of Computer Science (FOCS '00)*, pages 339–348, 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA, 2000. IEEE Computer Society Press.

[15] A. Meyerson. Online facility location. In IEEE, editor, *42nd IEEE Symposium on Foundations of Computer Science: proceedings: October 14–17, 2001, Las Vegas, Nevada, USA*, pages 426–431, 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA, 2001. IEEE Computer Society Press.

[16] L. Michel and P. Van Hentenryck. A Simple Tabu-Search Algorithm for Warehouse Location. *European Journal of Operations Research*, 2003. to appear.

[17] M. Mitzenmacher. *The Power of Two Choices in Randomized Load Balancing.* PhD thesis, University of California, Berkeley, August 1996.

[18] S. M. Ross. *A First Course in Probability.* Macmillan, London, 5th edition, 1998.

[19] D. Shmoys. Approximation algorithms for facility location problems. In *Proceedings of Third International Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX '00)*, volume 1913 of *Lecture Notes in Computer Science*, pages 27–33, 2000.

[20] D. B. Shmoys, É. Tardos, and K. Aardal. Approximation algorithms for facility location problems. In *Proceedings of the 29th Annual ACM Symposium on the Theory of Computing (STOC '97)*, pages 265–274, New York, May 1997. Association for Computing Machinery.

[21] United States Census Bureau. Year 2000 Population Estimates. *http://www.census.gov/*, 2000.