A Primal-Dual Algorithm for Online Non-Uniform Facility Location

Dimitris Fotakis

Department of Information and Communication Systems Engineering, University of the Aegean
83200 Karlovasi, Samos, Greece. Email: fotakis@aegean.gr

Abstract. In the online version of Facility Location, the demand points arrive one at a time and must be irrevocably assigned to an open facility upon arrival. The objective is to minimize the sum of facility and assignment costs. We consider the general case of non-uniform facility costs and present a simple primal-dual deterministic algorithm. We prove that its competitive ratio is no greater than $4\log(n+1)+2$, which is close to the lower bound of $\Theta(\frac{\log n}{\log\log n})$.

1 Introduction

In the (metric uncapacitated) Facility Location problem, we are given a metric space along with a facility cost for each point and a (multi)set of demand points, and we seek for a set of facility locations which minimize the sum of facility and assignment costs. Facility Location provides a simple and natural model for network design and data clustering problems and has been the subject of intensive research over the last decade (e.g., see [12] for a survey and [6] for approximation algorithms and applications). In addition to the offline setting, there are many practical applications where either the demand points are not known in advance or the solution must be constructed incrementally using limited (if any) information about future demands (e.g., see [11] for some examples from the areas of network design and data clustering).

The definition of *Online Facility Location* [11] is motivated by similar considerations. In Online Facility Location, the demand points arrive one at a time and must be irrevocably assigned to either an existing or a new facility upon arrival. The objective is to minimize the sum of facility and assignment costs, where the assignment cost of a demand is the distance from the facility to which the demand is assigned.

Related Work. In the offline setting, where the demand points are fully known in advance, there are constant factor approximation algorithms based on Linear Programming rounding (e.g., [13, 14]), local search (e.g., [4, 2]), and the primal-dual method (e.g., [9, 8]). The best known polynomial-time algorithm achieves an approximation ratio of 1.52 [10], while no polynomial-time algorithm can have an approximation ratio less than 1.463 unless $\mathrm{NP} = \mathrm{DTIME}(n^{\mathrm{O}(\log\log n)})$ [7].

Meyerson [11] introduced the online version of Facility Location and presented a randomized algorithm which achieves a constant performance ratio if the demands are examined in random order. In the standard framework of competitive analysis (e.g., [3]), where both the set of demands and their arrival order are selected by an oblivious adversary, Meyerson's algorithm achieves a competitive ratio of $\mathrm{O}(\log n)$. Meyerson also proved a lower bound of $\omega(1)$ on the competitive ratio of any online algorithm.

In previous work [5], we improved the lower bound to $\Omega(\frac{\log n}{\log\log n})$ even for the special case that the metric space is a line segment and the facility costs are uniform. We also presented a deterministic algorithm with an (asymptotically) optimal competitive ratio of $\Theta(\frac{\log n}{\log\log n})$ for general metric spaces and non-uniform facility costs. The algorithm is rather complicated to formulate and implement. Its analysis exploits locality in a highly non-trivial way and is lengthy and quite technical. Hence despite being a contribution towards understanding the problem from a theoretical point of view, the algorithm of [5] is of limited practical applicability.

This observation motivated a recent work by Anagnostopoulos et al. [1], who presented a simple $\Theta(2^d \log n)$-competitive deterministic algorithm for Online Facility Location in d-dimensional Euclidean spaces. The algorithm works only for the fixed-location uniform-cost model, where the facilities have the same opening cost and can be located at a fixed subset of points. For the Euclidean plane, the algorithm is quite fast and its performance in practice is comparable to the performance of the algorithms in [11, 5]. Nevertheless, the approach of [1] cannot be applied to general metric spaces and the general case of non-uniform facility costs.

Contribution. We present a new deterministic algorithm for Online Facility Location. The algorithm can be described as a primal-dual online algorithm and works for every metric space and non-uniform facility costs (e.g., see [15] for an introduction to primal-dual approximation algorithms).

The algorithm maintains the invariant that the distances of each demand to the nearest algorithm's facility constitute a feasible solution to the dual of a natural Linear Programming relaxation for Facility Location. Every time a dual constraint is violated due to the arrival of a new demand, the algorithm opens a new facility at the location corresponding to the most violated dual constraint, and the dual feasibility is restored.

We prove that the algorithm's competitive ratio is no greater than $4\log(n+1)+2$, which is quite close to the lower bound of $\Theta(\frac{\log n}{\log\log n})$ [5]. The analysis is simple and gives new insight into the behaviour of previously known algorithms for Online Facility Location and related problems.

Our algorithm is the first deterministic algorithm which works for every metric space and the general case of non-uniform facility costs and is simple to formulate and analyze, easy to implement, and quite fast in terms of running time.

The only previously known deterministic algorithm which works for every metric space and non-uniform facility costs is the algorithm of [5]. That algorithm achieves an (asymptotically) optimal competitive ratio of $\Theta(\frac{\log n}{\log\log n})$, but is quite complicated to formulate, implement, and analyze. Moreover, since there are quite large multiplicative and additive constants involved in its competitive ratio, we expect that the algorithm presented in this paper will outperform the algorithm of [5] in practice.

The algorithm of [1] only works for Euclidean spaces of small dimension and uniform facility costs. In addition, its competitive ratio is exponential on the dimension of the metric space. Hence, our results are not comparable to the results of [1].

Problem Definition and Notation. We evaluate the performance of online algorithms using *competitive analysis* (e.g., [3]). An online algorithm is c-competitive if for all instances, the cost incurred by the algorithm is at most c times the cost incurred by an optimal offline algorithm, which knows the demand sequence in advance.

In Online Facility Location, we are given a metric space $\mathcal{M} = (M, d)$, where M denotes the set of points and $d : M \times M \mapsto \mathbb{R}_+$ denotes the distance function which is non-negative, symmetric and satisfies the triangle inequality. For each point $w \in M$, we are also given the cost f_w of opening a facility at w. The facility costs of some points may be infinite, in which case we are not allowed

$F \leftarrow \emptyset$; $L \leftarrow \emptyset$; initializePotentials();	initializePotentials()
for each new demand u:	for all $z \in M$ do
$L \leftarrow L \cup \{u\}$;	$p(z) \leftarrow 0$;
updatePotentials(F, u);	updatePotentials(F, u)
$w \leftarrow \arg\max_{z \in M}\{p(z) - f_z\}$;	for all $z \in M$ do
if $p(w) - f_w > 0$ then	$p(z) \leftarrow p(z) + d(F, u) \dot{-} d(z, u)$;
$F \leftarrow F \cup \{w\}$;	computeNewPotentials(F, L)
computeNewPotentials(F, L);	for all $z \in M$ do
assign u to the nearest facility in F;	$p(z) \leftarrow \sum_{v \in L} d(F, v) \dot{-} d(z, v)$;

Fig. 1. The algorithm Simple Non-Uniform Facility Location – SNFL .

to open a facility there. We slightly abuse the notation by letting the same symbol denote both a demand/facility and the corresponding point of the metric space.

The demand sequence consists of (not necessarily distinct) points $u \in M$. When a new demand u arrives, the algorithm can open some new facilities, in which case it incurs the corresponding opening cost. The decision of opening a facility at a particular location is *irrevocable*. Then, u is *irrevocably* assigned to the nearest facility. If u is assigned to a facility at w, u's assignment cost is $d(w, u)$. The objective is to minimize the sum of facility and assignment costs.

We only consider unit demands and allow multiple demands to be located at the same point. We use n to denote the total number of demands. We consider the general case of *non-uniform* facility costs, where the opening costs depend on the location and there are no restrictions on them.

A metric space $\mathcal{M} = (M, d)$ is usually identified by its point set M. For a point $u \in M$ and a subset of points $M' \subseteq M$, $d(M', u) \equiv \min_{v \in M'}\{d(v, u)\}$ denotes the distance between u and the nearest point in M'. We let $d(\emptyset, u) = \infty$. We use $\log x$ to denote the base-2 logarithm of a positive number x. For every numbers x, y, let $x \dot{-} y \equiv \max\{x - y, 0\}$. In the technical part of the paper, we repeatedly use $x \dot{-} y \geq x - y$.

Organization. In Section 2, we formulate the algorithm and discuss the intuition behind it. We also state our main result, namely that the algorithm's competitive ratio is no greater than $4\log(n+1)+2$, and outline its proof. The main invariant maintained by the algorithm is formally proven in Section 3. Using this invariant, we bound the algorithm's assignment cost in Section 4 and the algorithm's facility cost in Section 5. The conclusions and some directions for further research are discussed in Section 6.

2 The Primal-Dual Algorithm

The algorithm *Simple Non-Uniform Facility Location* (SNFL, Fig. 1) maintains the *set of demands* considered so far, denoted L, its *facility configuration*, denoted F, and the *potential* of each point z, denoted $p(z)$. At any point in time, $p(z) = \sum_{v \in L} d(F, v) \dot{-} d(z, v)$ for all $z \in M$. The algorithm maintains the invariant that the potential of each point z does not exceed the cost of opening a facility at z, namely $p(z) \leq f_z$. Every time this invariant is violated, the algorithm opens a new facility at an appropriately selected location and the invariant is restored.

Initially, $L = \emptyset$, $F = \emptyset$, and $p(z) = 0$ for all $z \in M$. When a new demand arrives, the algorithm updates the potentials of all points and computes the point w of maximum $p(w) - f_w$. If $p(w) > f_w$, the algorithm opens a new facility at w and computes the potentials again according to the new facility configuration. Due to w's choice, opening a new facility at w restores the invariant $p(z) \leq f_z$ for all points z (cf. Lemma 1). If $p(w) \leq f_w$, the invariant holds for all points and no new facilities open. Finally, the new demand is assigned to the nearest facility in the updated facility configuration.

Intuition. Before establishing the competitive ratio, we give an intuitive description and a primal-dual interpretation of SNFL. We regard the distance of each demand $v \in L$ to the nearest algorithm's facility, namely the distance $d(F, v)$, as a credit currently held by v. The demands contribute their credit towards opening new facilities closer to them. More specifically, when a new facility w opens, the credit of each demand v becomes $d(F \cup \{w\}, v) \leq d(F, v)$. The difference $d(F, v) - d(F \cup \{w\}, v) = d(F, v) \dot{-} d(w, v)$ is regarded as the contribution of v to w's opening cost.

The potential of each point z corresponds to the decrease in the total demands' credit if a facility at z opens. The algorithm opens a new facility only if the decrease in the demands' credit exceeds the corresponding opening cost. In simple words, a new facility opens only if the demands are willing to spend an adequate amount of credit for it. Moreover, the location of the new facility is the point maximizing the difference between the credit spent for the new facility and the actual opening cost.

We formalize the intuitive description above by resorting to Linear Programming duality. The offline version of Facility Location is formulated as the following $0-1$ Integer Program:

$$
\begin{array}{llll}
\min \sum_{z \in M} f_z y_z + \sum_{z \in M} \sum_{v \in L} x_{zv} d(z, v) & & & \\
\text{s.t} \quad \sum_{z \in M} x_{zv} = 1 & \forall v \in L & & \\
\qquad x_{zv} \leq y_z & \forall z \in M, \ \forall v \in L & & \text{(IP)} \\
\qquad y_z \in \{0,1\}, x_{zv} \in \{0,1\} & \forall z \in M, \ \forall v \in L & &
\end{array}
$$

Setting the variable y_z to 1 corresponds to opening a facility at z and setting the variable x_{zv} to 1 corresponds to assigning demand v to facility z. We obtain a Linear Programming relaxation of (IP) by replacing the $0-1$ constraints with $y_z \geq 0$ and $x_{zv} \geq 0$ respectively. The dual of the Linear Programming relaxation is:

$$
\begin{array}{llll}
 & \max \sum_{v \in L} c_v & & \\
\text{s.t} & \sum_{v \in L} [c_v \dot{-} d(z, v)] \leq f_z & \forall z \in M & \text{(DP)} \\
 & c_v \geq 0 & \forall v \in L &
\end{array}
$$

It becomes clear that SNFL is a primal-dual online algorithm. It maintains its facility configuration F so as the distances $d(F, v)$, $v \in L$, (in other words, the demand credits) to constitute a feasible solution to (DP). Every time a dual constraint is violated due to the arrival of a new demand, a new facility opens at the location corresponding to the most violated dual constraint, and the dual feasibility is restored.

The running time of SNFL is $\mathrm{O}(n\,|M|\,|F|)$, where n is the number of demands, $|M|$ is the number of points with finite facility cost, and $|F|$ is the number of facilities opened by the algorithm. The remaining part of the paper is devoted to the proof of the algorithm's competitive ratio.

Theorem 1. *The competitive ratio of* SNFL *is no greater than* $4 \log(n+1) + 2$.

2.1 Preliminaries

For an arbitrary fixed sequence of n demands, we compare the algorithm's cost with the cost of the offline optimal solution. We denote the set of optimal facilities by F^*. To avoid confusing the algorithm's facilities with the optimal facilities, we use the term *optimal center*, or simply *center*, to refer to an optimal facility in F^* and the term *facility* to refer to an algorithm's facility in F.

We let the optimal solution F^* consist of k centers $c_1, c_2, \ldots, c_k$. We use f_{c_i}, $i = 1, \ldots, k$, to denote the opening costs of the optimal centers. In the optimal solution, each demand is assigned to the nearest center in F^*. Hence, F^* partitions the demand sequence into optimal clusters $C_1, C_2, \ldots, C_k$. For each demand v, let $d_v^* \equiv d(F^*, v)$ denote the assignment cost of v in the optimal solution, let $\mathrm{Asg}^* \equiv \sum_v d_v^*$ denote the total optimal assignment cost, and let $\mathrm{Fac}^* \equiv \sum_{c_i \in F^*} f_{c_i}$ denote the total optimal facility cost. The total optimal cost is $\mathrm{Fac}^* + \mathrm{Asg}^*$. For any demand set C, let $\mathrm{Asg}^*(C) \equiv \sum_{v \in C} d_v^*$ denote the total optimal assignment cost for the demands in C.

We usually distinguish between the arrival and the assignment time of a new demand because the algorithm's configuration may have changed in between. We use the convention that unprimed symbols refer to the algorithm's configuration just before a new demand arrives and primed symbols refer to the updated algorithm's configuration at the demand's assignment time.

2.2 Outline of the Analysis

We start by showing that SNFL maintains the invariant that $p(z) \leq f_z$ for all $z \in M$ (Lemma 1). Therefore, after j demands from cluster C_i have been considered, there is a facility within a distance of $\frac{1}{j}[f_{c_i} + 2\mathrm{Asg}^*(C_i)]$ from the optimal center c_i (Corollary 1). This implies that the algorithm's assignment cost is within a logarithmic factor of the total optimal cost (Lemma 2).

We allocate a credit of $c(u) = \min\{d(F, u), \min_{z \in M}\{f_z - p(z) + d(z, u)\}\}$ to each new demand u, where F denotes the facility configuration and $p(z)$ denotes z's potential just before u arrives. We show that the algorithm's facility cost never exceeds the total credit allocated to the demands in L (Lemma 4). Corollary 1 implies that the total credit is within a logarithmic factor of the total optimal cost (Lemma 5).

3 Basic Properties

We start by establishing the algorithm's main invariant.

Lemma 1 (Main Invariant). *Let L be the demand set, and let F be the facility configuration just before a new demand arrives. For all $z \in M$, $p(z) \leq f_z$.*

Proof. We prove the lemma by induction on the number of demands considered by the algorithm. The invariant holds before the first demand arrives because initially $p(z) = 0 \leq f_z$ for all $z \in M$. We inductively assume that the invariant holds just before a new demand u arrives and prove that the invariant holds after u's assignment.

Let F be the algorithm's facility configuration, let L be the demand set, and let $p(z) = \sum_{v \in L} d(F, v) \dot{-} d(z, v) \leq f_z$ be the potential of each point z just before u arrives. We denote by $p'(z)$ and $p''(z)$ the potential of each point z when w is computed and at u's assignment time respectively. We show that for all $z \in M$, $p''(z) \leq f_z$.

The new demand does not open a new facility only if $p'(z) \leq f_z$ for all $z \in M$. Then the facility configuration remains the same and $p''(z) = p'(z) \leq f_z$.

If a new facility opens at w, let $F' = F \cup \{w\}$ denote the facility configuration at u's assignment time. To reach a contradiction, we assume that after w opens and the new potentials are computed, there is a point $\hat{w}$ with $p''(\hat{w}) > f_{\hat{w}}$.

We first observe that for every $z \in M$,

$$p(z) = \textstyle\sum_{v \in L}[d(F, v) \dot{-} d(z, v)] \geq \sum_{v \in L}[d(F', v) \dot{-} d(z, v)] \tag{1}$$

because $F \subseteq F'$ and $d(F', v) \leq d(F, v)$ for all demands $v \in L$.

We claim that w is closer to u than any facility in F. The claim follows from the following inequalities:

$$\begin{aligned} 0 < p'(w) - f_w &= d(F, u) \dot{-} d(w, u) + p(w) - f_w \\ &\leq d(F, u) \dot{-} d(w, u) = d(F, u) - d(w, u) \end{aligned} \tag{2}$$

The first inequality holds because a new facility at w opens. The next equality follows from $p'(w) = p(w) + d(F, u) \dot{-} d(w, u)$. The next inequality uses $p(w) - f_w \leq 0$ by inductive hypothesis. Finally, we observe that $d(F, u) \dot{-} d(w, u) > 0$.

The point $\hat{w}$ violating the invariant after w opens should be even closer to u:

$$\begin{aligned} 0 < p''(\hat{w}) - f_{\hat{w}} &= d(F', u) \dot{-} d(\hat{w}, u) + \textstyle\sum_{v \in L}[d(F', v) \dot{-} d(\hat{w}, v)] - f_{\hat{w}} \\ &\leq d(w, u) \dot{-} d(\hat{w}, u) + p(\hat{w}) - f_{\hat{w}} \\ &\leq d(w, u) \dot{-} d(\hat{w}, u) = d(w, u) - d(\hat{w}, u) \end{aligned}$$

The first equality follows from the definition of $p''(\hat{w})$. The next inequality uses Ineq. (1). The last inequality holds because $p(\hat{w}) - f_{\hat{w}} \leq 0$ by inductive hypothesis. Finally, we observe that $d(w, u) \dot{-} d(\hat{w}, u) > 0$.

After the potentials of all points have been updated, the algorithm selects w instead of $\hat{w}$ as the location of the new facility. Therefore, $p'(w) - f_w \geq p'(\hat{w}) - f_{\hat{w}}$ and:

$$\begin{aligned} d(F, u) - d(w, u) \geq p'(w) - f_w &\geq p'(\hat{w}) - f_{\hat{w}} \\ &= d(F, u) \dot{-} d(\hat{w}, u) + p(\hat{w}) - f_{\hat{w}} \\ &\geq d(F, u) - d(\hat{w}, u) + \textstyle\sum_{v \in L}[d(F', v) \dot{-} d(\hat{w}, v)] - f_{\hat{w}} \end{aligned}$$

The first inequality follows from Ineq. (2). The equality follows from the definition of $p'(\hat{w})$. For the last inequality, we use $d(F, u) > d(\hat{w}, u)$ and Ineq. (1). Therefore,

$$0 \geq d(w, u) - d(\hat{w}, u) + \textstyle\sum_{v \in L}[d(F', v) \dot{-} d(\hat{w}, v)] - f_{\hat{w}} = p''(\hat{w}) - f_{\hat{w}} \tag{3}$$

We use $d(F, u) > d(w, u) > d(\hat{w}, u)$. Hence, $d(F', u) = d(w, u)$ and $d(w, u) - d(\hat{w}, u) = d(F', u) \dot{-} d(\hat{w}, u)$. The equality follows from the definition of $p''(\hat{w})$.

Ineq. (3) contradicts to the choice of $\hat{w}$ as a point with $p''(\hat{w}) > f_{\hat{w}}$. Consequently, after w opens, there is no point violating the invariant. □

The invariant of Lemma 1 implies that the algorithm's facility configuration converges fast to the optimal centers as more and more demands are considered.

Corollary 1. *Let L be the demand set, and let F be the facility configuration after all demands in L have been considered. For each optimal cluster C_i with center c_i,*

$$|L \cap C_i|\, d(F, c_i) \leq f_{c_i} + 2\,\mathrm{Asg}^*(C_i)$$

Proof. We apply the invariant of Lemma 1 for the optimal center c_i:

$$\begin{aligned} f_{c_i} \geq p(c_i) = \sum_{v \in L} d(F, v) \dot{-} d(c_i, v) &\geq \sum_{v \in L \cap C_i} d(F, v) \dot{-} d^*_v \\ \geq \sum_{v \in L \cap C_i} [(d(F, c_i) - d^*_v) - d^*_v] &= |L \cap C_i|\, d(F, c_i) - 2 \sum_{v \in L \cap C_i} d^*_v \end{aligned}$$

The second inequality holds because the first sum consists of non-negative terms. In addition, for all $v \in C_i$, $d(c_i, v) = d^*_v$. For the third inequality, we apply the triangle inequality and $x \dot{-} y \geq x - y$. Finally, we observe that $\mathrm{Asg}^*(L \cap C_i) \leq \mathrm{Asg}^*(C_i)$. □

4 Assignment Cost

Then we use Corollary 1 and bound the algorithm's assignment cost.

Lemma 2. *The algorithm's assignment cost is no greater than*

$$\log(n+1)\mathrm{Fac}^* + (2\log(n+1) + 1)\mathrm{Asg}^*$$

Proof. Let C_i be an optimal cluster with center c_i, let $n_i \equiv |C_i|$ be the number of demands in C_i, and let $u_1, u_2, \ldots, u_{n_i}$, be the demands of C_i in the order considered by the algorithm. For each demand u_j, let F'_{u_j} be the facility configuration at u_j's assignment time. Hence, the algorithm's assignment cost for u_j is:

$$d(F'_{u_j}, u_j) \leq d(F'_{u_j}, c_i) + d^*_{u_j} \leq \tfrac{1}{j}[f_{c_i} + 2\mathrm{Asg}^*(C_i)] + d^*_{u_j} \tag{4}$$

We first apply the triangle inequality. For the second inequality, we use Corollary 1 with $L \cap C_i = \{u_1, \ldots, u_j\}$ since the algorithm has considered the first j demands from C_i and reached the facility configuration F'_{u_j}.

Summing up Ineq. (4) for all $j = 1, \ldots, n_i$, we conclude that the algorithm's assignment cost for the demands in C_i is:

$$\begin{aligned} \sum_{j=1}^{n_i} d(F'_{u_j}, u_j) &\leq [f_{c_i} + 2\mathrm{Asg}^*(C_i)] \sum_{j=1}^{n_i} \tfrac{1}{j} + \sum_{j=1}^{n_i} d^*_{u_j} \\ &\leq \log(n_i + 1) f_{c_i} + (2\log(n_i+1) + 1)\mathrm{Asg}^*(C_i) \end{aligned}$$

For the last inequality, we use $\sum_{j=1}^{n} 1/j \leq \log(n+1)$ for all $n \in \mathbb{N}$. The lemma follows by summing up the inequality above over all optimal clusters. □

5 Facility Cost

We allocate a credit of $c(u) = \min\{d(F,u), \min_{z \in M}\{f_z - p(z) + d(z,u)\}\}$ to each new demand u, where F denotes the facility configuration and $p(z)$ denotes z's potential just before u arrives.

Lemma 3. *For each new demand u, $c(u) = f_w - p(w) + d(w,u)$ if u opens a new facility at w, and $c(u) = d(F,u)$ otherwise.*

Proof. Let F be the facility configuration, and let $p(z)$ be the potential of each point z just before u arrives. Let also $p'(z) = p(z) + d(F,u) \dot{-} d(z,u)$ denote the updated potential of each point z.

Let the new demand u open a new facility at w. Due to the choice of w, $p'(w) - f_w \geq p'(z) - f_z$ for all points $z \in M$. Therefore,

$$\begin{aligned} p'(w) - f_w &= p(w) + d(F,u) - d(w,u) - f_w \\ &\geq p(z) + d(F,u) \dot{-} d(z,u) - f_z \geq p(z) + d(F,u) - d(z,u) - f_z \end{aligned}$$

The equality holds because w is closer to u than any facility in F (see also Ineq. (2) in the proof of Lemma 1). Hence, w is a minimizer of $f_z - p(z) + d(z,u)$, and $f_w - p(w) + d(w,u) = \min_{z \in M}\{f_z - p(z) + d(z,u)\}$. Since a new facility opens, $p'(w) - f_w > 0$ and $d(F,u) > f_w - p(w) + d(w,u)$. Consequently, if the new demand u opens a new facility at w, u's credit is $c(u) = f_w - p(w) + d(w,u)$.

If the new demand u does not open a new facility, for all points $z \in M$,

$$f_z \geq p'(z) = p(z) + d(F,u) \dot{-} d(z,u) \geq p(z) + d(F,u) - d(z,u)$$

Therefore, $d(F,u) \leq f_z - p(z) + d(z,u)$, and u's credit is $c(u) = d(F,u)$. □

Using Lemma 3, we show that the cost for the facilities in F does not exceed the total credit allocated to the demands in L.

Lemma 4. *Let L be the demand set, and let F be the facility configuration after all demands in L have been considered. Then, $\sum_{w \in F} f_w \leq \sum_{v \in L} c(v)$.*

Proof. We prove the lemma by a potential function argument. We define the potential function $\Phi = \sum_{v \in L} d(F,v)$ and calculate the change $\Delta\Phi$ in the value of the potential function when a new demand u is considered. Let L be the demand set, let F be the facility configuration, and let $p(z)$ be the potential of each point z just before u arrives.

If u does not open a new facility, then $\Delta\Phi = d(F,u) = c(u)$, by Lemma 3.

If u opens a new facility at w, let $F' = F \cup \{w\}$. By definition, $d(F,v) - d(F',v) = d(F,v) \dot{-} d(w,v)$ for all demands v. Therefore,

$$\begin{aligned} \Delta\Phi &= d(w,u) - \textstyle\sum_{v \in L}[d(F,v) - d(F',v)] \\ &= d(w,u) - \textstyle\sum_{v \in L}[d(F,v) \dot{-} d(w,v)] = d(w,u) - p(w) \end{aligned}$$

By Lemma 3, u's credit is $c(u) = f_w - p(w) + d(w,u)$, and $\Delta\Phi + f_w = c(u)$.

Thus, we have shown that the credit allocated to each new demand u is equal to the change in the value of Φ plus the cost for the facility opened by u (possibly none). Initially, $\Phi = 0$ because the demand set is empty. By definition, Φ remains non-negative throughout the execution of the algorithm. Therefore, the opening cost for the facilities in F never exceeds the total credit allocated to the demands in L. □

Finally, we use Corollary 1 and bound the total credit allocated to the demand set.

Lemma 5. *Let L be the demand set. Then,*

$$\sum_{v \in L} c(v) \le (\log n + 1)\mathrm{Fac}^* + (2\log n + 1)\mathrm{Asg}^*$$

Proof. As in the proof of Lemma 2, let C_i be an optimal cluster with center c_i, let $n_i \equiv |C_i|$, and let $u_1, u_2, \ldots, u_{n_i}$, be the demands of C_i in the order considered by the algorithm. For each demand u_j, let F_{u_j} be the facility configuration just before u_j arrives.

The credit of each demand u_j is $c(u_j) \le \min\{d(F_{u_j}, u_j), f_{c_i} + d^*_{u_j}\}$ because the potential of c_i is non-negative. For the first demand, we use $c(u_1) \le f_{c_i} + d^*_{u_1}$. For each of the remaining demands $u_j, j = 2, \ldots, n_i$, we use

$$c(u_j) \le d(F_{u_j}, u_j) \le d(F_{u_j}, c_i) + d^*_{u_j} \le \tfrac{1}{j-1}[f_{c_i} + 2\mathrm{Asg}^*(C_i)] + d^*_{u_j} \tag{5}$$

For the second inequality, we use Corollary 1 with $L \cap C_i = \{u_1, \ldots, u_{j-1}\}$ since the algorithm has considered the first $j-1$ demands from C_i just before u_j arrives.

Summing up the above inequalities for all $j = 1, \ldots, n_i$, we conclude that the total credit allocated to the demands in C_i is:

$$\begin{aligned}\sum_{j=1}^{n_i} c(u_j) &\le f_{c_i} + [f_{c_i} + 2\mathrm{Asg}^*(C_i)] \sum_{j=2}^{n_i} \tfrac{1}{j-1} + \sum_{j=1}^{n_i} d^*_{u_j} \\ &\le (\log n_i + 1) f_{c_i} + (2\log n_i + 1)\mathrm{Asg}^*(C_i)\end{aligned}$$

For the last inequality, we use $\sum_{j=1}^{n-1} 1/j \le \log n$ for all $n \in \mathbb{N}$. The lemma follows by summing up the previous inequality for all optimal clusters. $\square$

Lemmas 4 and 5 imply that the facility cost is no greater than $(\log n + 1)\mathrm{Fac}^* + (2\log n + 1)\mathrm{Asg}^*$. Combining this bound with the bound of Lemma 2, we obtain that the total algorithm's cost is no greater than

$$(2\log(n+1) + 1)\mathrm{Fac}^* + (4\log(n+1) + 2)\mathrm{Asg}^* \le (4\log(n+1) + 2)(\mathrm{Fac}^* + \mathrm{Asg}^*)$$

This concludes the proof of Theorem 1.

6 Conclusions

We presented a simple and practical primal-dual algorithm for Online Facility Location with non-uniform facility costs. The algorithm maintains the invariant that the distances of each demand to the nearest facility constitute a feasible dual solution. Every time a dual constraint is violated due to the arrival of a new demand, a new facility opens at the location corresponding to the most violated dual constraint, and the dual feasibility is restored. We prove that the algorithm's competitive ratio is $4\log(n+1) + 2$, which is close to the lower bound of $\Theta(\frac{\log n}{\log\log n})$.

We are not aware of any examples establishing that the algorithm's competitive ratio is $\Omega(\log n)$. Thus, it remains open whether the competitive ratio of SNFL is $\Theta(\log n)$ or can be improved by a more careful analysis.

In retrospect, we observe that all known algorithms for Online Facility Location can be regarded as maintaining a similar invariant *implicitly*. More specifically, all known algorithms have the property that the distances of each demand to the nearest facility form an *approximately* feasible dual solution. For instance in the deterministic algorithms of [5, 1], the decision of opening a facility at a particular location only depends on the number of "unsatisfied" demands in its neighborhood. The crucial observation is that these demands contribute more than a constant fraction of each location's potential.

Instead of considering many different neighborhoods and distinguishing between "satisfied" and "unsatisfied" demands, the deterministic algorithm presented in this paper treats all demands uniformly and regards the entire metric space as a single neighborhood. This makes the analysis easier and more intuitive, and the algorithm simpler, faster, and more practical.

In the light of this observation, it is natural to ask whether there is a generic primal-dual framework for formulating and establishing the competitive ratios of all known algorithms for Online Facility Location and related problems.

References

1. A. Anagnostopoulos, R. Bent, E. Upfal, and P. Van Hentenryck. A Simple and Deterministic Competitive Algorithm for Online Facility Location. *Information and Computation*, 194:175–202, 2004.
2. V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit. Local Search Heuristics for k-Median and Facility Location Problems. *Proc. of STOC '01*, pp. 21–29, 2001.
3. A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.
4. M. Charikar and S. Guha. Improved Combinatorial Algorithms for the Facility Location and k-Median Problems. *Proc. of FOCS '99*, pp. 378–388, 1999.
5. D. Fotakis. On the Competitive Ratio for Online Facility Location. *Proc. of ICALP '03*, LNCS 2719, pp. 637–652, 2003.
6. S. Guha. *Approximation Algorithms for Facility Location Problems*. PhD thesis, Stanford University, 2000.
7. S. Guha and S. Khuller. Greedy Strikes Back: Improved Facility Location Algorithms. *Proc. of SODA '98*, pp. 649–657, 1998.
8. K. Jain, M. Mahdian, and A. Saberi. A New Greedy Approach for Facility Location Problems. *Proc. of STOC '02*, pp. 731–740, 2002.
9. K. Jain and V. Vazirani. Approximation Algorithms for Metric Facility Location and k-Median Problems Using the Primal-Dual Schema and Lagrangian Relaxation. *J. of the ACM*, 48(2):274–296, 2001.
10. M. Mahdian, Y. Ye, and J. Zhang. Improved Approximation Algorithms for Metric Facility Location Problems. *Proc. of APPROX '02*, LNCS 2462, pp. 229–242, 2002.
11. A. Meyerson. Online Facility Location. *Proc. of FOCS '01*, pp. 426–431, 2001.
12. D. Shmoys. Approximation Algorithms for Facility Location Problems. *Proc. of APPROX '00*, LNCS 1913, pp. 27–33, 2000.
13. D. Shmoys, E. Tardos, and K. Aardal. Approximation Algorithms for Facility Location Problems. *Proc. of STOC '97*, pp. 265–274, 1997.
14. M. Sviridenko. An Improved Approximation Algorithm for the Metric Uncapacitated Facility Location Problem. *Proc. of IPCO '02*, LNCS 2337, pp. 240–257, 2002.
15. V. Vazirani. *Approximation Algorithms*. Springer, 2001.