Facility Location in Sublinear Time

Mihai Bădoiu[1], Artur Czumaj[2,⋆], Piotr Indyk[1], and Christian Sohler[3,⋆⋆]

[1] MIT Computer Science and Artificial Intelligence Laboratory,
Stata Center, Cambridge, Massachusetts 02139, USA
{mihai, indyk}@theory.lcs.mit.edu

[2] Department of Computer Science,
New Jersey Institute of Technology,
Newark, NJ 07102, USA
czumaj@cis.njit.edu

[3] Heinz Nixdorf Institute and Computer Science Department,
University of Paderborn, D-33102 Paderborn, Germany
csohler@uni-paderborn.de

Abstract. In this paper we present a randomized constant factor approximation algorithm for the problem of computing the optimal *cost* of the metric Minimum Facility Location problem, in the case of uniform costs and uniform demands, and in which every point can open a facility. By exploiting the fact that we are approximating the optimal cost without computing an actual solution, we give the first algorithm for this problem with running time $O(n \log^2 n)$, where n is the number of metric space points. Since the size of the representation of an n-point metric space is $\Theta(n^2)$, the complexity of our algorithm is *sublinear* with respect to the input size.

We consider also the general version of the metric Minimum Facility Location problem and we show that there is no $o(n^2)$-time algorithm, even a randomized one, that approximates the optimal solution to within any factor. This result can be generalized to some related problems, and in particular, the cost of minimum-cost matching, the cost of bichromatic matching, or the cost of $n/2$-median cannot be approximated in $o(n^2)$-time.

1 Introduction

The design of algorithms operating on massive data sets has received a lot of attention in recent years. The practical motivation of this study is that polynomial-time algorithms that are efficient in relatively small inputs, may become impractical for input sizes of several gigabytes. For example, when we consider approximation algorithms for clustering problems in metric spaces then they typically

[⋆] Research supported in part by NSF grant ITR-CCR-0313219.

[⋆⋆] Research supported in part by DFG grant Me 872/8-2 and by the IST program of EC under contract no. IST-2002-001-907 (DELIS).

L. Caires et al. (Eds.): ICALP 2005, LNCS 3580, pp. 866–877, 2005.

have $\Omega(n^2)$ running time where n is the number of input points. Clearly, such a running time is not feasible for massive data sets. But for many problems — like the facility location problem considered in this paper — such a running time is provably unavoidable. Surprisingly, these lower bounds do not necessarily hold when one wants to estimate the *cost* of an optimal solution. In this paper we will indeed show that one can find a constant factor approximation algorithm for the metric uncapacitated facility location problem with uniform costs and in which every point can open a facility, that runs in $O(n \log^2 n)$ time, that is, in time *sublinear* in the input size.

Our approach is motivated by the fact that in many applications it suffices to know an *approximate cost* of the facility location problem rather than to find an approximate solution to the facility location problem. Let us consider the example that a company wants to invest money and it can relate the cost of the facility location problem to the possible return on investment. Then it would first solve an instance of the problem for every market to find out the most profitable one. In such a situation it is sufficient to know the return on investment before one decides which market to enter. It is not (yet) necessary to know how to achieve it. Finally, when one knows which market to enter one only has to compute a solution to a single instance of the problem. Therefore, if one could approximate the *cost* of an optimal solution significantly faster than finding such a particular approximate solution this would significantly speed up the market analysis.

Similar arguments hold for another popular application of facility location algorithms, that of clustering data sets. In particular, it is good to know if the data can be "well-clustered" before actually attempting to find the clustering.

1.1 Our Results

In this paper we consider the metric Minimum Facility Location problem with uniform opening costs and demands, and in which every point can open a facility. We give a randomized $O(1)$-approximation algorithm for this problem that runs in time $O(n \log^2 n)$, where n is the number of metric space points. Since the size of the representation of an n-point metric space is $\Theta(n^2)$, the complexity of our algorithm is *sublinear* with respect to the input size. No $o(n^2)$-time approximation algorithm for this problem was known before. It has been known that any constant factor approximation algorithm that returns not only the cost, but also a solution itself, requires the running time of $\Omega(n^2)$ [14].

Next, we prove that if the set of facilities and the cities (points that are to be connected to the facilities) are allowed to be disjoint, then any, *even randomized*, approximation algorithm for the cost of the Minimum Facility Location that guarantees any bounded approximation ratio for the cost, requires time $\Omega(n^2)$. This bound holds even when the opening costs and demands are uniform. Furthermore, our proof can be extended to the problems of estimating the *cost of minimum-cost matching*, the *cost of bi-chromatic matching*, and the *cost of k-median* for $k = n/2$; all these problems require $\Omega(n^2)$ to estimate the cost of their optimal solution to within any factor. We feel that these results demonstrate that most optimization problems for metric instances do not have sublinear-time algorithms even to estimate well the cost of the optimal solution; results like our sublinear-time algorithm for a $O(1)$-factor approximation of the cost of the optimum solution for the metric uniform Minimum Facility Location problem are rare (see however, [4, 6, 7]).

1.2 Our Techniques

Our analysis of a sublinear-time algorithm consists of two principal steps: we first prove the existence of an appropriated estimator for the cost of the Minimum Facility Location problem and then we show how such an estimator can be approximated in time $O(n \log^2 n)$. Our estimator is obtained by extending the primal-dual approach from [12]: for each point we define an approximation of the contribution of that point to the total cost, and then we prove that the sum of the contributions for all the points approximates the cost of the Minimum Facility Location problem. An important property of our estimator is that it can be efficiently approximated by *adaptive sampling*. We first prove that the individual value of an estimator for any single point can be efficiently approximated by sampling with the running time depending on the value of the estimator, and then we apply another adaptive sampling scheme to efficiently approximate the sum of the estimators. A similar approach has been used in recent sublinear-time algorithms for estimating the cost of the minimum spanning tree problem in [2] and [4].

1.3 Definition of the Problem

The formal definition of the general form of the (Metric) *Minimum Facility Location* problem is as follows: We are given a metric (P, D), and a subset $\mathcal{F} \subseteq P$ of *facilities*. For each facility $v \in \mathcal{F}$, we are given a nonnegative *cost* $\mathfrak{f}(v)$, and for each point $u \in P$, a nonnegative *demand* $d(u)$. The problem consists of finding a set $F \subseteq \mathcal{F}$, so as to minimize

$$\sum_{v \in F} \mathfrak{f}(v) + \sum_{u \in P} d(u) \cdot D(u, F) \enspace ,$$

where $D(u, F) = \min_{v \in F} D(u, v)$.

In this paper we focus on the variant of the facility location problem with $\mathcal{F} = P$ and in which the costs as well as the demands are uniform. That is, for each $v \in \mathcal{F}$, $\mathfrak{f}(v) = c$ for some $c > 0$, and for each $u \in P$, $d(u) = 1$. Observe that we can assume that $c = 1$, if we re-scale the given metric, by dividing all the distances by c. In what follows, we will refer to this variant of the facility location problem as *uniform*.

The key property of our formulation, is that we are interested in computing the cost of the optimal solution, without computing a solution itself. Thus, in what follows, our task is to approximate the value:

$$\min_{F \subseteq P} |F| + \sum_{u \in P} D(u, F) \enspace .$$

In the final part of the paper we also consider a more general variant of the problem when P and $\mathcal{F}$ do not have to be the same. We prove in Theorem 2 that in that case there is no hope to obtain a sublinear-time algorithm.

1.4 Previous Work

The *Minimum Facility Location* problem is one of the most extensively studied problems in combinatorial optimization. The problem is known to be $\mathcal{NP}$-hard and the first constant factor approximation algorithm was given by Shmoys et al. [13]. Several other approximation algorithms are given in [1, 3, 8]. The best approximation ratio of 1.52, is due to Madhian, Ye, and Zhang [10], while the best lower bound of 1.463 for the approximation ratio is due to Guha and Khuller [5].

The first constant factor approximation algorithm with almost linear running time (that is, the running time of $O(n^2 \log n)$) was given by Jain and Vazirani [9]; Mettu and Plaxton [12] gave a simple $O(n^2)$-time constant approximation ratio algorithm. Thorup [14] considered the facility location problem in metric spaces defined by a graph. If the underlying graph has m edges, then even though the metric space is of size $\Theta(n^2)$, Thorup gives a constant-factor approximation algorithm running in time $\tilde{O}(m)$; this is a sublinear time for sparse graphs. On the other hand, it has been shown [14] that for general metric spaces, any constant factor approximation algorithm, even a randomized one, requires running time of $\Omega(n^2)$. Notice that this does not exclude the possibility of approximating the *cost* of the Minimum Facility Location problem in sublinear time, in particular, in time $O(n \,\mathrm{polylog}(n))$.

2 Estimating the Cost of Uniform Minimum Facility Location

In this section we present an $O(n \log^2 n)$ time algorithm that approximates the cost of the Minimum Facility Location in the uniform case, that is, when the costs as well as the demands are uniform.

2.1 Preliminaries

Let (P, D) be a metric with a point set $P = \{p_1, \dots, p_n\}$. For any point $p_i \in P$, and for any $r \geq 0$, we denote by $B(p_i, r)$ the set of points in P which are at distance at most r from p_i. For each i, $1 \leq i \leq n$, let $r_i > 0$ be the number satisfying

$$\sum_{p \in B(p_i, r_i)} (r_i - D(p_i, p)) = 1 \ .$$

Observe that the value $\sum_{p \in B(p_i, r)} (r - D(p_i, p))$ is continuous and strictly monotonically increasing with r. Thus, there exists a unique value r_i satisfying the above equality. Moreover, for any i, $1 \leq i \leq n$, we have $1/n \leq r_i \leq 1$.

We begin with a lemma that establishes the relation between the value of r_i and the size of $B(p_i, r_i)$.

Lemma 1. *For every i, with $1 \le i \le n$, we have* $\frac{1}{|B(p_i,r_i)|} \le r_i \le \frac{2}{|B(p_i,r_i/2)|}$.

Proof. By the definition of r_i, we have $\sum_{p \in B(p_i,r_i)} (r_i - D(p_i, p)) = 1$, which implies $\sum_{p \in B(p_i,r_i)} r_i \ge 1$, and thus $r_i \ge 1/|B(p_i, r_i)|$. The other inequality follows directly from the following,

$$1 = \sum_{p \in B(p_i,r_i)} (r_i - D(p_i, p)) \ge \sum_{p \in B(p_i,r_i/2)} (r_i - D(p_i, p)) \ge |B(p_i, r_i/2)| \cdot r_i/2. \quad \square$$

MP algorithm. In our analysis we will use a simple approximation algorithm for the Minimum Facility Location problem due to Mettu and Plaxton [12]; we will refer to that algorithm as the *MP algorithm.*

1. Compute the value of r_i for every $p_i \in P$.
2. Sort the input such that $r_1 \le r_2 \le \cdots \le r_n$.
3. For $i = 1$ to n: if there is no open facility in $B(p_i, 2\,r_i)$ then open the facility at p_i.

Mettu and Plaxton [12] proved that this simple algorithm will return a set of open facilities for which the total cost is at most 3 times the minimum.

2.2 Cost Estimation

In this section, we show that the sum of the radii approximates the optimal cost of the facility location to within a constant factor. Our analysis uses the relation between the sum $\sum_{p_i \in P} r_i$ and the cost of optimal solution and that of the solution obtained by the MP algorithm discussed above.

Let C_{OPT} be the cost of an optimal solution. Let also F_{MP} be the set of facilities computed by the MP algorithm. For this solution given by the MP algorithm, we define C_{MP}, C^c_{MP}, and C^f_{MP} to be the total cost, the connection cost, and the facility cost respectively.

The following lemma shows that the sum of the radii estimates well C_{OPT}.

Lemma 2. $\frac{1}{4} \cdot C_{OPT} \le \sum_{p_i \in P} r_i \le 6 \cdot C_{OPT}$.

Proof. We first prove the lower bound that $C_{OPT} \le 4 \cdot \sum_{p_i \in P} r_i$ and then the upper bound that $\sum_{p_i \in P} r_i \le 6 \cdot C_{OPT}$.

Lower bound: Since in the MP algorithm for every $p_i \in P$ there is an open facility within distance at most $2\,r_i$ (for if not, then the algorithm would open the facility at p_i), we get that $2 \sum_{p_i \in P} r_i \ge C^c_{MP}$.

It remains to show that $\sum_{p_i \in P} r_i$ is an upper bound for C^f_{MP}. We first observe that every $p_i \in P$ is contained in at most one ball $B(p_j, r_j)$, for some $p_j \in F_{MP}$. Indeed, if $p_i \in B(p_j, r_j) \cap B(p_k, r_k)$ for some $p_j, p_k \in F_{MP}$, $j < k$, then since

$r_j \le r_k$, we would have $p_j \in B(p_k, 2\,r_k)$. But this implies that the MP algorithm would not open the facility at p_k, a contradiction.

This observation yields:

$$\sum_{p_i \in P} r_i \;\ge\; \sum_{p_j \in F_{MP}} \; \sum_{p_k \in B(p_j, r_j)} r_k \;\; . \tag{1}$$

Next, we observe that if $p_j \in F_{MP}$ and $p_k \in B(p_j, r_j)$, then we must have $r_j \le 2\,r_k$. Indeed, for if not, then we would have $B(p_k, 2\,r_k) \subseteq B(p_k, r_j) \subseteq B(p_k, r_j + D(p_j, p_k)) \subseteq B(p_j, 2\,r_j)$, and thus the MP algorithm would not open the facility at p_j, a contradiction. This observation can be now combined with (1) to conclude:

$$\begin{aligned} \sum_{p_i \in P} r_i \ge \sum_{p_j \in F_{MP}} \; \sum_{p_k \in B(p_j, r_j)} r_k \;&\ge\; \sum_{p_j \in F_{MP}} \; \sum_{p_k \in B(p_j, r_j)} r_j/2 \\ = \tfrac{1}{2} \cdot \sum_{p_j \in F_{MP}} r_j \cdot |B(p_j, r_j)| \;&\ge\; \tfrac{1}{2} \cdot \sum_{p_j \in F_{MP}} 1 \;=\; \tfrac{1}{2} \cdot C^f_{MP} \;\; , \end{aligned}$$

where the second inequality follows from the fact that $r_j \ge 1/|B(p_j, r_j)|$ (Lemma 1). Thus, we have $2 \cdot \sum_{p_i \in P} r_i \ge C^c_{MP}/2 + C^f_{MP}/2 \ge C_{MP}/2 \ge C_{OPT}/2$.

Upper bound: Next, we show that the sum of the radii is not much bigger than the cost of optimal solution. Before we proceed, we introduce one definition from [12]. For a set $X \subseteq P$ and a point $p_i \in P$, we define

$$\mathit{charge}(p_i, X) = D(p_i, X) + \sum_{p_j \in X} \max\{0, r_j - D(p_i, p_j)\} \;\; .$$

Mettu and Plaxton proved [12] that $C_{MP} = \sum_{p_i \in P} \mathit{charge}(p_i, F_{MP})$.

Now we are ready to prove that $\sum_{p_i \in P} r_i \le 2 \cdot C_{MP}$ what will imply that $\sum_{p_i \in P} r_i \le 6 \cdot C_{OPT}$. We have,

$$\begin{aligned} 2 \cdot C_{MP} &= 2 \cdot \sum_{p_i \in P} \mathit{charge}(p_i, F_{MP}) \\ &\ge 2 \cdot \left(\sum_{p_i \in F_{MP}} r_i + \sum_{p_j \in P \setminus F_{MP}} \max\{r_{\delta(j)}, D(p_j, p_{\delta(j)})\} \right) \;\; , \end{aligned}$$

where $\delta(j)$ denotes the index of the facility in F_{MP} that is closest to p_j. We want to show

$$2 \cdot \left(\sum_{p_i \in F_{MP}} r_i + \sum_{p_j \in P \setminus F_{MP}} \max\{r_{\delta(j)}, D(p_j, p_{\delta(j)})\} \right) \ge \sum_{p_i \in P} r_i \;\; .$$

We will show that $r_j \le D(p_j, p_{\delta(j)}) + r_{\delta(j)}$, which immediately implies the above inequality because then $\max\{r_{\delta(j)}, D(p_j, p_{\delta(j)})\} \ge r_j/2$. Assume $r_j > D(p_j, p_{\delta(j)}) + r_{\delta(j)}$. In this case we have $B(p_{\delta(j)}, r_{\delta(j)}) \subseteq B(p_j, r_j)$. We get

$$\begin{aligned}\sum_{p\in B(p_j,r_j)} (r_j - D(p_j,p)) &\geq \sum_{p\in B(p_{\delta(j)},r_{\delta(j)})} (r_j - D(p_j,p)) \\ &> \sum_{p\in B(p_{\delta(j)},r_{\delta(j)})} (r_{\delta(j)} - D(p_{\delta(j)},p)) \ = \ 1 \ .\end{aligned}$$

This is a contradiction because the definition of r_j requires

$$\sum_{p\in B(p_j,r_j)} (r_j - D(p_j,p)) = 1 \ .$$

To summarize, we have proven that $2 \cdot C_{MP} \geq \sum_{p_i \in P} r_i$, and now the lower bound follows from the fact that $C_{MP} \leq 3 \cdot C_{OPT}$ [12]. □

2.3 Estimating the Cost of the Facility Location Problem

From the previous section we know that to approximate the cost of the facility location problem it suffices to estimate the sum $\sum_i r_i$ of the radii $r_1, \ldots, r_n$ of the points $p_1, \ldots, p_n$. A standard approach to this problem would be to sample a set of s points (for a suitable s), determine (possibly approximately) their radii, and then output n times their average radius as an approximation for $\sum_i r_i$. However, this approach cannot lead to a sublinear-time algorithm for the following reason. In general, the time to determine the radius of a point in $\Omega(n)$. For example, this might be the case when the radius is constant, because there is only a constant number of points within the radius. Therefore, to certify that a point has constant radius the algorithm must be able to certify that no more than a constant number of points are within the radius. This task cannot be done in $o(n)$ time (even if one aims at an approximation and uses randomization). We also note that, in general, $s = \Omega(n)$, if we need a constant factor approximation of $\sum_i r_i$. This follows from standard Chernoff-Hoeffding bounds (which are essentially tight in this setting) and the fact that the average radius can be as small as $1/n$. Therefore, this standard sampling approach would not give us a sublinear time algorithm.

In the following we will show that an *adaptive sampling* algorithm can estimate the size of r_i in $O(r_i \, n \log n)$ time (recall that $r_i < 1$). We start with a constant size sample of points and determine their average radius. If our sample is too small we double it and continue until we have found a sample of sufficient size. For the analysis we will parameterize the sample size s by the average value of the r_i. Combining this with the running time of the adaptive algorithm leads to a sublinear algorithm. Details follow in the next two subsections.

2.4 Estimating r_i

In this section we present an algorithm that for a given i, in time $O(r_i \, n \log n)$ approximate the value of r_i to within a constant factor, with high probability.

Let us fix i. Our approach of estimating the value of r_i is by approximating the value of r for which $B(p_i, r)$ contains approximately $1/r$ points. This is formalized in the following lemma.

Lemma 3. *Let j_0 be the maximum integer j, with $1 \leq j \leq \log n$, such that $|B(p_i, 2^{-j})| \geq 2^j$. Then, we have $2^{-(j_0+1)} \leq r_i \leq 2^{-j_0+1}$.*

Proof. We will use Lemma 1. By our assumption about j_0, we have $|B(p_i, 2^{-(j_0+1)})| < 2^{j_0+1}$ and $|B(p_i, 2^{-j_0})| \geq 2^{j_0}$. The first inequality implies that for any $r < 2^{-(j_0+1)}$, $|B(p_i, r)| \leq |B(p_i, 2^{-(j_0+1)})| < 2^{j_0+1} < 1/r$. This bound together with the lower bound in Lemma 1 yield that $r_i \geq 2^{-(j_0+1)}$. On the other hand, the inequality $|B(p_i, 2^{-j_0})| \geq 2^{j_0}$ implies that for any $r > 2^{-j_0+1}$, $|B(p_i, r/2)| \geq |B(p_i, 2^{-j_0})| \geq 2^{j_0} > 2/r$. Therefore, by the upper bound in Lemma 1 we must have $r_i \leq 2^{-j_0+1}$. □

Lemma 3 implies that in order to estimate r_i, it suffices to estimate the value of j_0. Our algorithm to estimate j_0 runs as follows: We begin with setting $j = \log n$, and then we are decreasing j by one until for the first time $|B(p_i, 2^{-j})| \geq 2^j$. Since computing $|B(p_i, 2^{-j})|$ exactly requires $\Omega(n)$ time, we only approximate $|B(p_i, 2^{-j})|$ by *random sampling*. This reduces the running time. At each step, we pick uniformly at random, and with replacement, $K_j = c\, 2^{-j}\, n \log n$ sample points to estimate the value of $|B(p_i, 2^{-j})|$, where c is a sufficiently large constant. Let N_j be the number of sample points that are inside the ball $B(p_i, 2^{-j})$. We return $\beta_j = n\, N_j / K_j$ as the estimator of $|B(p_i, 2^{-j})|$.

In the following three lemmas we first analyze the quality of the estimator β_j and then discuss the running time of this sampling scheme.

Lemma 4. *If $j \geq j_0 + 2$, then $\mathbf{Pr}[\beta_j \geq 2^j] < 1/poly(n)$.*

Proof. Since $j \geq j_0 + 2$, it follows that $B(p_i, 2^{-j}) \subseteq B(p_i, 2^{-(j_0+1)})$. Let q be the probability that a randomly chosen sample point is in $B(p_i, 2^{-j})$. We have $q \leq |B(p_i, 2^{-(j_0+1)})|/n$. By the choice of j_0, we have $|B(p_i, 2^{-(j_0+1)})| < 2^{j_0+1}$, and thus $q < 2^{j_0+1}/n \leq 2^{j-1}/n$.

The expected number of sample points that fall inside $B(p_i, 2^{-j})$ is $\mathbf{E}[N_j] = qK_j < \frac{c \log n}{2}$. Applying the Chernoff bound, we obtain

$$\mathbf{Pr}[\beta_j \geq 2^j] = \mathbf{Pr}[N_j \geq c \log n] \ < \ 1/poly(n) \ . \qquad \square$$

Lemma 5. *If $j \leq j_0 - 1$, then $\mathbf{Pr}[\beta_j \geq 2^j] > 1 - 1/poly(n)$.*

Proof. Since $j \leq j_0 - 1$, it follows that $|B(p_i, 2^{-j})| \geq |B(p_i, 2^{-j_0})| \geq 2^{j_0} \geq 2^{j+1}$. Let q be the probability that a randomly chosen sample point is in $B(p_i, 2^{-j})$. We have that $q \geq 2^{j+1}/n$.

The expected number of sample points that fall inside $B(p_i, 2^{-j})$ is $\mathbf{E}[N_j] = q\, K_j \geq 2\, c\, \log n$. Applying the Chernoff bound, we obtain

$$\mathbf{Pr}[\beta_j \geq 2^j] = \mathbf{Pr}[N_j \geq c\, \log n] \ > \ 1 - 1/poly(n) \ . \qquad \square$$

Lemma 6. *The described procedure estimates the value of r_i to within a constant factor in time $O(r_i\, n \log n)$, with high probability.*

Proof. Let j_0' be the estimated value of j_0. By Lemmas 4 and 5, it follows that with high probability, $j_0 \leq j_0' \leq j_0 + 1$. If we use the value $r_i' = 2^{-j_0'}$ as an estimation of r_i, then by Lemma 3 we obtain that $r_i/2 \leq r_i' \leq 4\,r_i$.

Moreover, with high probability, the running time of the procedure is at most $\sum_{j=j_0}^{\log n} O(K_j) = O(r_i\,n\,\log n)$. □

2.5 Estimating the Sum of the Radii

In this section we show how to estimate $\sum_i r_i$ in time almost linear in n. Let us first assume that we know the cost of the solution c, and we sample a set of s points independently and uniformly at random, where $s = \Theta(\frac{n}{c}\log n)$. Since by Lemma 6, the running time to estimate a radius r_i is $O(r_i\,n\,\log n)$, the total expected running time of the algorithm is

$$\mathbf{E}[\text{time}] = s \cdot \mathbf{E}[\text{one step}] = s \cdot O(\tfrac{1}{n} \cdot \sum_i r_i\,n\,\log n) = O(n \log^2 n)\ .$$

Let x_i, for $i \in \{1, 2, \ldots, s\}$, be the radii of the sample points taken by the algorithm. We have

$$\mathbf{E}[x_i] = \frac{\sum_j r_j}{n}\ .$$

Let $S = \sum_{i=1}^{s} x_i$ and hence, $\mathbf{E}[S] = \frac{s \cdot \sum_i r_i}{n} = \frac{\Theta(\frac{n}{c}\log n)\cdot\sum_i r_i}{n} = \Theta\Big(\frac{\sum_i r_i}{c} \cdot \log n\Big) = \Theta(\log n)$. Let $\epsilon > 0$ be arbitrary. Our goal is to use the value of S as the estimator of $\frac{n}{s}\sum_i r_i$. To show the quality of this estimator we will bound $\mathbf{Pr}[|S - \mathbf{E}[S]| \geq \epsilon \cdot \mathbf{E}[S]]$. By using the fact that $0 \leq x_i \leq 1$ for every i, we apply a variant of the Hoeffding inequality, see [11, Theorem 2.3], to obtain

$$\mathbf{Pr}[S \geq (1+\epsilon)\cdot\mathbf{E}[S]] \ \leq\ e^{-\frac{\epsilon^2\cdot\mathbf{E}[S]}{2(1+\epsilon/3)}}\ ,$$

$$\mathbf{Pr}[S \leq (1-\epsilon)\cdot\mathbf{E}[S]] \ \leq\ e^{-\frac{1}{2}\cdot\epsilon^2\cdot\mathbf{E}[S]}\ .$$

This immediately implies the following bound for any $0 < \epsilon \leq 1$,

$$\mathbf{Pr}[|S - \mathbf{E}[S]| \geq \epsilon\cdot\mathbf{E}[S]] \leq 2\,e^{-\Theta(\epsilon^2\cdot\mathbf{E}[S])} \ =\ 2\,e^{-\Theta(\epsilon^2\cdot\log n)}\ .$$

We now show how to remove the assumption that we know the cost of the solution. We run the algorithm in phases: we start in the first phase by "guessing" $c = n$, because we know that the cost of the optimal solution is not bigger than n. If $S < \frac{s}{n}\cdot c$, then we start a new phase with estimated cost $c/2$, and so on. If $S \geq \frac{s}{n}\cdot c$, we return $S \cdot n/s$ as the approximation of the cost. The probability that the algorithm ends in a bad phase (when S far away from $\frac{s}{n}\cdot c$) is low, because $\mathbf{Pr}[S \geq (1+\epsilon)\cdot\mathbf{E}[S]] < 1/poly(n)$, as shown above. Since we need to have at least one facility in a solution, we have $c \geq 1$, therefore we have at most a logarithmic number of phases.

Note that we only get a constant slowdown by running these phases to guess c, because the last phase, for the smallest c, dominates the running time of all the other phases. Thus we obtain the following theorem.

Theorem 1. *There exists a constant factor approximation algorithm for the uniform case of the Minimum Facility Location problem which runs in time $O(n \log^2 n)$ with high probability.*

3 Lower Bounds: Estimating the Cost in the General Case of the Uniform Minimum Facility Location Problem Requires $\Omega(n^2)$ Time (Even for Randomized Algorithms)

In this section, we consider a general case of the Minimum Facility Location problem in which we do not impose the restriction that $\mathcal{F} = P$ (that is, we allow only for a subset of points to be able to open a facility). We focus again on the uniform case, and the goal is to minimize the following cost:

$$\min_{F \subseteq \mathcal{F}} \left(|F| + \sum_{p \in P} d(p, F) \right) .$$

Our main result is the following theorem.

Theorem 2. *For any $\varrho \geq 1$, every approximation algorithm (even a randomized one) with approximation ratio ϱ for the cost of the Minimum Facility Location problem as defined above requires time $\Omega(n^2)$.*

Proof. We show the existence of two instances of the metric spaces which are undistinguishable by any $o(n^2)$-time algorithms and such that the cost of the

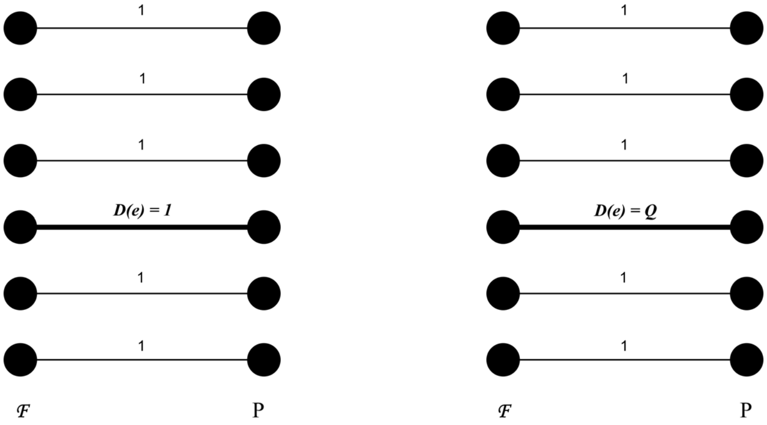

Fig. 1. Two metric spaces undistinguishable by any $o(n^2)$-time algorithms whose costs of the Minimum Facility Location differ by factor ϱ. The perfect matching connecting $\mathcal{F}$ with P is selected at random and the edge e is selected as a random edge from the matching. We set $Q = 2\,n\,(\varrho - 1) + 2$. The distances not shown are all equal to $n^3\,\varrho$

Minimum Facility Location in one instance is greater than ϱ times than the one in the other instance (see Fig. 1).

Let us consider the metric space with $2\,n$ points: n points in P and n points in $\mathcal{F}$. Take a random perfect matching $\mathbb{M}$ between the points in P and $\mathcal{F}$, and choose an edge $e \in \mathbb{M}$ at random. Now, we define the distances in $(P \cup \mathcal{F}, D)$ according to the following:

– for any $e^* \in \mathbb{M} \setminus \{e\}$, $D(e^*) = 1$,
– $D(e)$ is either 1 or $Q = 2\,n\,(\varrho - 1) + 2$, and
– for any pair of points x, y not connected by an edge from $\mathbb{M}$, $D(x, y) = n^3\,\varrho$.

It is easy to see that both instances define properly a metric space $(P \cup \mathcal{F}, D)$. Furthermore, that for such problem instances, the solution to the Minimum Facility Location will open all facilities and the cost of the Minimum Facility Location problem will depend on the choice of $D(e)$: if $D(e) = Q$ then the cost will be $2n - 1 + Q > 2\,n\,\varrho$, and if $D(e) = 1$, then the cost will be $2\,n$. Hence, any ϱ-factor approximation algorithm for the matching problem must distinguish between these two problem instances. However, this requires to find if there is an edge of length Q, and this is known to require time $\Omega(n^2)$, even if a randomized algorithm is used. □

3.1 Extensions

It is not difficult to see that almost an identical proof will also work for estimating the cost of *minimum-cost matching*, the cost of *minimum-cost bi-chromatic matching*, and also the cost of *k-median* for $k = n/2$; all these problems require $\Omega(n^2)$ to estimate the cost of their optimal solution to within any factor. No such lower bounds have been previously known.

Theorem 3. *For any $\varrho \geq 1$, every approximation algorithm (even a randomized one) with approximation ratio ϱ for each of the following problems requires time $\Omega(n^2)$:*

– *estimating the cost of minimum-cost matching for a set of n points in a metric space,*
– *estimating the cost of minimum-cost bi-chromatic matching for a set of n points in a metric space,*
– *estimating the cost of metric k-median for $k = n/2$.*

References

1. M. Charikar and S. Guha. Improved combinatorial algorithms for the facility location and k-median problems. *Proceedings of the 40th IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 378–388, 1999.
2. B. Chazelle, R. Rubinfeld, and L. Trevisan. Approximating the minimum spanning tree weight in sublinear time. *Proceedings of the 28th Annual International Colloquium on Automata, Languages and Programming (ICALP)*, pp. 190–200, 2001.

3. F. A. Chudak. Improved approximation algorithms for uncapacitated facility location. *Proceedings of the 6th International Integer Programming and Combinatorial Optimization Conference (IPCO)*, pp. 180–194, 1998.
4. A. Czumaj and C. Sohler. Estimating the weight of metric minimum spanning trees in sublinear time. *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 175–183, 2004.
5. S. Guha and S. Khuller. Greedy strikes back: Improved facility location algorithms. *Journal of Algorithms*, 31(1): 228–248, 1999.
6. P. Indyk. Sublinear time algorithms for metric space problems. *Proceedings of the 31st Annual ACM Symposium on Theory of Computing (STOC)*, pp. 428–434, 1999.
7. P. Indyk. A sublinear time approximation scheme for clustering in metric spaces. *Proceedings of the 40th IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 154–159, 1999.
8. K. Jain, M. Mahdian, and A. Saberi. A new greedy approach for facility location problems. *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 731–740, 2002.
9. K. Jain and V. Vazirani. Approximaton algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *Journal of the ACM*, 48(2): 274–296, 2001.
10. M. Mahdian, Y. Ye, and J. Zhang. Improved approximation algorithms for metric facility location problems. *Proceedings of the 5th International Workshop on Approximation Algorithms for Combinatorial Optimization(APPROX)*, pp. 229–242, 2002.
11. C. McDiarmid. Concentration. In M. Habib, C. McDiarmid, J. Ramirez-Alfonsin, and B. Reed, editors, *Probabilistic Methods for Algorithmic Discrete Mathematics*, Algorithms and Combinatorics, pp. 195–247. Springer-Verlag, Berlin, 1998.
12. R. R. Mettu and C. G. Plaxton. The online median problem. *SIAM Journal on Computing*, 32(3): 816–832, 2003.
13. D. B. Shmoys, E. Tardos, and K. Aardal. Approximation algorithms for facility location problems. *Proceedings of the 29th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 265–274, 1997.
14. M. Thorup. Quick k-median, k-center, and facility location for sparse graphs. *SIAM Journal on Computing*, 34(2):405–432, 2005.