Online Facility Location with Recurring Maximum Demand

Jourdain Lamperski[†*] Luca Wrabetz[†] Oleg A. Prokopyev[‡]

Abstract

We study an *online facility location problem* in which the demand points arrive at discrete times. There is a fixed cost for locating a facility, and there is a variable cost at each time for serving points that have arrived up to that time. The considered variable cost reflects *recurring maximum demand* and is defined as the maximum distance between the points and their nearest facilities. The existing online location/clustering studies on *one-time* demand (i.e., each point is served only at the time it arrives) demonstrate that constant-factor competitive guarantees are not attainable unless facilities/centers can be relocated/updated (or other simplifying assumptions hold). While centers can often be updated in clustering applications, facilities typically cannot be relocated in many practically relevant settings. In this work we introduce and study the *Cumulative Variable Cost Thresholding* (CCT) algorithm. CCT irrevocably locates facilities at points that are farthest from an existing facility when the cumulative variable cost exceeds a specified threshold. As our main contribution, we show that CCT is constant-factor competitive for online facility location with recurring maximum demand. We also establish the offline foundations that underpin the proposed approach. Finally, we investigate the empirical approximation performance of CCT with synthetic and real-world instances.

Keywords. online facility location, recurring demand, competitive analysis, approximation algorithms

1 Introduction

1.1 Motivation

Facility location is a fundamental problem in operations research and computer science. While classically studied in various offline settings, many real-life facility location problems are inherently online. For instance, businesses often start with a single facility (e.g, a retail store, restaurant, etc.)

[*]Corresponding author. Address: 1025 Benedum Hall, Pittsburgh, PA 15261, USA.
[†‡]Emails: {lamperski, luw28}@pitt.edu; oleg.prokopyev@business.uzh.ch.

and decide when to locate new facilities based on customer demand. There is a growing body of work within the computer science literature on online facility location problems in which demand points arrive over time; see the surveys [22, 37]. The primary focus of this body of work is on *competitive analysis* [3, 17], that is, analyzing algorithm performance in terms of worst-case approximation ratios. Competitive analysis plays a fundamental role in a number of online operations research problems; see, for example, [5, 6, 16, 29, 36].

Most of the existing online facility location work builds upon the seminal study of Meyerson [38]. Meyerson considers a setup in which demand points arrive one-by-one at discrete times. At each time, one must decide whether to open a facility at the demand point that arrives at that time (at a fixed cost) or to use an existing facility to serve the newly arrived demand point (at the cost of the distance between the demand point and one of its nearest facilities). The demand is *one-time* in the sense that each demand point only needs to be serviced at its arrival time. The goal is to minimize the sum of the fixed and service costs; accordingly, the problem can be interpreted as an online version of the *uncapacitated facility location problem* (UFLP), also known as the *fixed cost median problem* [27]. Meyerson shows that, in general, it is not possible to construct a constant-factor competitive algorithm for online facility location with one-time demand. Meyerson also develops an algorithm that has a competitive ratio that is logarithmic in the number of times (or, equivalently, demand points). Interestingly, Meyerson shows that the algorithm is constant-factor competitive under the assumption that the demand points arrive in random order.

The story is similar for the closely related class of online clustering problems. In contrast to facility location work, clustering studies consider budgeted (as opposed to fixed cost) problems, where there is an upper bound k on the number of *centers* (the clustering analog of facilities) that can be specified. There are a number of different types of online clustering problems studied in the literature [10, 18, 19, 26, 32, 34]. For all of these problems, it is not hard to demonstrate that an algorithm that irrevocably locate centers cannot be competitive (at all); see, for example, [34]. Accordingly, online clustering studies consider algorithms that can update centers. In contrast to the outlined clustering context, however, facilities cannot be relocated in many practical, real-world situations. That said, relocating facilities enables algorithms to be constant-factor competitive for various online facility location problems [12].

In this work, we break away from Meyerson's one-time demand model and closely related online clustering models. We study a *recurring* demand model in which the demand points need to be serviced at all times after they arrive, not only the time that they arrive. Recurring demand arises in a variety of important application settings. Consider, for instance, distribution centers that serve their clients (e.g., wholesalers, retail stores) on a regular basis. Additionally, retail stores (e.g., department stores, grocery stores) often serve the same pool of customers over time. Other closely related examples include public transportation hubs, utility service and waste management facilities, and so on.

In the recurring demand model that we study, there is a fixed cost for locating a facility, and there is a *variable cost* at each time for serving points that have arrived up to that time. The variable costs captures the recurring demand. We define the variable cost to be the maximum distance between the points and their nearest facilities. Accordingly, the variable cost captures *recurring maximum demand.* By minimizing the variable cost we obtain a *minimax* problem, which simply ensures that the worst-case service cost stays low, unlike a *minisum* problem, which minimizes the total demand across all points. We refer to the outlined problem as *online facility location with recurring maximum demand.*

It is important to note that in the one-time demand model, the underlying primary concern is that, after locating a facility at a certain point, few or no future demand points are likely to arrive nearby. In the recurring demand model, this issue is less of a concern since all demand points need to be continuously served over time. Thus, this discussion naturally brings us to the following question:

Can we develop an algorithm that irrevocably locates facilities and is constant-factor competitive for online facility location with recurring maximum demand?

As our main contribution, we provide a positive answer to this question. Next, in Subsection 1.2, we describe the contributions of our work as a whole.

1.2 Summary of contributions and outline of the paper

Online facility location with recurring maximum demand. We first present a detailed description of the online facility location with recurring maximum demand model in Subsection 2.1. With the description in hand, we then present a precise comparison of our problem with the online facility location problem of Meyerson [38]; see Remark 2.1.

Then, in Subsection 2.2 we formally introduce our definition of the competitive ratio and the corresponding offline problems. Specifically, we consider a *fully offline* problem in which the facilities are all built before any demand point arrives and where we have full knowledge of the sequence in which points arrive in the future. We define the competitive ratio in terms of the optimal objective function value of the fully offline problem. We also consider a *semi-offline* problem, where facilities are constructed with complete knowledge of the future sequence of the demand point arrivals, but a facility can be built at a point only after that point has arrived. The setup of the semi-offline problem prevents the unrealistic phenomena of facilities being built in locations that are currently not near any points. To the best of our knowledge, the existing online facility location studies do not account for this consideration.

Finally, in Subsection 2.3, we present some additional background on offline facility location and clustering problems. We also further position our study within the existing literature.

Offline foundations. The aforementioned offline and semi-offline problems are novel variations of existing facility location models, and to the best of our knowledge, have yet to be explored in the related literature. Accordingly, we first establish *offline foundations* in Section 3.

Specifically, we show that both considered offline problems are NP-hard; see Subsection 3.1. Next, in Subsection 3.2 we provide mixed-integer linear programming (MILP) formulations of the problems. Finally, in Subsection 3.3, we show that the optimal value of the semi-offline problem is within a constant multiplicative factor of the optimal objective function value of the fully offline problem. We also show that the factor is asymptotically tight. Hence, while the fully offline problem can produce unrealistic solutions, the corresponding optimal objective function values are not that different (in a multiplicative sense). Consequently, we are not being overly conservative by defining the competitive ration in terms of the optimal fully offline objective value. We investigate this consideration from an empirical standpoint through the computational experiments in Section 5 as well.

Cumulative variable cost thresholding (CCT) algorithm. In Section 4, we propose and study an approximation algorithm that we call *Cumulative Variable Cost Thresholding* (CCT). The algorithm (irrevocably) locates a facility at a point that is farthest from an existing facility whenever its count of the cumulative variable cost incurred grows larger than a threshold, which we take to be the fixed cost. After constructing a center, CCT resets its count of the cumulative variable cost. Importantly, CCT is an intuitive algorithm that is reasonable for real-world online facility location applications: whenever the cost of serving demand points grows larger than the fixed cost of constructing a new facility, CCT builds a new facility.

As our main contribution, we show that CCT is 8-competitive. The result stands in contrast with the competitive analysis for the online facility location problems with one-time demand; recall that it is not possible to develop a constant-factor competitive algorithm in the context of one-time demand. Our result also implies that a constant-factor competitive ratio can be obtained by an intuitive rule that is simple enough to be used in practice. Finally, we also establish a lower bound on the competitive ratio of CCT.

Computational study. In Section 5, we explore the empirical performance of CCT using synthetic and real-world demand points; see Subsections 5.1 and 5.2, respectively. In the latter case, by considering DHL facility locations in the northeastern United States, we also examine the impact of the arrival order of the demand points on the performance of the proposed online and offline methods.

2 Preliminaries

We present a description of online facility location with recurring maximum demand in Subsection 2.1. Next, in Subsection 2.2, we introduce the corresponding fully offline and the semi-offline problems as well as our definition of the competitive ratio. Finally, we position the proposed problem setting within the related literature in Subsection 2.3.

2.1 Online facility location with recurring maximum demand

Let X be a set and $d : X \times X \to \mathbb{R}_{\geq 0}$ be a metric on X. A demand point $x_t \in X$ arrives at each time $t = 1, 2, \ldots$. Immediately after point x_t arrives, we decide whether or not to locate a facility at any (possibly multiple) of the points x_i, $i \in [t] := \{1, \ldots, t\}$ that have already arrived (and that do not already have facilities). We assume that the facilities cannot be relocated.

The fixed cost of constructing a facility is denoted by $\Gamma \in \mathbb{R}_{\geq 0}$. Assume that there is a given facility already built at $x_0 \in X$ at time $t = 0$ at no fixed cost. Let $F_0 := \{0\}$, and define $F_t \subseteq \{0\} \cup [t]$ to be the indices of the points that we select to locate facilities at up to and including time $t \geq 1$. It follows that $0 \in F_t$ for all times $t \geq 0$. We refer to F_t as a *facility index set.*

At each time $t \geq 1$, we model the variable cost for serving the points that have arrived up to time t as the maximum distance between any of these points and one of its nearest facilities that has been built. In other words our goal is to capture the worst case demand. Formally:

$$v_t(F_t) := \max_{i \in [t]} \min_{j \in F_t} d(x_i, x_j),$$

where $d(x_i, x_j)$ is the distance between x_i and x_j under the metric d. Note that our definition of the variable cost assumes that a point is served by one of its nearest facilities at every time. Accordingly, a point may be served by different facilities over the time horizon. Thus, the *total cost* incurred up to and including time $T \geq 1$ is given by:

$$c_T(F_1, \ldots, F_T) := \Gamma(|F_T| - 1) + \sum_{t=1}^{T} v_t(F_t),$$

where the first and the second terms represent the total cost of constructing facilities and the total variable cost, respectively. Recall also that $0 \in F_T$; hence, we have $\Gamma(|F_T|-1)$ instead of $\Gamma|F_T|$.

Remark 2.1. In the online facility location problem of Meyerson [38], the variable cost at time t is given by $v_t(F_t) = \min_{j\in F_t} d(x_t, x_j)$. ■

2.2 Offline problems and competitive ratio

Fully offline problem. In the fully offline setting, all facilities can be constructed at time $t = 0$ and the corresponding optimization problem is given by

$$\min_{0\in F\subseteq\{0\}\cup[T]} \Gamma(|F|-1) + \sum_{t=1}^{T} v_t(F), \tag{1}$$

where the assumption is that, when solving (1), we have full knowledge of the arrival sequence of the demand points in the future.

Remark 2.2. It is important to point out that, while all facilities are constructed at time $t = 0$ in the fully offline problem, the corresponding optimal solution of (1) still depends on the order in which the points arrive, ultimately due to the recurring demand. ■

Competitive ratio. Let OPT denote the optimal fully offline objective function value of problem (1). For $\alpha \geq 1$, we say that an algorithm is *α-competitive for online facility location with recurring maximum demand* if the algorithm constructs facility index sets $(F_1,\dots,F_T)$ such that

$$c_T(F_1,\dots,F_T) \leq \alpha \cdot \text{OPT}.$$

That is, the total cost incurred up to and including time T is no larger than a factor of α times the optimal fully offline cost up to and including time T. Finally, the *competitive ratio* of an α-competitive algorithm is simply the value of α.

Semi-offline problem. One could argue that the the fully offline problem is overly conservative by assuming that facilities can be constructed at points before they are realized in the arrival sequence. Accordingly, we also consider a *semi-offline* problem, where all facilities are still constructed with full knowledge of the arrival sequence of the demand points, but a facility can only be located at a point only after the point arrives. Formally, the *semi-offline* problem is given by:

$$\begin{aligned} \min_{F_1,\dots,F_T} \quad & c_T(F_1,\dots,F_T) \\ \text{s.t.} \quad & F_t \subseteq F_{t+1} && \forall t\in[T-1] \\ & 0\in F_t \subseteq \{0\}\cup[t] && \forall t\in[T]. \end{aligned} \tag{2}$$

To the best our knowledge, both problems (1) and (2) are novel variants of the facility location problem and have not yet been studied in the existing literature. Moreover, as far as we are aware, analogous notions of semi-offline problems have also not been explored in the context of other online facility location and clustering problems. Hence, in Subsection 3.1, we first explore the theoretical computational complexity of (1) and (2). Afterwards, in Subsection 3.2 we describe MILP reformulations for both of these problems.

Finally, let OPT_{semi} denote the optimal objective function value of the semi-offline problem (2). Clearly, it holds that $\text{OPT} \leq \text{OPT}_{\text{semi}}$. In Subsection 3.3 we show that OPT_{semi} is bounded above by a constant multiplicative factor of OPT. Consequently, we are – at least in some sense – not being overly conservative by using OPT instead of OPT_{semi} in our definition of a competitive algorithm.

2.3 Offline background and related online work

We first briefly overview some background on offline facility location and clustering. Then, we discuss the most closely related work on the corresponding online versions of these problem classes.

Offline background. We direct the reader to [9, 15] for a broad overview of the facility location literature, which is extensive. Hence, our focus below is only on the most relevant offline work. Recall from Subsection 1.1 that Meyerson's problem [38] can be thought of as an online version of the uncapacitated facility location problem (UFLP) [27, 41]. In our notation, UFLP is given by

$$\min_{0 \in F \subseteq \{0\} \cup [T]} \; \Gamma(|F| - 1) + \sum_{t=1}^{T} \min_{j \in F} d(x_i, x_j).$$

There is a substantial body of research for this class of problems. There are several MILP formulations and decomposition-based solution schemes along with the corresponding polyhedral analyses; see, for example, [13, 14, 42]. There is also a long line of work on approximation algorithms. Without any assumption on the distances d (e.g., without assuming that d is a metric), UFLP admits only $\log(T)$-approximation algorithms based on a set-cover reformulation as well as the rounding and filtering techniques [27, 35]. In contrast, there are constant-factor approximation algorithms for the *metric* UFLP [7, 11, 30, 33, 40]. Guha & Kuller [25] show that there is no α-approximation scheme with $\alpha < 1.463$ for the *metric* UFLP unless $NP \subseteq DTIME(n^{\log\log(n)})$. Li [33] gives the approximation algorithm with the best known (to date) approximation factor of 1.488.

A closely related clustering problem is the k-median problem given by:

$$\min_{F \subseteq [T]:\ |F| \leq k} \; \sum_{t=1}^{T} \min_{j \in F} d(x_i, x_j),$$

which clearly can be interpreted as a budgeted version of UFLP. Naturally, there are also a number of constant-factor approximation algorithms for the k-median problem [8, 30, 35].

The k-center problem is another well-studied clustering problem, where the goal is to minimize the maximum distance from any point to its closest center. Formally, in our notation, it is given by:

$$\min_{F\subseteq[T]:\ |F|\leq k} \max_{i\in[T]} \min_{j\in F} d(x_i, x_j),$$

and one can observe that the max-min objective function of the k-center problem takes the same form as the variable costs $v_t(F_T)$, $t \in [T]$, in the online facility location problem of interest. There are also a number of constant-factor approximation algorithms for this clustering problem; see, e.g., [23, 24, 28, 39]. In particular, the classical greedy algorithm of Gonzalez [24] successively locates new facilities at the points that are farthest from the already placed facilities. Thus, our CCT algorithm can be seen as an online adaptation of this algorithm. Finally, the approach similar in spirit to Gonzalez's algorithm is also exploited in a recent work [31] for the adaptive optimization setting.

Related online work. The surveys in [22, 37] provide an overview of the results on online facility location. The related studies mostly build upon Meyerson's work, who developed a randomized $O(\log(T))$-competitive algorithm for the one-time demand model. Fotakis [21] later showed that the algorithm is actually $O(\log(T)/\log(\log(T)))$-competitive. There also exist deterministic $O(\log(T))$-competitive algorithms [4, 20]. In comparison, CCT proposed in this study is a deterministic constant-factor competitive algorithm for online facility location with recurring maximum demand.

With respect to online clustering problems, there is prior work on online k-median [18, 19, 26], online k-center [10] and online k-means [32, 34] clustering problems. Recall that clustering algorithms update centers (that is, facilities), as competitive guarantees are otherwise unattainable. Of course, it is not reasonable to assume that the offline problem can be resolved at every time. The earlier online clustering studies [10, 19] on *incremental clustering* assume that clusters can be merged, ensuring that clusters are, in some sense, preserved. Finally, more recent studies [18, 26, 32, 34] on *consistent clustering* do not require merging, but rather aim to minimize the number of center updates (called *recourse*) required to maintain a constant-factor competitive ratio.

3 Offline foundations

First, we demonstrate that both the fully offline and semi-offline problems are NP-hard in Subsection 3.1. Then, in Subsection 3.2, we present the corresponding MILP formulations, which are later used in our numerical experiments. Finally, in Subsection 3.3, we show that the optimal objective function value for the semi-offline problem is bounded above by a constant multiplicative factor of the optimal objective function value for the fully offline problem.

3.1 Computational complexity

Fully offline problem. Define the *budgeted fully offline* (BFO) optimization problem as follows:

$$\min_{F} \left\{ \sum_{t=1}^{T} v_t(F) \ : \ |F| \leq k+1 \text{ and } 0 \in F \subseteq \{0\} \cup [T] \right\},$$

where the budget parameter k specifies the upper bound on the maximum number of facilities.

From a standard binary search argument, to establish hardness of the fully offline optimization problem (1), it is sufficient to show that a decision version of the BFO problem is NP-complete. Let

$$\mathcal{F}_{BFO} := \{F \in \{0\} \cup [T] : |F| \leq k+1 \text{ and } 0 \in F \subseteq \{0\} \cup [T]\}$$

denote the feasible region of the BFO optimization problem. Accordingly, we define:

<u>BFO decision problem</u>

- *Instance.* Points $x_0, x_1, \ldots, x_T \in X$, metric $d : X \times X \to \mathbb{R}_{\geq 0}$, budget parameter $k \in \mathbb{Z}_{>0}$, and target objective value $\alpha \in \mathbb{Z}_{>0}$.
- *Question.* Is there a solution $F \in \mathcal{F}_{BFO}$ such that $\sum_{t=1}^{T} v_t(F) \leq \alpha$?

Our proof of hardness (i.e., of Proposition 3.1 below) is based on a reduction from the *dominating set* problem. Recall that a dominating set in a graph $G = (V, E)$ is a subset of vertices $V' \subseteq V$ such that every vertex not in V' is adjacent to a vertex in V'. The goal in the dominating set problem is to determine whether a given graph contains a dominating set of a given size.

Proposition 3.1. *The BFO decision problem is NP-complete.*

Proof. See Appendix A.1. □

Semi-offline problem. Our proof of hardness for the semi-offline problem proceeds in as similar spirit to the argument provided above for the fully offline problem. We show that (a decision version of) the *budgeted semi-offline* (BSO) optimization problem

$$\begin{aligned}
\min_{F_1,\ldots,F_T} \quad & \sum_{t=1}^{T} v_t(F_1, \ldots, F_t) & \\
\text{s.t.} \quad & F_t \subseteq F_{t+1} & t \in [T-1] \\
& |F_T| \leq k+1 & \\
& 0 \in F_t \subseteq \{0\} \cup [t] & t \in [T]
\end{aligned}$$

is NP-hard. Let $\mathcal{F}_{BSO}$ denote the feasible region of the BSO optimization problem. Then, define:

BSO decision problem

- *Instance.* Points $x_0, x_1, \ldots, x_T \in X$, metric $d : X \times X \to \mathbb{R}_{\geq 0}$, budget parameter $k \in \mathbb{Z}_{>0}$, and target objective value $\alpha \in \mathbb{Z}_{>0}$.
- *Question.* Is there a solution $(F_1, \ldots, F_T) \in \mathcal{F}_{BSO}$ such that $\sum_{t=1}^{T} v_t(F_1, \ldots, F_t) \leq \alpha$?

The proof of hardness for the BSO problem is also based upon the dominating set problem.

Proposition 3.2. *The BSO decision problem is NP-complete.*

Proof. See Appendix A.2. □

3.2 MILP formulations

Fully offline problem. The decision variables of our MILP formulation for the fully offline problem are as follows. For each $i \in [T]$, we introduce a binary variable $y_i \in \{0,1\}$ that indicates whether or not we locate a facility at point x_i. Also, for each $(i,j) \in [T] \times (\{0\} \cup [T])$, we introduce a binary variable $z_{ij} \in \{0,1\}$ that indicates whether or not point x_i is "assigned" to a facility at point x_j (i.e., if a facility at x_j is a nearest facility to point x_i, where ties are broken arbitrarily).

Then, the MILP model of the fully offline problem is as follows:

$$\min_{y,z,\zeta} \quad \Gamma \sum_{j=1}^{T} y_j + \sum_{t=1}^{T} \zeta^t$$

$$\text{s.t.} \quad \zeta^t \geq \sum_{j=0}^{T} d(x_i, x_j) z_{ij} \qquad \forall t \in [T],\ \forall i \in [t] \tag{3a}$$

$$z_{ij} \leq y_j \qquad \forall (i,j) \in [T] \times [T] \tag{3b}$$

$$\sum_{j=0}^{T} z_{ij} = 1 \qquad \forall i \in [T] \tag{3c}$$

$$y \in \{0,1\}^T,\ z \in \{0,1\}^{T \times (T+1)},\ \zeta \in \mathbb{R}_+^T,$$

where (3a) ensures that ζ^t is at least the maximum distance between a point that arrives up to and including time t and its nearest facility. Constraints (3b) guarantee that a facility can be assigned points only if it is constructed, while (3c) enforces that each point must be assigned to one facility.

Semi-offline problem. In the corresponding MILP, for each $t \in [T]$ and $j \in [t]$, we introduce a binary decision variable $y_j^t \in \{0,1\}$ such that $y_j^t = 1$ if we locate a facility at point x_j by time t, and $y_j^t = 0$ otherwise. Also, for each $t \in [T]$, we introduce a continuous variable $\zeta^t \in \mathbb{R}$ that models the variable cost $v_t(F_t)$. Finally, we introduce a binary variable $z_{ij}^t \in \{0, 1\}$ for each $t \in [T]$, $i \in [t]$, and $j \in \{0\} \cup [t]$ that indicates whether or not point x_i is "assigned" to a facility at x_j at time t (i.e., if a facility at x_j is a nearest facility to point x_i at time t, where ties are broken arbitrarily).

Then, the MILP reformulation of the semi-offline problem is given by:

$$\min_{\zeta, y, z} \quad \Gamma \sum_{j=1}^{T} y_j^T + \sum_{t=1}^{T} \zeta^t$$

$$\text{s.t.} \quad \zeta^t \geq \sum_{j=0}^{t} d(x_i, x_j) \cdot z_{ij}^t \qquad \forall t \in [T],\ \forall i \in [t] \tag{4a}$$

$$\sum_{j=0}^{t} z_{ij}^t = 1 \qquad \forall t \in [T],\ \forall i \in [t] \tag{4b}$$

$$z_{ij}^t \leq y_j^t \qquad \forall t \in [T],\ \forall i \in [t],\ \forall j \in [t] \tag{4c}$$

$$y_j^t \leq y_j^{t+1} \qquad \forall t \in [T-1],\ \forall j \in [t] \tag{4d}$$

$$\zeta^t \in \mathbb{R}_+,\ y^t \in \{0,1\}^t,\ z^t \in \{0,1\}^{t \times (t+1)} \qquad \forall t \in [T],$$

where constraints (4a) are incorporated to capture the variable costs in the objective. Constraints (4b) ensure that a point that has arrived thus far is assigned to a facility, while constraints (4c) ensure that a point can only be assigned to a facility that has actually been constructed. Finally, constraints (4d) enforce that a facility remains constructed once it is constructed.

3.3 Optimal objective function values comparison

Recall from Subsection 2.2 that OPT and OPT_{semi} denote the optimal fully offline and optimal semi-offline objective function values, respectively. Also, recall that $\text{OPT} \leq \text{OPT}_{\text{semi}}$. Next, we address the following question: How much larger (in a multiplicative sense) can OPT_{semi} be than OPT?

Our proof of the result below follows from constructing a feasible solution to the semi-offline problem from an optimal solution to the fully offline problem. Specifically, the feasible solution to the semi-offline problem locates a facility at the first point to arrive in each cluster induced by the optimal fully offline solution. The factor of 2 then follows from the triangle inequality. Formally:

Proposition 3.3. $\text{OPT}_{\text{semi}} \leq 2\text{OPT}$.

Proof. See Appendix A.3. □

Is the factor of 2 in Proposition 3.3 tight? Next, we show that it is asymptotically tight, i.e., it is tight up to a term that vanishes as $T \to \infty$; see Proposition 3.4 below. We present an illustration of the instance presented in the proof of the proposition for $T = 4$ in Figure 1. Taking the fixed cost to be $\Gamma = 2$, the optimal fully offline solution locates a facility at the point x_4, which arrives last. Hence, $\mathrm{OPT} = 2 + T$. A semi-offline solution can only locate a facility at a point after it arrives, so the optimal semi-offline solution locates no facilities. Accordingly, $\mathrm{OPT}_{\mathrm{semi}} = 2T$. Thus, the key idea is to require that the demand point, at which the optimal offline facility is located, to arrive last. On a related note, we explore the empirical impact of the arrival order of demand points in Section 5.

Proposition 3.4. *For each integer $T \geq 2$, there is an instance of online facility with recurring maximum demand such that* $\mathrm{OPT}_{\mathrm{semi}} = \left(2 - \frac{4}{T+2}\right) \mathrm{OPT}$.

Proof. See Appendix A.4. □

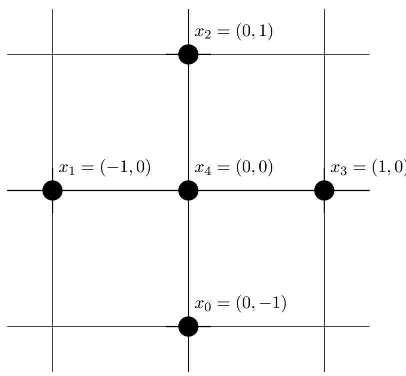


Figure 1: Illustration of the instance presented in the proof of Proposition 3.4 for $T = 4$. The metric for the instance is the taxicab metric, i.e., $d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$ for $x, y \in \mathbb{R}^n$.

4 Cumulative variable cost thresholding (CCT) algorithm

We present a detailed description of the Cumulative Variable Cost Thresholding (CCT) algorithm in Algorithm 1. Specifically, Step 1 initializes the cumulative variable cost counter cumVarCost. Step 3 verifies whether the cumulative variable cost counter is at least the fixed cost Γ. If it is, then the index of the point that is farthest from one of its nearest facilities is selected in Step 4. The index is then added to the facility index set in Step 5, and the cumulative variable cost counter is reset in Step 6. If the cumulative variable cost counter does not exceed Γ, then the facility index set and cumulative variable cost counter are updated in Steps 8 and 9, respectively. Finally, the facility index sets constructed by CCT are returned in Step 12.

Computational complexity. Step 4 is the most computationally intensive step of each iteration. At the t-th iteration, the step requires $O(t^2)$ evaluations of the distance metric. If the distances from previous iterations are stored in memory, then the step only requires $O(t)$ evaluations.

Algorithm 1 Cumulative Variable Cost Thresholding (CCT)

Input: Points $x_t \in \mathbb{R}^n$, $t \in \{0\} \cup [T]$, metric $d : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_{\geq 0}$, and fixed cost Γ

cumVarCost $\leftarrow 0$
for $t = 1, \ldots, T$ **do**
  if cumVarCost $+ v_t(F_{t-1}) \geq \Gamma$ **then**
    $\ell \leftarrow \ell' \in \text{argmax}_{i \in [t]} \min_{j \in F_{t-1}} d(x_i, x_j)$
    $F_t \leftarrow F_{t-1} \cup \{\ell\}$
    cumVarCost $\leftarrow 0$
  else
    $F_t \leftarrow F_{t-1}$
    cumVarCost $\leftarrow$ cumVarCost $+ v_t(F_t)$
  end if
end for
Return $(F_1, \ldots, F_T)$

Competitive ratio guarantee. We establish that CCT is 8-competitive; see Theorem 4.1 below. Accordingly, online facility location with recurring maximum demand is constant-factor approximable in contrast to the well-studied one-time demand setup.

Theorem 4.1. *CCT (Algorithm 1) is* 8*-competitive for online facility location with recurring maximum demand.*

Proof. See Subsection 4.2. □

However, it should be pointed out that it is not clear whether our analysis of CCT is tight. We establish an asymptotic lower bound of 4 for on the competitive ratio of CCT; see Proposition 4.1 below. The competitive ratio of α tends to 4 as $T \to \infty$ and $\epsilon \to 0$. We leave it as an open problem to close the gap between 4 and 8; see Section 6 for further discussion.

Proposition 4.1. *For each integer $T \geq 3$ and $0 < \epsilon < \Gamma$, there exists an instance of online facility location with recurring maximum demand per which CCT provides an α-approximation for*

$$\alpha := \frac{(2\Gamma - \epsilon)T - 2\Gamma}{(\Gamma/2)T + \Gamma - \epsilon}.$$

Proof. See Subsection 4.3. □

The remainder of this section is dedicated to the competitive analysis of CCT. We establish preliminaries in Subsection 4.1, provide a proof of the main result, Theorem 4.1, in Subsection 4.2, and, finally, present a proof of Proposition 4.1 in Subsection 4.3.

4.1 Preliminaries

CCT output and optimal offline solution. Let $(F_1, \ldots, F_T)$ denote the output of CCT (i.e., Algorithm 1), and let $F^* \subseteq \{0\} \cup [T]$ denote an optimal solution to the fully offline problem (1). Suppose that CCT locates k facilities up to and including time T, i.e., $|F_T| = k+1$ (recall that $0 \in F_T$).

Without loss of generality, we assume that CCT locates at least one facility, i.e., $k \geq 1$. Indeed, in the event that CCT does not locate any facilities, there is a (unique) optimal offline solution that does not locate any facilities (as the fixed cost of locating a facility exceeds the variable costs incurred otherwise); hence, CCT is 1-competitive in this case. Let $k^* := |F^*| - 1$ denote the number of facilities that are constructed under the optimal offline solution F^*. Define

$$C_j^* := \left\{ i \in \{0\} \cup [T] \ : \ j \in \operatorname*{argmin}_{\ell \in F^*} d(x_i, x_\ell) \right\}, \quad j \in F^* \tag{5}$$

to be the subsets (referred further as *clusters*) that are induced by F^*; we say they are *optimal clusters*. That is, $i \in C_j^*$ if and only if the facility at x_j is nearest to point x_i. Because we do not break the ties in any way, the clusters do not necessarily partition $\{0\} \cup [T]$, but they do cover $\{0\} \cup [T]$.

CCT build times and locations. Define $t_1 < \ldots < t_k \in [T]$ to be the times at which CCT locates facilities (i.e., not including the facility at point x_0). Also, define $t_0 := 0$ and $t_{k+1} := T$. Note that $t_0 < t_1$. Furthermore, note that if CCT locates a facility at time T, then $t_k = t_{k+1}$. Otherwise, it holds that $t_k < t_{k+1}$. For each $\ell \in [k]$, let $f_\ell \in [T]$ denote the index of the point at which CCT locates a facility at time t_ℓ. It follows that $f_\ell \leq t_\ell$ for each $\ell \in [k]$, and $F_T = \{0, f_1, \ldots, f_k\}$.

Discrete time intervals and associated costs. Define the discrete time intervals:

$$I_\ell := (t_{\ell-1}, t_\ell] := \{t_{\ell-1} + 1, \ldots, t_\ell\}, \quad \ell \in [k+1].$$

In the event that CCT locates a facility at time T (i.e., $t_k = t_{k+1}$), we assume the convention that $I_{k+1} := \emptyset$. The time intervals I_ℓ, $\ell \in [k+1]$, form a partition of $[T]$. For $\ell \in [k]$, CCT locates a facility at the last time t_ℓ in the interval I_ℓ and no other times in the interval. CCT does not locate any facilities at times in the interval I_{k+1}.

For each $\ell \in [k+1]$, define

$$v^{(\ell)} := \sum_{t \in I_\ell} v_t(F_t)$$

to be the total variable cost incurred under CCT during the nterval I_ℓ. Also, for each $\ell \in [k]$, define

$$c^{(\ell)} := \Gamma + v^{(\ell)}$$

to be the total cost incurred under CCT during the time interval I_ℓ. With these definitions in hand, we can decompose the total cost incurred under CCT as follows:

$$\begin{aligned} c_T(F_1, \ldots, F_T) = \Gamma(|F_T| - 1) + \sum_{t=1}^{T} v_t(F_t) = \Gamma k + \sum_{\ell=1}^{k+1} \sum_{t \in I_\ell} v_t(F_t) \\ = \Gamma k + \sum_{\ell=1}^{k+1} v^{(\ell)} = \sum_{\ell=1}^{k} c^{(\ell)} + v^{(k+1)}, \end{aligned} \tag{6}$$

where the second equality follows from $|F_T| = k+1$ and the fact that the intervals I_ℓ, $\ell \in [k+1]$, form a partition $[T]$; the third equality follows from the definition of $v^{(\ell)}$; and the last inequality follows from the definition of $c^{(\ell)}$.

Basic properties. In Proposition 4.2 below, we make record of some basic properties of the variable costs. The properties follow from our definition of the variable costs.

Proposition 4.2. *Let $\hat{F}, \tilde{F} \subseteq \{0\} \cup [T]$ such that $0 \in \hat{F}$ and $0 \in \tilde{F}$. Then:*

(i) if $\hat{F} \subseteq \tilde{F}$, then $v_t(\hat{F}) \geq v_t(\tilde{F})$ for each $t \in [T]$;

(ii) $v_t(\hat{F}) \leq v_{t+1}(\hat{F})$ for each $t \in [T-1]$.

Proof. Regarding statement (i), we have:

$$v_t(\hat{F}) = \max_{i \in [t]} \min_{j \in \hat{F}} d(x_i, x_j) \geq \max_{i \in [t]} \min_{j \in \tilde{F}} d(x_i, x_j) = v_t(\tilde{F}),$$

where the inequality follows from $\hat{F} \subseteq \tilde{F}$. For statement (ii), we have:

$$v_t(\hat{F}) = \max_{i \in [t]} \min_{j \in \hat{F}} d(x_i, x_j) \leq \max_{i \in [t+1]} \min_{j \in \hat{F}} d(x_i, x_j) = v_{t+1}(\hat{F}),$$

which completes the proof. □

In Proposition 4.3 below, we provide some basic properties of CCT in terms of the discrete time interval notation introduced above. The first two properties follow immediately. The last property requires an inductive argument.

Proposition 4.3. *The following statements hold:*

(i) $\sum_{t \in I_\ell \setminus \{t_\ell\}} v_t(F_{t_{\ell-1}}) < \Gamma$ for each $\ell \in [k]$, and $v^{(k+1)} < \Gamma$.

(ii) $\sum_{t \in I_\ell} v_t(F_{t_{\ell-1}}) \geq \Gamma$ for each $\ell \in [k]$.

(iii) $v_t(F_t) \leq \Gamma$ for each $t \in [T]$.

Proof. Statements (i) and (ii) are relatively straightforward. They follow directly from the fact that CCT locates facilities at times $t_1, \dots, t_k$ when the algorithm's count of the cumulative variable cost is at least Γ. Next, we apply induction to establish statement (iii) as follows.

Base case. Suppose that $v_1(F_0) \leq \Gamma$. Because $F_0 \subseteq F_1$, we have from part (i) of Proposition 4.2 that $v_1(F_1) \leq \Gamma$. Suppose that $v_1(F_0) > \Gamma$. Then, CCT locates a facility at point x_1 at time 1, so $F_1 = \{0, 1\}$. Consequently, $v_1(F_1) = 0 \leq \Gamma$.

Inductive step. Suppose that $v_t(F_t) \leq \Gamma$ for some $t \geq 1$. If $v_{t+1}(F_t) \leq \Gamma$, then by part (i) of Proposition 4.2, $v_{t+1}(F_{t+1}) \leq v_{t+1}(F_t) \leq \Gamma$. Next, suppose that $v_{t+1}(F_t) > \Gamma$. Then, CCT locates a facility at point x_{t+1} at time $t+1$ because $v_t(F_t) \leq \Gamma$. Consequently, $v_{t+1}(F_{t+1}) \leq \Gamma$. □

Finally, we derive a simple upper bound on $d(x_i, x_j)$ for $j \in F^*$ and $i \in C_j^*$.

Proposition 4.4. *For $j \in F^*$, $i \in C_j^*$, and $t \in [T]$ such that $i \leq t$,*

$$d(x_i, x_j) \leq v_t(F^*).$$

Proof. Because $i \leq t$, we can (inductively) apply part (ii) of Proposition 4.2 to obtain that

$$v_t(F^*) \geq v_i(F^*) = \max_{u \in [i]} \min_{v \in F} d(x_u, x_v) \geq \min_{v \in F} d(x_i, x_v) = d(x_i, x_j),$$

where the last equality follows from $i \in C_j^*$. □

4.2 Competitive analysis

We carry over the notation and the assumptions established in Subsection 4.1. We begin with an overview of our competitive analysis of CCT:

Overview of competitive analysis. First, we carefully construct a particular subset $L \subseteq [k]$ that we use to partition the intervals I_ℓ, $\ell \in [k]$; see (7) below together with the surrounding discussion. Next, we establish properties of L; see Lemmas 4.1 and 4.2. Then, we show the total cost $\sum_{\ell \in [k] \setminus L} c^{(\ell)}$ incurred across intervals indexed by $[k] \setminus L$ is within a constant factor of the total optimal offline variable cost $\sum_{t=1}^{T} v_t(F^*)$; see Lemma 4.3. Thereafter, we proceed to show that the total cost $\sum_{\ell \in L} c^{(\ell)}$ incurred across intervals indexed by L is at most a constant factor times Γk^*; see Lemma 4.4. The desired result then follows. In the discussion below, whenever we say CCT "selects an index," we mean CCT constructs a facility at the respective point; see Algorithm 1.

Construction of set L. Recall the definition (5) of the optimal clusters C_j^*, $j \in F^*$, from Subsection 4.1. Define

$$S_1 := \{j \in F^* \setminus \{0\} : C_j^* \cap F_T \neq \emptyset\}.$$

That is, for $j \in F^* \setminus \{0\}$, we have that $j \in S_1$ if and only if CCT selects an index from C_j^* to be a facility index. For $j \in S_1$, define

$$b_j := \min_{i \in [k]} \{t_i : f_i \in C_j^*\}$$

to be the first time at which CCT selects an index from C_j^* to be a facility index. We collect the indices of these selection times into the set

$$L_1 := \{\ell \in [k] : t_\ell = b_j \text{ for some } j \in S_1\}.$$

Also, define

$$a_j := \min\{t \in \{0\} \cup [T] : t \in C_j^*\}$$

to be the first time that a point indexed by cluster C_j^* arrives. For $j \in F^* \setminus \{0\}$, define the set

$$S_2 := \left\{j \in F^* \setminus \{0\} : a_j \leq \max_{\ell \in [k]} \{t_\ell : \ell \notin L_1\}\right\}.$$

That is, for $j \in F^* \setminus \{0\}$, we have that $j \in S_2$ if and only if after or at time a_j, CCT selects an index from an optimal cluster that CCT already selected an index from. For $j \in S_2$, define

$$d_j := \min_{\ell \in [k]} \{t_\ell : \ell \notin L_1 \text{ and } a_j \leq t_\ell\}$$

to be the first time after or at a_j that CCT selects an index from an optimal cluster that CCT has already selected an index from. We collect the indices of these selection times into the set

$$L_2 := \{\ell \in [k] : t_\ell = d_j \text{ for some } j \in S_2\}.$$

Finally, we define

$$L := L_1 \cup L_2. \tag{7}$$

Note that

$$|L| \leq |L_1| + |L_2| = |S_1| + |S_2| \leq 2|F^* \setminus \{0\}| \leq 2k^*. \tag{8}$$

Properties of L. Lemma 4.1 states sufficient conditions for membership in L. Following the proof of the lemma, we discuss how we use these conditions in the overall proof of Theorem 4.1. The proof of Lemma 4.1 simply follows from the construction of L.

Lemma 4.1. *Let $\ell \in [k]$. Suppose that there exists $i \leq t_\ell$ such that $i \in C_j^*$ for some $j \in F^* \setminus \{0\}$.*

Further suppose that either

(i) $j \notin S_2$*, or*

(ii) $j \in S_2$ *and* $t_\ell \leq d_j$*.*

Then, it holds that $\ell \in L$*.*

Proof. Note that $t_\ell \geq i \geq a_j$, where the second inequality follows from the fact that $i \in C_j^*$ and a_j is the first time at which a point indexed by C_j^* arrives.

Suppose that $j \notin S_2$. Then, after and including time a_j, CCT does not select an index from an optimal cluster that CCT has already selected an index from. So, at time $t_\ell \geq a_j$, CCT selects an index from an optimal cluster that CCT has not yet selected an index from. In other words, we have that $\ell \in L_1 \subseteq L$.

Suppose that $j \in S_2$ and $t_\ell \leq d_j$. In the case that $t_\ell = d_j$, it holds that $\ell \in L_2 \subseteq L$. Accordingly, suppose that $t_\ell < d_j$. Recall that d_j is the first time after or at a_j that CCT selects an index from an optimal cluster that CCT has already selected an index from. Because $d_j > t_\ell \geq i \geq a_j$, it must be the case at time t_ℓ that CCT selects an index from an optimal cluster that CCT has not yet selected an index from. That is, $\ell \in L_1 \subseteq L$. □

We exploit Lemma 4.1 to prove Lemma 4.2 below. Lemma 4.2 states a bound on $v_t(F_{t_{\ell-1}})$ for each $\ell \in [k] \setminus L$ and $t \in I_\ell$. Ultimately, we use these bounds to, in turn, bound the total cost $c^{(\ell)}$ incurred in each interval I_ℓ indexed by $\ell \in [k] \setminus L$.

We apply Lemma 4.1 in the second case of the proof of Lemma 4.2 as follows; see also the accompanying illustration in Figure 2. Let $\ell \in [k] \setminus L$ and $t \in I_\ell$. The overall goal in the proof is to show that any point x_i that arrives before or at time t is not "too far" from a facility built by time $t_{\ell-1}$. The second case of the proof supposes that $i \in C_j^*$ for $j \in F^* \setminus \{0\}$. Recalling $\ell \in [k] \setminus L$, Lemma 4.1 implies that $j \in S_2$ and $t_\ell > d_j$. Note that it follows that $t_{\ell-1} \geq d_j$ because $d_j \in \{t_1, \ldots, t_k\}$. CCT selects an index f_h from an optimal cluster C_q^* that CCT has already selected an index from at time $d_j \geq a_j$, so the point x_{a_j} must not be too far from a facility built by time $d_j \leq t_{\ell-1}$ (otherwise, CCT would have selected a_j). In Figure 2, point x_{a_j} is not too far from the facility at point x_{f_g}. Furthermore, we know that x_i is not too far from x_{a_j} because their indices lie in the same optimal cluster. Thus, point x_i cannot be too far from the facility that x_{a_j} is not too far from, namely the facility at x_{f_g} in Figure 2.

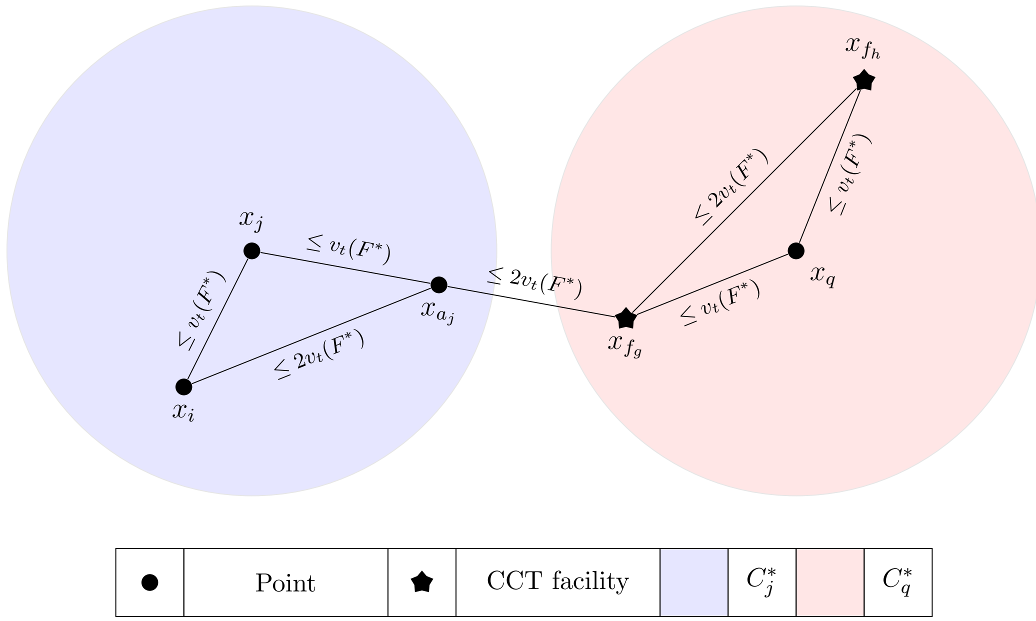


Figure 2: Illustration of the second case in the proof of Lemma 4.2.

Lemma 4.2. *For $\ell \in [k] \setminus L$ and $t \in I_\ell$,*

$$v_t(F_{t_{\ell-1}}) \leq 4v_t(F^*).$$

Proof. Recall that $v_t(F_{t_{\ell-1}}) = \max_{i \in [t]} \min_{u \in F_{t_{\ell-1}}} d(x_i, x_u)$. Accordingly, fixing an arbitrary $i \in [t]$, it is sufficient to show that

$$\min_{u \in F_{t_{\ell-1}}} d(x_i, x_u) \leq 4v_t(F^*). \tag{9}$$

Take $j \in F^*$ such that $i \in C_j^*$. We consider two cases:

Case 1. Suppose that $j = 0$. Because $0 \in F_{t_{\ell-1}}$,

$$\min_{u \in F_{t_{\ell-1}}} d(x_i, x_u) \leq d(x_i, x_0) \leq v_t(F^*),$$

where the second inequality follows from Proposition 4.4 (recall $i \in C_0^*$ and $i \leq t$). Thus, (9) holds.

Case 2. Suppose that $j \neq 0$. Because $\ell \in [k] \setminus L$, it follows from Lemma 4.1 that $j \in S_2$ and $t_\ell > d_j$. It further follows that $t_{\ell-1} \geq d_j$ because $d_j \in \{t_1, \ldots, t_k\}$.

Let $h \in [k]$ such $d_j = t_h$. At time t_h, CCT selects index $f_h \in C_q^*$ for some $q \in F^*$. Furthermore, because $t_h = d_j$, CCT must have selected another index from C_q^* at a time $t_g < t_h$. That is, CCT selects $f_g \in C_q^*$ at time t_g.

Because $t_{\ell-1} \geq d_j = t_h \geq t_{h-1}$ (note that t_{h-1} is well-defined because t_g is well-defined and

$t_g < t_h$), we have that

$$\min_{u \in F_{t_{\ell-1}}} d(x_{a_j}, x_u) \leq \min_{u \in F_{t_{h-1}}} d(x_{a_j}, x_u) \leq \min_{u \in F_{t_{h-1}}} d(x_{f_h}, x_u) \leq d(x_{f_g}, x_{f_h})$$
$$\leq d(x_{f_g}, x_q) + d(x_q, x_{f_h}) \leq 2v_t(F^*), \quad (10)$$

where the second inequality follows from the fact that CCT selects index f_h at time $d_j = t_h$ instead of index a_j (recall line 4 in Algorithm 1); the third inequality follows from $f_g \in F_{t_g} \subseteq F_{t_{h-1}}$; the fourth inequality follows from the triangle inequality; finally, the last inequality follows from Proposition 4.4, which is applicable because $f_g, f_h \in C^*_q$, $f_g \leq t_g \leq t_{\ell-1} < t$, and $f_h \leq t_h \leq t_{\ell-1} < t$.

Take $w \in \text{argmin}_{u \in F_{t_{\ell-1}}} d(x_{a_j}, x_u)$, so $d(x_{a_j}, x_w) \leq 2v_t(F^*)$ from (10). Because $w \in F_{t_{\ell-1}}$:

$$\min_{u \in F_{t_{\ell-1}}} d(x_i, x_u) \leq d(x_i, x_w) \leq d(x_i, x_{a_j}) + d(x_{a_j}, x_w)$$
$$\leq d(x_i, x_j) + d(x_j, x_{a_j}) + 2v_t(F^*) \leq 4v_t(F^*),$$

where the second inequality follows from the triangle inequality; the third inequality follows from the triangle inequality and $d(x_{a_j}, x_w) \leq 2v_t(F^*)$; and the last inequality follows from Proposition 4.4, which can be applied because $a_j, i \in C^*_j$, $a_j \leq d_j \leq t_{\ell-1} < t$, and $i \leq t$. Thus, (9) holds, which completes the proof. □

Bounding costs incurred. We turn our attention to bounding the costs incurred under CCT. The bounds more or less follow from Lemma 4.2 and Proposition 4.3.

Lemma 4.3. $\sum_{\ell \in [k] \setminus L} c^{(\ell)} \leq 8 \sum_{t=1}^{T} v_t(F^*)$.

Proof. If $[k] \setminus L = \emptyset$, then the desired result trivially holds. Thus, suppose that $[k] \setminus L \neq \emptyset$. Let $\ell \in [k] \setminus L$. Note from the definition of $v^{(\ell)}$ that

$$v^{(\ell)} = \sum_{t \in I_\ell} v_t(F_t) \leq \sum_{t \in I_\ell} v_t(F_{t_{\ell-1}}) \leq 4 \sum_{t \in I_\ell} v_t(F^*), \quad (11)$$

where the first inequality follows from part (i) of Proposition 4.2 ($F_{t_{\ell-1}} \subseteq F_t$ for $t \in I_\ell$), and the second inequality follows from Lemma 4.2. Also, note from part (ii) of Proposition 4.3 that

$$\Gamma \leq \sum_{t \in I_\ell} v_t(F_{t_{\ell-1}}) \leq 4 \sum_{t \in I_\ell} v_t(F^*), \quad (12)$$

where the second inequality follows from Lemma 4.2.

Finally, from the definition of $c^{(\ell)}$, we have that:

$$\sum_{\ell\in[k]\setminus L} c^{(\ell)} = \sum_{\ell\in[k]\setminus L} (\Gamma + v^{(\ell)}) \leq 8 \sum_{\ell\in[k]\setminus L} \sum_{t\in I_\ell} v_t(F^*) \leq 8 \sum_{t=1}^{T} v_t(F^*),$$

where the first inequality follows from (11) and (12); and the last inequality follows from the fact that the intervals I_ℓ, $\ell \in [k+1]$, form a partition of $[T]$, by their definition. □

Lemma 4.4. *If $k^* > 0$, then $\sum_{\ell\in L} c^{(\ell)} + v^{(k+1)} \leq 7\Gamma k^*$.*

Proof. If $L = \emptyset$, then the desired result trivially holds. Thus, suppose that $L \neq \emptyset$. For $\ell \in L$, we have from the definition of $c^{(\ell)}$ that

$$c^{(\ell)} = \Gamma + v^{(\ell)} = \Gamma + \sum_{t\in I_\ell} v_t(F_t) = \Gamma + \sum_{t\in I_\ell\setminus\{t_\ell\}} v_t(F_t) + v_{t_\ell}(F_{t_\ell}) \leq 3\Gamma,$$

where the inequality follows from parts (i) and (iii) of Proposition 4.3. So, from (8), we have that

$$\sum_{\ell\in L} c^{(\ell)} + v^{(k+1)} \leq 6\Gamma k^* + v^{(k+1)} \leq 6\Gamma k^* + \Gamma \leq 7\Gamma k^*,$$

where the second inequality follows from part (i) of Proposition 4.3, and we use the fact that $k^* > 0$ in the last inequality. □

Finally, we are prepared to prove Theorem 4.1.

Proof of Theorem 4.1. We consider two cases, namely, when $k^* = 0$ and $k^* > 0$. We present an ad-hoc analysis for the first case, and then, we exploit the established results of this subsection to tackle the second case.

Case 1. Suppose that $k^* = 0$. Then, $F^* = \{0\}$, and consequently, $F^* \subseteq F_t$ for each $t \in [T]$. Thus, from part (i) of Proposition 4.2, we have that:

$$v_t(F_t) \leq v_t(F^*), \quad \forall t \in [T]. \tag{13}$$

For each $\ell \in [k]$, part (ii) of Proposition 4.3 implies that:

$$\Gamma \leq \sum_{t\in I_\ell} v_t(F_{t_{\ell-1}}) \leq \sum_{t\in I_\ell} v_t(F_t) \leq \sum_{t\in I_\ell} v_t(F^*), \tag{14}$$

where the second inequality follows from part (i) of Proposition 4.2 ($F_{t_{\ell-1}} \subseteq F_t$ for $t \in I_\ell$), and the last inequality follows from (13). Finally, we observe that:

$$c_T(F_1,\dots,F_T) = \sum_{\ell\in[k]} c^{(\ell)} + v^{(k+1)} = \sum_{\ell\in[k]} \Big(\Gamma + \sum_{t\in I_\ell} v_t(F_t)\Big) + \sum_{t\in I_{k+1}} v_t(F_t)$$

$$\leq 2 \sum_{\ell \in [k]} \sum_{t \in I_\ell} v_t(F^*) + \sum_{t \in I_{k+1}} v_t(F^*) \leq 2\Big(\sum_{\ell \in [k]} \sum_{t \in I_\ell} v_t(F^*) + \sum_{t \in I_{k+1}} v_t(F^*) \Big)$$
$$= 2 \sum_{t \in [T]} v_t(F^*) = 2\mathrm{OPT},$$

where the second equality above follows from the definitions of $c^{(\ell)}$, $\ell \in [k]$, and $v^{(k+1)}$; the first inequality follows from (13) and (14); the third equality follows from the fact that the intervals I_ℓ, $\ell \in [k+1]$, form a partition of $[T]$; the last equality follows from the fact that no facilities are located under F^* (recall the supposition that $k^* = 0$).

Case 2. Suppose that $k^* > 0$. From the cost decomposition outlined in (6), we obtain that:

$$c_T(F_1, \dots, F_T) = \sum_{\ell \in [k]} c^{(\ell)} + v^{(k+1)} = \sum_{\ell \in L} c^{(\ell)} + v^{(k+1)} + \sum_{\ell \in [k] \setminus L} c^{(\ell)}$$
$$\leq 7\Gamma k^* + 8 \sum_{t=1}^{T} v_t(F^*) \leq 8\Big(\Gamma k^* + \sum_{t=1}^{T} v_t(F^*)\Big) = 8\mathrm{OPT},$$

where the first inequality follows from Lemmas 4.3 and 4.4. □

4.3 Lower bounds on the competitive ratio

The instances we consider in the proof of Proposition 4.1 are similar to those used in the proof of Proposition 3.4. Specifically, we direct the reader to Figure 3 for an illustration of the instance for $T = 4$. The optimal fully offline solution locates a facility at the point x_4. It follows that $\mathrm{OPT} = (\Gamma/2)T + \Gamma - \epsilon$. CCT locates facilities at times $2, \dots, T-1$. Thus, the total costs incurred under CCT are given by $(2\Gamma - \epsilon)T - 2\Gamma$.

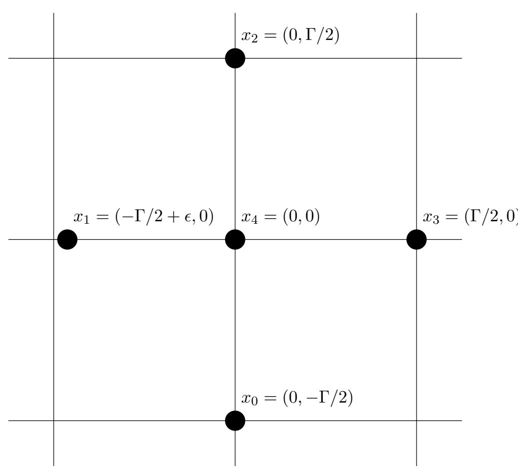


Figure 3: Illustration of the instance presented in the proof of Proposition 4.1 for $T = 4$. The metric for the instance is the taxicab metric, i.e., $d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$ for $x, y \in \mathbb{R}^n$.

Proof of Proposition 4.1. Let $n = \lceil T/2 \rceil$. Also, let e_i denote the i-th standard unit vector in $\mathbb{R}^n$ for each $i \in [n]$. Consider the following instance. Take $\hat{X} = \{e_i\}_{i\in[n]} \cup \{-e_i\}_{i\in[n]}$, and take $\hat{x}_0, \hat{x}_1, \ldots, \hat{x}_{T-1} \in \hat{X}$ to be distinct. Further take

$$x_t = \begin{cases} (\Gamma/2)\hat{x}_t & t = 0 \\ (\Gamma/2 - \epsilon)\hat{x}_t & t = 1 \\ (\Gamma/2)\hat{x}_t & t \in \{2, \ldots, T-1\} \\ 0 & t = T, \end{cases} \qquad t \in \{0\} \cup [T].$$

Finally, take the metric d to be 1-norm distance, i.e., $d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$ for $x, y \in \mathbb{R}^n$.

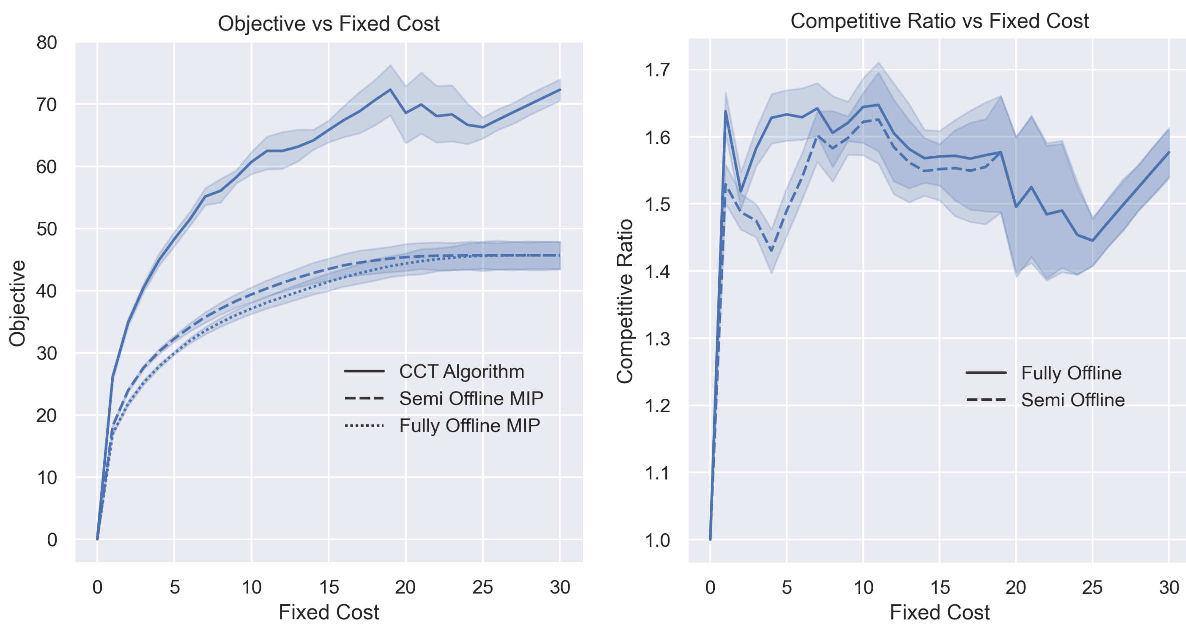
It is readily verified that $F^* = \{0, T\}$ is an optimal solution to the fully offline problem. The variable costs under F^* are given by $v_1(F^*) = \Gamma/2 - \epsilon$ and $v_t(F^*) = \Gamma/2$ for $t = 2, \ldots, T$. Thus, $\mathrm{OPT} = \Gamma(|F^*| - 1) + \sum_{t=1}^{T} v_t(F^*) = (\Gamma/2)T + \Gamma - \epsilon$.

It is also easily verified that CCT returns the solution $(F_1, \ldots, F_T)$ defined by $F_1 = \{0\}$ and $F_t = \{0, 2, \ldots, t\}$ for $t = 2, \ldots, T-1$, and $F_T = \{0, 2, \ldots, T-1\}$. The variable costs under $(F_1, \ldots, F_T)$ are given by $v_t(F_T) = \Gamma - \epsilon$, $t \in [T]$. Thus, the total cost incurred under CCT is given by $c_T(F_1, \ldots, F_T) = \Gamma(T-2) + (\Gamma - \epsilon)T = (2\Gamma - \epsilon)T - 2\Gamma$. The result then follows. □

5 Computational experiments

In this section, we support our theoretical developments by examining the empirical performance of the CCT algorithm and the offline models. First, in Subsection 5.1, we focus on synthetic test instances with randomly generated demand points. Next, in Subsection 5.2, we present our experiments using instances constructed from real-life demand points based on DHL facility locations. Note that some of the tables and figures are relegated to Appendix B due to space limitations.

Hardware and Software. Our computational study is performed on an HP Z440 Server running the Ubuntu Linux 20.04.5 LTS operating system equipped with Intel Xeon E5-1630 processor (CPU 3.70Ghz, 4 core, 8 thread) and 64 GB RAM. The mixed-integer programming formulations are all solved using Gurobi Optimizer 9.5 (via Gurobi's Python API), and Algorithm 1 is programmed using Python. We use Python to aid in instance generation and to interact with the Google Maps Distance Matrix API [1], which we use to preprocess our DHL facility location dataset.



(a) (b)

Figure 4: The synthetic online facility location instances — the average results for 30 instances with 95% confidence interval and $\Gamma \in \{0, 1, \dots, 30\}$. In (a), we depict the objective function values for all three considered methods; in (b), we depict the empirical competitive ratio of the CCT algorithm with respect to the fully offline and the semi-offline models; see (1) and (2), respectively.

5.1 Synthetic online facility location instances

We generate a testbed of synthetic instances as follows. First, we independently sample $T + 1 = 51$ demand points uniformly at random from the unit square $[0, 1]^2$. Recall our assumption that there is a facility already built at $x_0 \in X$ at time $t = 0$, which incurs no fixed cost; hence, exactly T demand points arrive at $t \in \{1, \dots, 50\}$. We then measure the distances between the points using the Euclidean metric. We repeat this procedure 30 times to obtain 30 sets of 51 demand points with the corresponding distances. Finally, we consider a range of fixed costs given by $\Gamma \in \{0, 1, \dots, 30\}$.

Next, we explore the performance of the CCT algorithm (i.e., Algorithm 1) as well as the corresponding fully offline (1) and semi-offline (2) optimization models. The results for the latter two are computed using MILPs (3) and (4), respectively.

Solution quality. We first depict the average objective function values (i.e., the total cost incurred up to and including time T) computed by all three considered approaches for $\Gamma \in \{0, 1, \dots, 30\}$; see Figure 4(a). Furthermore, we also report the empirical competitive ratio of CCT relative to the fully offline and semi-offline solutions; see Figure 4(b). That is, these ratios represent the objective

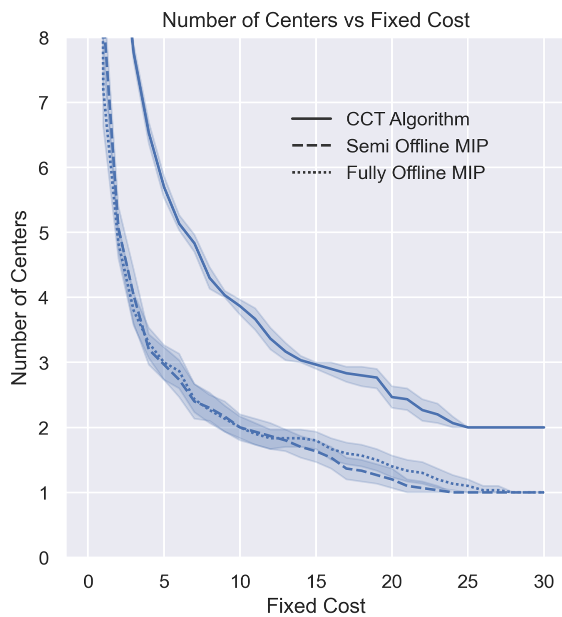


Figure 5: The synthetic online facility location instances – the average number of facilities located over 30 instances with 95% confidence interval for $\Gamma \in \{0, 1, \ldots, 30\}$. All methods locate facilities at all $T = 51$ demand points for $\Gamma = 0$. Thus, we limit the vertical range of the depicted results to 8 facilities. The average number of facilities built by CCT for $\Gamma \in \{1, 2\}$ is 14 and 9, respectively.

function values obtained by CCT relative to the optimal objective function values of the fully offline and semi-offline models, respectively; recall our discussion in Section 2.1.

Naturally, as the value of the fixed cost Γ increases the corresponding objective function values for all approaches tend to increase as well. Note that they are guaranteed to monotonically increase for the offline problems; however, it does not necessarily occur for the CCT solutions. This observation is quite intuitive given that CCT constructs more facilities than the optimal solutions of the offline problems (we explore this issue further below). Also, as expected, the average optimal objective function values of the offline problems (and the average objective function value of CCT to a lesser extent) eventually plateau as the fixed cost becomes increasingly prohibitive.

One notable observation is that for each value of Γ, the average optimal objective function values of the fully offline and semi-offline problems are relatively close to each other. That is, the ratio of these objective function values is significantly smaller than the worst-case ratio of 2 established in Proposition 3.3. This empirical evidence further supports our earlier claim that we are not being overly conservative in using the fully offline optimal objective function value for the theoretical competitive ratio in Theorem 4.1; recall also our discussion in Subsection 3.3.

Finally, perhaps, the most important observation in this set of experiments is that the average empirical competitive ratio of CCT is significantly smaller than the worst-case theoretical competitive ratio of 8 provided by Theorem 4.1. Recall from our discussion of Proposition 4.1 that it is unclear whether our analysis of CCT is tight. In fact, in Proposition 4.1, we establish an asymptotic lower bound of only 4 on the competitive ratio of CCT. Hence, there may be some empirical evidence suggesting that our main result in Theorem 4.1 could potentially be strengthened.

Solution structure. In Figure 5, we depict the average number of facilities constructed by each of the considered three methods as the fixed cost Γ increases. Unsurprisingly, as the value of Γ increases, each method locates fewer facilities.

For sufficiently large Γ, the offline solutions eventually locate no facilities, aside from the facility given at time 0 that comes at no fixed cost. This observation explains the plateau in Figure 4(a). Taking Γ to be larger, we would eventually also observe a plateau at one facility for CCT as well. For each value of Γ, CCT, on average, constructs more facilities than the offline solutions. Indeed, unlike the offline models, CCT does not have access to full information and tends to "over-build" at sub-optimal demand points early on, and possibly continues to add facilities when it may not lead to significant improvements in the total costs.

Solution times. We also investigate the solution times of the fully offline (3) and semi-offline (4) MILPs as well as CCT. For conciseness, all these results are relegated to Appendix B.

5.2 Locating DHL facilities in the Northeastern United States

We use a dataset from the Homeland Infrastructure Foundation-level Data database [2] that contains geographic coordinates and metadata for DHL facilities in the United States. The dataset includes information on multiple types of DHL facilities, namely, drop boxes, drop-off facilities, authorized shipping centers, and fully staffed DHL facilities. Drop boxes and drop-off facilities are express shipping locations for customers to drop off packages, and authorized shipping centers may be shared with other carriers; hence, we restrict our attention to fully-staffed DHL facilities.

In this set of experiments, we simulate the growth of the DHL network in the United States, with the company deciding which fully staffed DHL facilities to upgrade to *central* (or *hub*) facilities in an online fashion, as new DHL facilities continue to be built. In terms of our problem statement, the DHL facilities from the dataset serve as the demand points, and upgrading a DHL facility into a central (hub) facility corresponds to locating a facility at a demand point.

The original dataset contains 326 fully-staffed DHL facilities across the United States. In order to run an experiment with a manageable instance size for the fully offline MILPs (so we can evaluate the empirical competitive ratio of Algorithm 1), we restrict our attention to $T = 40$ DHL facilities from the Northeastern United States. We compute the road distances between DHL facilities using the the Google Maps Distance Matrix API.

The dataset does not provide information on the order in which the DHL facilities (demand points) were built (arrived). We generate three different arrival orders as follows. First, we select the initial demand point x_0 uniformly at random from the considered 40 locations. Then, we compute the following three arrival orders (using the same initial demand point x_0):

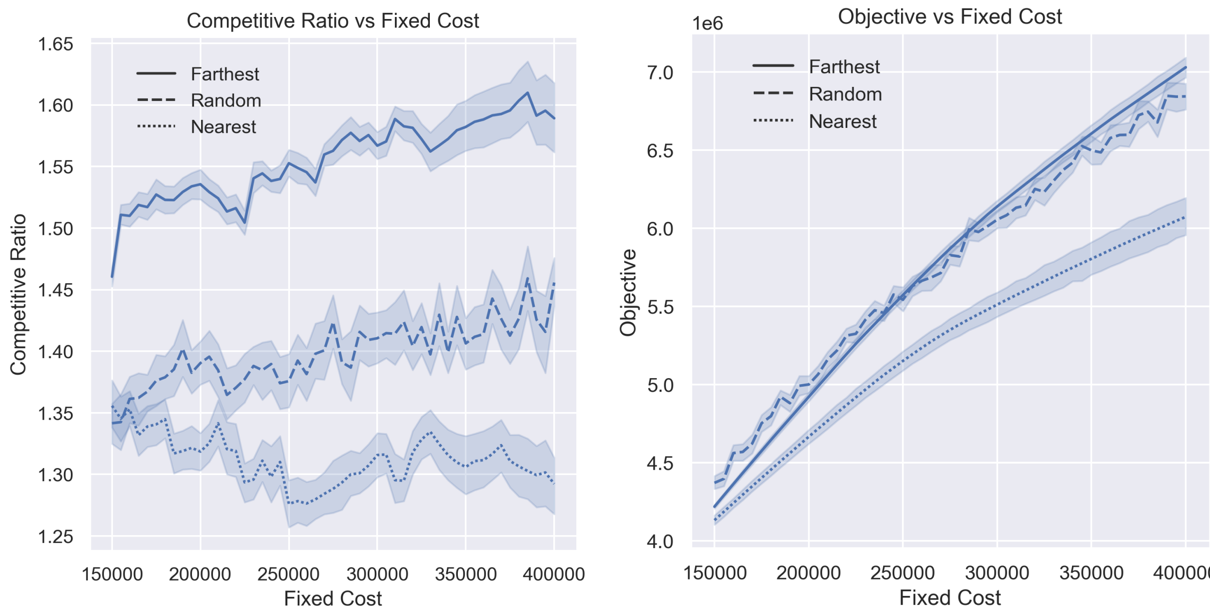
(i) *Random arrival order.* We sample the demand points $x_1, \ldots, x_{39}$ uniformly at random without replacement from the remaining 39 DHL facility locations.

(ii) *Nearest arrival order.* For $t \in [39]$, we successively take x_t to be a location amongst the remaining $40-t$ DHL facility locations that is nearest to the already chosen locations, i.e., we select x_t that minimizes $\min_{i \in \{0\} \cup [t-1]} d(x_t, x_i)$. One can argue that the nearest arrival order simulates a natural growth pattern for the industrial facility location application at hand.

(iii) *Farthest arrival order.* For $t \in [39]$, we successively take x_t to be a location amongst the remaining $40 - t$ DHL facility locations that is farthest to the already chosen locations, i.e., we select x_t that maximizes $\min_{i \in \{0\} \cup [t-1]} d(x_t, x_i)$. In a sense, the farthest arrival order simulates a pathological growth pattern of the underlying facility network.

We repeat the outlined randomization processes 30 times for each order type, obtaining a total of 90 different arrival orders. For each of these arrival orders, we consider fixed costs $\Gamma \in \{240{,}000, 245{,}000, \ldots, 400{,}000\}$, yielding the total number of instances to $90 \times 33 = 2{,}970$.

Solution quality. We plot the average empirical competitive ratio of CCT against the fixed cost Γ for all three arrival orders in Figure 6(a). It is quite intuitive that CCT performs best for the nearest arrival order, worst for the farthest arrival order, and somewhere in between for the random arrival order. Indeed, when the demand points arrive in the nearest arrival order, any facility that is upgraded to a hub facility is likely to be close to DHL facilities that are constructed soon after the upgrade, reducing the impact of sub-optimally locating hub facilities. On the other hand, when the demand points arrive in the farthest order, a facility that is upgraded to a hub facility is not guaranteed to be close to the DHL facilities that arrive soon after the hub facility is established. This discussion is somewhat reminiscent of the intuition behind the worst-case instances



(a) Competitive ratio of CCT

(b) Objective of fully offline model

Figure 6: DHL facilities in the Northeastern United States – the average results for 30 instances with 95% confidence interval and $\Gamma \in \{240{,}000, 245{,}000, \ldots, 400{,}000\}$. In (a), we depict the empirical competitive ratio of CCT for all three arrival orders; in (b), we depict the optimal objective function value of the fully offline model, see (1), for all three arrival orders.

constructed in the proofs of Propositions 3.4 and 4.1. Finally, it is worth pointing out that for this set of test instances the empirical competitive ratio of CCT is relatively similar to that reported for the synthetic test instances. As before, it is significantly smaller than the worst-case theoretical competitive ratio of 8 provided by Theorem 4.1.

In Figure 6(b), we depict the optimal objective function value of the fully offline model. Recall that the resulting costs depend on the arrival order. Interestingly, for smaller values of Γ, the costs associated with the farthest arrival order are lower than those of the random arrival order. This finding may be attributed to the fact that, for smaller values of Γ, the fully offline model upgrades more facilities. In general, the total costs associated with the farthest and random arrival orders are close to each other for the considered range of Γ. This observation perhaps explains why the corresponding empirical competitive ratios in Figure 6(a) exhibit a somewhat similar increasing trend.

Solution structure. We plot the optimal fully offline and CCT hub facility locations on the United States map for three different instances that each have different arrival orders, namely, random, nearest, and farthest; see Figures 7, 10 and 11, respectively. Due to space limitations and for brevity, the latter two figures are in Appendix B. In each of the instances, we use $\Gamma = 400{,}000$

and the hub facility given at time $t = 0$ to be in Manhattan, New York.

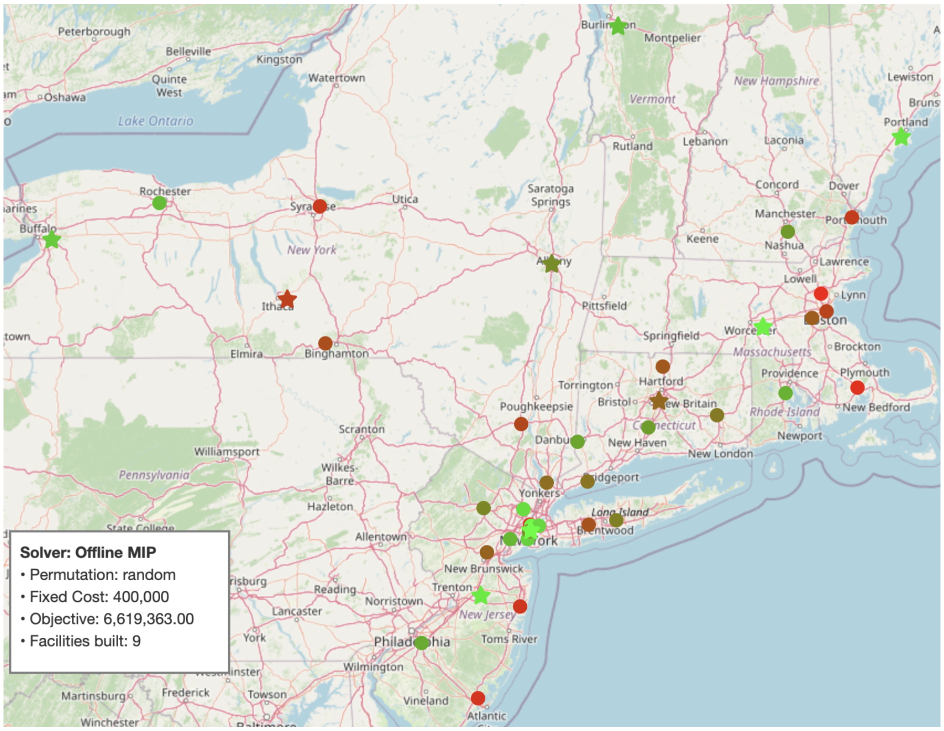
Unsurprisingly, CCT locates more hub facilities than the optimal fully offline solutions. Additionally, CCT locates fewest hub facilities for the nearest arrival order and most for the farthest arrival order. These observations are consistent with those discussed in relation to Figure 6(a). In the worst case, i.e., under the farthest arrival order, CCT locates almost twice as many more hub facilities than the optimal fully offline solution. However, the average ratio of the corresponding objective function values under CCT and the optimal fully offline solutions is not as large as this ratio of the number of facilities located. This finding implies that on average the total variable costs incurred under CCT is smaller than the total variable costs incurred under the optimal fully offline solution.

6 Concluding remarks

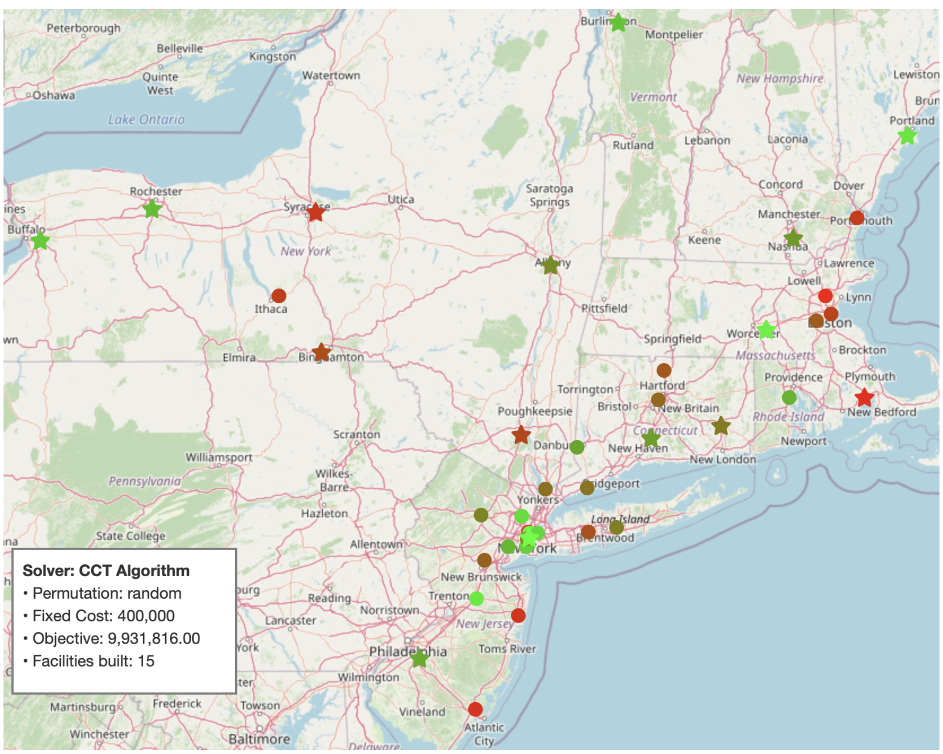
It is well known from the existing studies that it is not possible to develop constant-factor competitive algorithms for online facility location with one-time demand. For the one-time demand model, the underlying primary hurdle is that, after locating a facility at a certain point, few or even no future demand points are likely to arrive nearby. However, in the recurring demand model, all demand points need to be continuously served over time. Thus, the outlined issue is less of a concern. Motivated by this intuitive observation, we developed a constant-factor competitive algorithm for online facility location with recurring maximum demand. Specifically, we showed that the CCT algorithm is 8-competitive. Moreover, the proposed algorithm is rather intuitive, and, more importantly, simple enough to be used in practical settings.

Furthermore, we also studied the corresponding offline foundations. Perhaps, our most interesting observation is that the optimal objective function values of the fully offline and semi-offline problems are relatively close to each other, both theoretically and empirically in our experiments.

There are several intriguing directions for future research. First, it is naturally of interest to close the gap between the theoretical upper bound of 8 and the asymptotic lower bound of 4 for the CCT competitive ratio. Indeed, in our experiments with both the synthetic and real-life demand points, the average empirical competitive ratio of CCT is significantly smaller than the worst-case theoretical competitive ratio of 8. Furthermore, we focus on the *maximum* recurring demand, i.e., the worst-case demand in each arrival time, but one could consider other types of the recurring demand. For instance, one could take the variable costs to be sum-based rather than max-based. Finally, it may also be of interest to consider recurring cost structures in other online optimization contexts. More broadly, are problems with recurring cost structures easier to approximate?



(a) Fully offline model, random arrival order



(b) CCT, random arrival order

Figure 7: Northeastern United States with optimal fully offline and CCT solutions for the dataset with DHL facilities under random arrival order. Points in a star shape are hub facilities. Points are colored according to a linear gradient: earlier arrival points are colored more green, and later arrival points are colored more red.

References

[1] *Google Maps Distance Matrix API*, 2015. URL: `https://developers.google.com/maps/documentation/distance-matrix`, accessed August 2024.

[2] *DHL Facilities*, 2016. URL: `https://data.world/dhs/dhl-facilities`, accessed June 2024.

[3] S. Albers, *Online algorithms: a survey*, Mathematical Programming, 97 (2003), pp. 3–26.

[4] A. Anagnostopoulos, R. Bent, E. Upfal, and P. Van Hentenryck, *A simple and deterministic competitive algorithm for online facility location*, Information and Computation, 194 (2004), pp. 175–202.

[5] M. O. Ball and M. Queyranne, *Toward robust revenue management: Competitive analysis of online booking*, Operations Research, 57 (2009), pp. 950–963.

[6] N. Buchbinder, T. Kimbrel, R. Levi, K. Makarychev, and M. Sviridenko, *Online make-to-order joint replenishment model: Primal-dual competitive algorithms*, Operations Research, 61 (2013), pp. 1014–1029.

[7] J. Byrka and K. Aardal, *An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem*, SIAM Journal on Computing, 39 (2010), pp. 2212–2231.

[8] J. Byrka, T. Pensyl, B. Rybicki, A. Srinivasan, and K. Trinh, *An improved approximation for k-median and positive correlation in budgeted optimization*, ACM Transactions on Algorithms, 13 (2017).

[9] D. Celik Turkoglu and M. Erol Genevois, *A comparative survey of service facility location problems*, Annals of Operations Research, 292 (2020), pp. 399–468.

[10] M. Charikar, C. Chekuri, T. Feder, and R. Motwani, *Incremental clustering and dynamic information retrieval*, SIAM Journal on Computing, 33 (2004), pp. 1417–1440.

[11] M. Charikar and S. Guha, *Improved combinatorial algorithms for facility location problems*, SIAM Journal on Computing, 34 (2005), pp. 803–824.

[12] V. Cohen-Addad, N. O. D. Hjuler, N. Parotsidis, D. Saulpic, and C. Schwiegelshohn, *Fully dynamic consistent facility location*, Advances in Neural Information Processing Systems, 32 (2019).

[13] A. R. Conn and G. CornuéJols, *A projection method for the uncapacitated facility location problem*, Mathematical Programming, 46 (1990), pp. 273–298.

[14] G. Cornuéjols, G. Nemhauser, and L. Wolsey, *The uncapicitated facility location problem*, tech. rep., Cornell University Operations Research and Industrial Engineering, 1983.

[15] Z. Drezner and H. W. Hamacher, *Facility Location: Applications and Theory*, Springer Berlin, Heidelberg, 2004.

[16] C. Escribe, M. Hu, and R. Levi, *Competitive algorithms for the online minimum peak job scheduling*, Operations Research, (2023).

[17] A. Fiat and G. J. Woeginger, *Online algorithms: The state of the art*, vol. 1442, Springer, 1998.

[18] H. Fichtenberger, S. Lattanzi, A. Norouzi-Fard, and O. Svensson, *Consistent k-clustering for general metrics*, in Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, 2021, pp. 2660–2678.

[19] D. Fotakis, *Incremental algorithms for facility location and k-median*, Theoretical Computer Science, 361 (2006), pp. 275–313.

[20] ——, *A primal-dual algorithm for online non-uniform facility location*, Journal of Discrete Algorithms, 5 (2007), pp. 141–148.

[21] ——, *On the competitive ratio for online facility location*, Algorithmica, 50 (2008), pp. 1–57.

[22] ——, *Online and incremental algorithms for facility location*, ACM SIGACT News, 42 (2011), pp. 97–131.

[23] J. Garcia-Diaz, J. Sanchez-Hernandez, R. Menchaca-Mendez, and R. Menchaca-Mendez, *When a worse approximation factor gives better performance: A 3-approximation algorithm for the vertex k-center problem*, Journal of Heuristics, 23 (2017), pp. 349–366.

[24] T. F. Gonzalez, *Clustering to minimize the maximum intercluster distance*, Theoretical Computer Science, 38 (1985), pp. 293–306.

[25] S. Guha and S. Khuller, *Greedy strikes back: Improved facility location algorithms*, Journal of Algorithms, 31 (1999), pp. 228–248.

[26] X. Guo, J. Kulkarni, S. Li, and J. Xian, *Consistent k-median: Simpler, better and robust*, in International Conference on Artificial Intelligence and Statistics, PMLR, 2021, pp. 1135–1143.

[27] D. S. Hochbaum, *Heuristics for the fixed cost median problem*, Mathematical Programming, 22 (1982), pp. 148–162.

[28] D. S. Hochbaum and D. B. Shmoys, *A best possible heuristic for the k-center problem*, Mathematics of Operations Research, 10 (1985), pp. 180–184.

[29] P. Jaillet and M. R. Wagner, *Generalized online routing: New competitive ratios, resource augmentation, and asymptotic analyses*, Operations Research, 56 (2008), pp. 745–757.

[30] K. Jain and V. V. Vazirani, *Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation*, Journal of the ACM, 48 (2001), p. 274–296.

[31] J. Lamperski, O. A. Prokopyev, and L. G. Wrabetz, *Min-max-min optimization with smooth and strongly convex objectives*, SIAM Journal on Optimization, 33 (2023), pp. 2435–2456.

[32] S. Lattanzi and S. Vassilvitskii, *Consistent k-clustering*, in International Conference on Machine Learning, PMLR, 2017, pp. 1975–1984.

[33] S. Li, *A 1.488 approximation algorithm for the uncapacitated facility location problem*, Information and Computation, 222 (2013), pp. 45–58. 38th International Colloquium on Automata, Languages and Programming (ICALP 2011).

[34] E. Liberty, R. Sriharsha, and M. Sviridenko, *An algorithm for online k-means clustering*, in 2016 Proceedings of the Eighteenth Workshop on Algorithm Engineering and Experiments (ALENEX), SIAM, 2016, pp. 81–89.

[35] J.-H. Lin and J. S. Vitter, *e-approximations with minimum packing constraint violation (extended abstract)*, in Proceedings of the Twenty-Fourth Annual ACM Symposium on Theory of Computing, 1992, pp. 771—-782.

[36] W. Ma and D. Simchi-Levi, *Algorithms for online matching, assortment, and pricing with tight weight-dependent competitive ratios*, Operations Research, 68 (2020), pp. 1787–1803.

[37] C. Markarian, *Online algorithmic study of facility location problems: A survey*, IEEE Access, (2024).

[38] A. Meyerson, *Online facility location*, in Proceedings 42nd IEEE Symposium on Foundations of Computer Science, IEEE, 2001, pp. 426–431.

[39] D. B. Shmoys, *Computing near-optimal solutions to combinatorial optimization problems*, in Combinatorial Optimization, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, AMS Publications, 1995, pp. 355–397.

[40] D. B. Shmoys, É. Tardos, and K. Aardal, *Approximation algorithms for facility location problems*, in Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing, 1997, pp. 265–274.

[41] V. Verter, *Uncapacitated and Capacitated Facility Location Problems*, Springer US, New York, NY, 2011, pp. 25–37.

[42] L. A. Wolsey and G. L. Nemhauser, *Integer and combinatorial optimization*, John Wiley & Sons, 2014.

A Proofs omitted in the main body

A.1 Proof of Proposition 3.1

As mentioned earlier, the proof of Proposition 3.1 uses a reduction from the dominating set problem. We present a description of the dominating set problem below. Also, see Figure 8 for an illustration of the BFO decision instance that we construct in the proof of Proposition 3.1.

Dominating set problem:

- *Instance.* A graph $G = (V, E)$ and target dominating set size $k \in \mathbb{Z}_{>0}$.
- *Question.* Is there a dominating set of cardinality no greater than k in G?

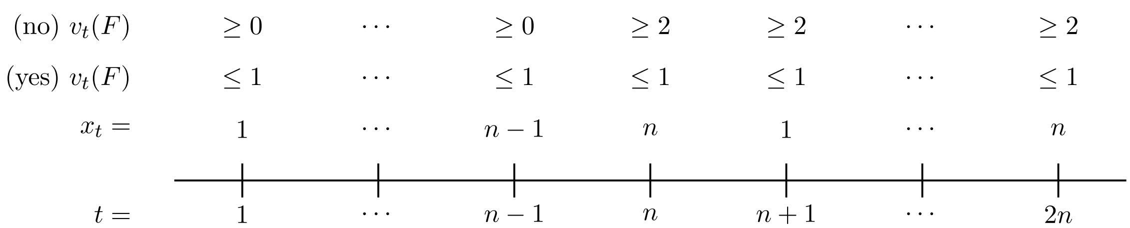


Figure 8: Illustration of the BFO decision instance constructed in the proof of Proposition 3.1.

Proof of Proposition 3.1. Let (G, k) be an instance of the dominating set problem. Suppose that $G = ([n], E)$, i.e., G has n vertices that are labeled by the set $[n]$. We construct an instance of the BFO decision problem from (G, k) as follows. Take $X = \{0\} \cup [n]$, $T = 2n$, and

$$x_i = \begin{cases} 0 & i = 0 \\ i & 1 \leq i \leq n \\ i - n & n+1 \leq i \leq 2n, \end{cases} \qquad i \in \{0\} \cup [2n].$$

That is, vertices $1, \ldots, n$ arrive at times $1, \ldots, n$, and they also arrive at times $n+1, \ldots, 2n$. Next, take d to be the metric such that $d(x_i, x_j) = 1$ if $x_i, x_j \in [n]$ and x_i is adjacent to x_j in G, and $d(x_i, x_j) = 2$, otherwise. It is readily verified that d is a metric. Finally, take k to be the budget parameter, and take $\alpha = 2n$ to be the target objective value.

Suppose that G has a dominating set $F' \subseteq [n]$ such that $|F'| \leq k$. Let $F = F' \cup \{0\} \in \mathcal{F}_{BFO}$. Then, for each vertex $u \in [n] \setminus F$, there exists $v \in F' \subseteq F$ such that $d(u, v) \leq 1$. Because each point that arrives is one of the vertices, it follows that

$$v_t(F) = \max_{i \in [t]} \min_{j \in F} d(x_i, x_j) \leq 1$$

for all $t \in [T]$. Consequently, $\sum_{t=1}^{T} v_t(F) \leq T = \alpha$, and the answer to the BFO decision problem instance is yes, as desired.

Suppose that G does not have a dominating set of size at most k. Then, for any $F \in \mathcal{F}_{BFO}$, it holds that $v_t(F) = 2$ for all $n \leq t \leq 2n$ because all vertices arrive (for the first time) by time n. Thus, for all $F \in \mathcal{F}_{BFO}$,

$$\sum_{t=1}^{T} v_t(F) = \sum_{t=1}^{2n} v_t(F) \geq \sum_{t=n}^{2n} v_t(F) = 2n + 2 > 2n = \alpha,$$

so the answer to the BFO decision problem instance is no, as desired. □

A.2 Proof of Proposition 3.2

Our proof of Proposition 3.2 is similar to our proof of Proposition 3.1. To account for the fact that facilities cannot be built at time 0 in the semi-offline problem, we construct a BSO decision instance with $T = 3n$ points instead of $2n$ points as in the proof of Proposition 3.1. Given that the proofs are similar, we do not include as many details.

Proof of Proposition 3.2. Let (G, k) be an instance of the dominating set problem. Suppose that $G = ([n], E)$, i.e., G has n vertices that are labeled by the set $[n]$. We construct an instance of the BSO decision problem from (G, k) as follows. Take $X = \{0\} \cup [n]$, $T = 3n$, and

$$x_i = \begin{cases} 0 & i = 0 \\ i & 1 \leq i \leq n \\ i - n & n + 1 \leq i \leq 2n \\ i - 2n & 2n + 1 \leq i \leq 3n \end{cases} \qquad i \in \{0\} \cup [3n].$$

Next, take d to be the metric such that $d(x_i, x_j) = 1$ if $x_i, x_j \in [n]$ and x_i is adjacent to x_j in G, and $d(x_i, x_j) = 2$, otherwise. It is readily verified that d is a metric. Finally, take k to be the budget parameter, and take $\alpha = 4n - 1$ to be the target objective value.

Suppose that G has a dominating set $F' \subseteq [n]$ such that $|F'| \leq k$. For $t \in [T]$, let $F_t = \{0\} \cup \{i \in$

$F' : i \leq t\}$, and consider the solution $(F_1, \ldots, F_T) \in \mathcal{F}_{BSO}$. Because F' is a dominating set in G, it holds that $v_t(F_1, \ldots, F_t) \leq 1$ for all $n \leq t \leq 3n$. Because $v_t(F_1, \ldots, F_t) \leq 2$ for all $t \in [n-1]$, we further have that

$$\sum_{t=1}^{T} v_t(F_1, \ldots, F_t) \leq 2(n-1) + 2n + 1 = 4n - 1 = \alpha,$$

so the answer to the BSO decision problem instance is yes, as desired.

Suppose that G does not have a dominating set of size at most k. Then, for any $(F_1, \ldots, F_T) \in \mathcal{F}_{BSO}$, it holds that $v_t(F_1, \ldots, F_t) \geq 2$ for all $n \leq t \leq 3n$. Thus,

$$\sum_{t=1}^{T} v_t(F_1, \ldots, F_t) \geq 4n + 2 > \alpha,$$

so the answer to the BSO decision problem instance is no, as desired. □

A.3 Proof of Proposition 3.3

Proof of Proposition 3.3. Let $F^* \subseteq \{0\} \cup [T]$ be an optimal solution to the fully offline problem (1). Define

$$C_j^* := \left\{ i \in \{0\} \cup [T] : j \in \operatorname*{argmin}_{\ell \in F^*} d(x_i, x_\ell) \right\}, \quad j \in F^*$$

to be the clusters that are induced by F^*. That is, $i \in C_j^*$ if and only if the facility at x_j is a facility that is nearest to point x_i. Because we do not break the ties in any way, the clusters do not necessarily partition $\{0\} \cup [T]$, but they do cover $\{0\} \cup [T]$. Also, define

$$a_j := \min\{t \in \{0\} \cup [T] : t \in C_j^*\}$$

to be the first time that a point indexed by cluster C_j^* arrives; clearly, $a_0 = 0$. Note that we define the same notions of optimal clusters and first arrival times in Subsections 4.1 and 4.2, respectively. Below we employ Proposition 4.4 from Subsection 4.1. Consider the solution $(F_1, \ldots, F_T)$ to the semi-offline problem defined by

$$F_t = \{a_j : j \in F^* \text{ and } a_j \leq t\}, \quad t \in [T].$$

The solution locates a facility at the first point to arrive in each cluster induced by the optimal fully offline solution.

First, we show that

$$v_t(F_t) \leq 2v_t(F^*) \tag{15}$$

for each $t \in [T]$. It is sufficient to show that $\min_{j \in F_t} d(x_i, x_j) \leq 2v_t(F^*)$, where $i \in [t]$. Take $\ell \in F^*$ such that $i \in C^*_\ell$. Then $a_\ell \in F_t$ because point x_i arrives before or at time t. Hence,

$$\min_{j \in F_t} d(x_i, x_j) \leq d(x_i, x_{a_\ell}) \leq d(x_i, x_\ell) + d(x_\ell, x_{a_\ell}) \leq 2v_t(F^*),$$

where the second inequality follows from the triangle inequality, and the last inequality follows from Proposition 4.4 (which we can apply because $i, a_\ell \in C^*_\ell$ and $i, a_\ell \leq t$). Thus, (15) holds.

Finally, we observe that

$$\begin{aligned} \mathrm{OPT}_{\mathrm{semi}} \leq \Gamma(|F_T| - 1) + \sum_{t=1}^{T} v_t(F_t) &\leq \Gamma(|F^*| - 1) + 2\sum_{t=1}^{T} v_t(F^*) \\ &\leq 2\left(\Gamma(|F^*| - 1) + \sum_{t=1}^{T} v_t(F^*)\right) = 2\mathrm{OPT}, \end{aligned}$$

where the second inequality follows from $|F_T| = |F^*|$ and (15). □

A.4 Proof of Proposition 3.4

Proof of Proposition 3.4. Let $n = \lceil T/2 \rceil$, and define e_i to be the i-th standard unit vector in $\mathbb{R}^n$ for each $i \in [n]$. Consider the following instance. Take $X = \{e_i\}_{i \in [n]} \cup \{-e_i\}_{i \in [n]}$, take $x_0, x_1, \ldots, x_{T-1} \in X$ to be distinct, and take $x_T = 0 \in \mathbb{R}^n$ to be the origin. Finally, take $\Gamma = 2$ and the metric d to be 1-norm distance, i.e., $d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$ for $x, y \in \mathbb{R}^n$.

It is readily verified that $F^* = \{0, T\}$ is an optimal solution to the fully offline problem. Because $v_t(F^*) = 1$ for each $t \in [T]$, we have that $\mathrm{OPT} = 2 + T$.

It is also easily verified that the solution defined by $F^*_t = \{0\}$, $t \in [T]$ (i.e., do not construct any centers) is an optimal solution to the semi-offline problem. Because $v_t(F^*_t) = 2$ for $t \in [T]$, we have that $\mathrm{OPT}_{\mathrm{semi}} = 2T$. The proposition then follows. □

B Computational experiments: additional results

We explore the scalability of the fully offline MILP (3) and the semi-offline MILP (4) formulations using an off-the-shelf solver; the details on the software and hardware used in our experiments are outlined in Section 5. Note that the fully offline MILP has $O(T^2)$ variables and $O(T^2)$ constraints, and the semi-offline MILP has $O(T^3)$ variables and $O(T^3)$ constraints.

In Figure 9 we plot the average solution times for the two considered MILPs with the test instances constructed in Subsection 5.1. In the first plot of the figure, we plot against the fixed cost Γ, and in the second plot of the figure, we plot against the number of demand points (i.e., we solve subsets of the instances we generated, using only the first points), fixing the fixed cost $\Gamma = 6$.

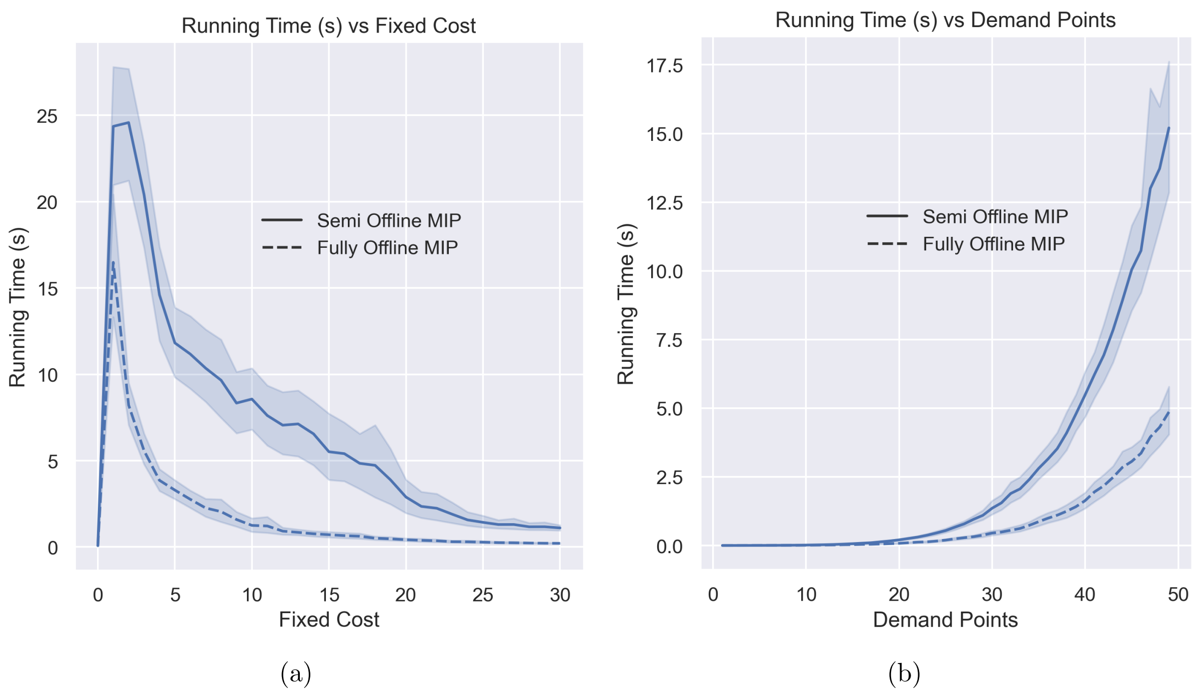


Figure 9: Average solution times in seconds (over 30 instances with 95% confidence interval) for the fully offline MILP (3) and semi-offline MILP (4) against (a) the fixed cost, Γ, and (b) the number of the demand points.

Interestingly, the average solution time for both MILPs at first grows rapidly as a function of the fixed cost and then decreases at a slower rate. This behavior can perhaps be explained by the fact that it "easy" to locate facilities when the fixed costs are either sufficiently small (locate facilities at every demand point) or sufficiently large (do not locate any facilities). As expected, the average solution times for both MILPs increases with the number of demand points. Furthermore, the average solution time of the semi-offline MILP is greater than that of the fully offline MILP, which is quite intuitive given the differences in the sizes of the formulations.

The MILP formulations are solved to optimality in under 30 seconds for the test instances in Subsection 5.1. Hence, we generate additional instances with $T \in \{75, 100, 200\}$ demand points. Using Figure 9 as a guideline, we generate instances with the most "challenging" fixed cost values. Specifically, we take $\Gamma \in \{3, 5, 10\}$. For each combination (T, Γ), we generate 30 instances, obtaining $3 \times 3 \times 30 = 270$ instances in total. For each instance, we solve the corresponding fully offline and semi-offline MILP formulations. We also apply CCT to each instance. Finally, we set a time limit equal to 1 hour.

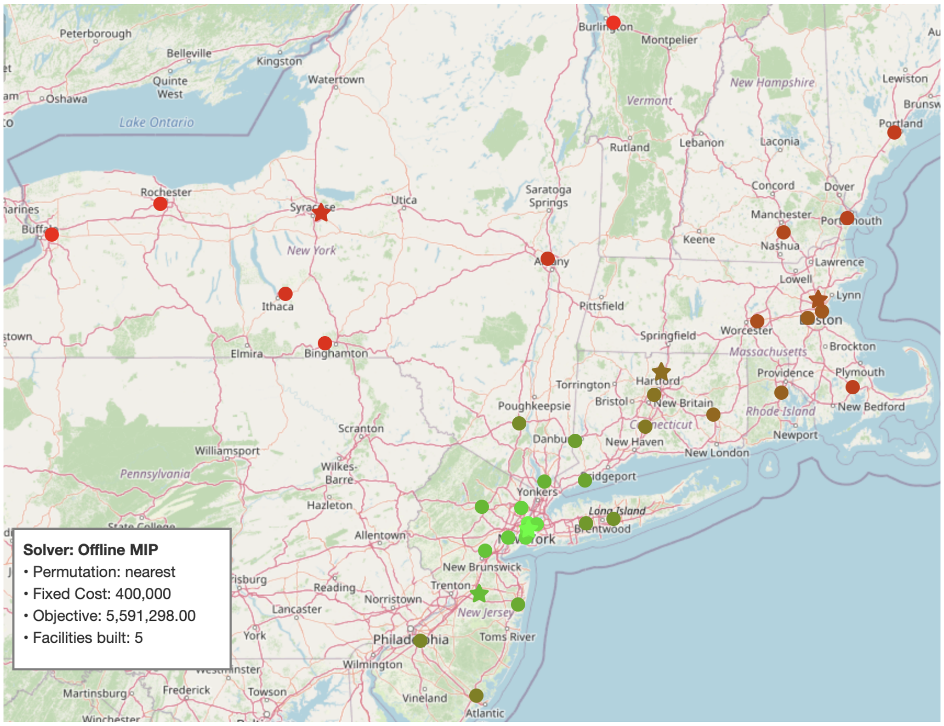
Table 1 displays the results for the instances with $T = 75$ and $T = 100$ demand points. In addition to the average running time, we report the percentage of the 30 instances that were solved to optimality within the time limit. Table 2 displays the same results for instances with $T = 200$ demand points. Table 2 does not include results for the semi-offline MILP because we were not able to solve the semi-offline MILP for any instances with 200 demand points.

Instance		**CCT**		**Fully Offline**		**Semi Offline**	
Points	Fixed Cost	Time (s)	Solved (%)	Time (s)	Solved (%)	Time (s)	Solved (%)
	3	0.04	100	58.80	100	484.16	100
75	5	0.04	100	31.41	100	404.97	100
	10	0.04	100	12.08	100	234.45	100
	3	0.05	100	402.40	100	2487.89	77
100	5	0.05	100	147.95	100	2787.46	67
	10	0.05	100	69.56	100	2200.71	87

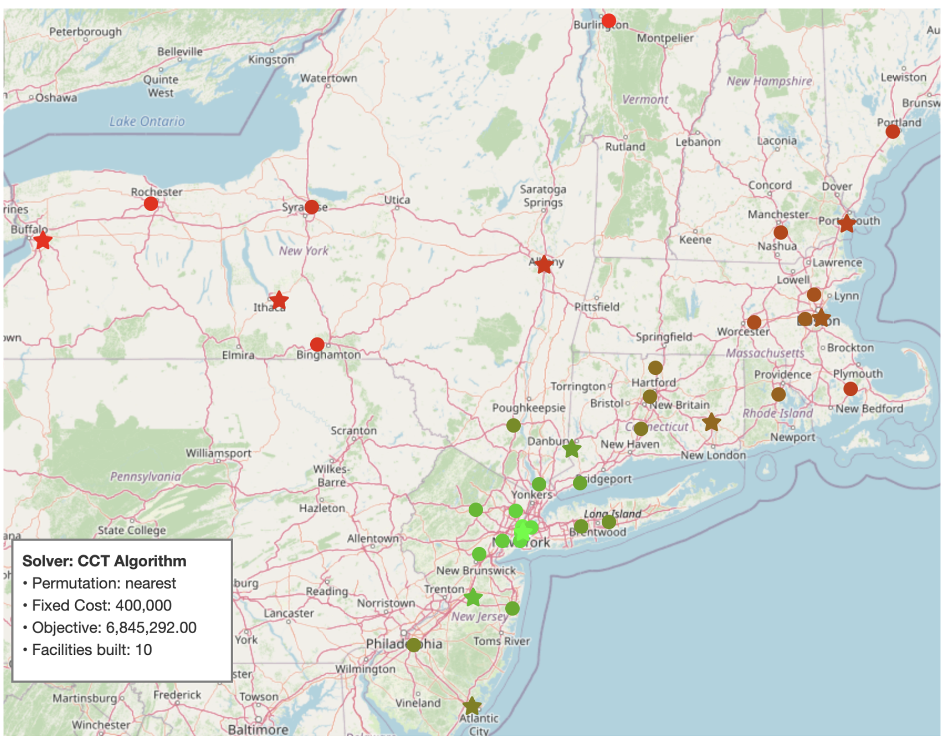
Table 1: The average running time (in seconds) and the percentage of the instances solved to optimality within a 1 hour time limit over 30 randomly generated instances of size $T \in \{75, 100\}$ with $\Gamma \in \{3, 5, 10\}$.

Instance		**CCT**		**Fully Offline**	
Points	Fixed Cost	Time (s)	Solved (%)	Time (s)	Solved (%)
	3	0.14	100	3600.00	0
200	5	0.13	100	3600.00	0
	10	0.13	100	2807.16	57

Table 2: The average running time (in seconds) and the percentage of the instances solved to optimality within a 1 hour time limit over 30 randomly generated instances of size $T = 200$ with $\Gamma \in \{3, 5, 10\}$.

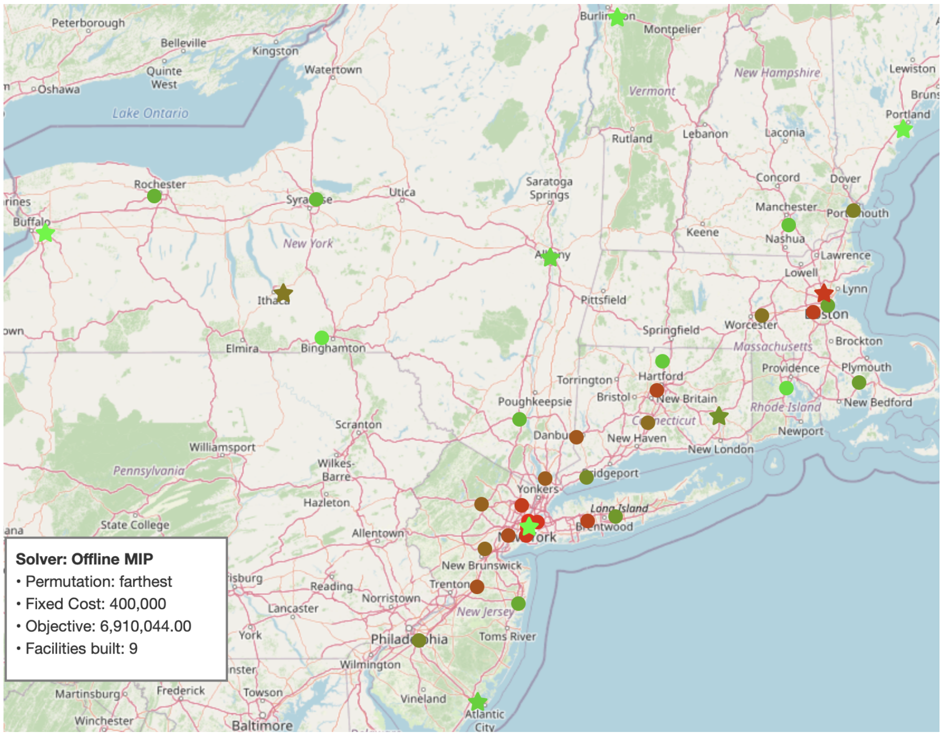


(a) Fully offline model, nearest arrival order

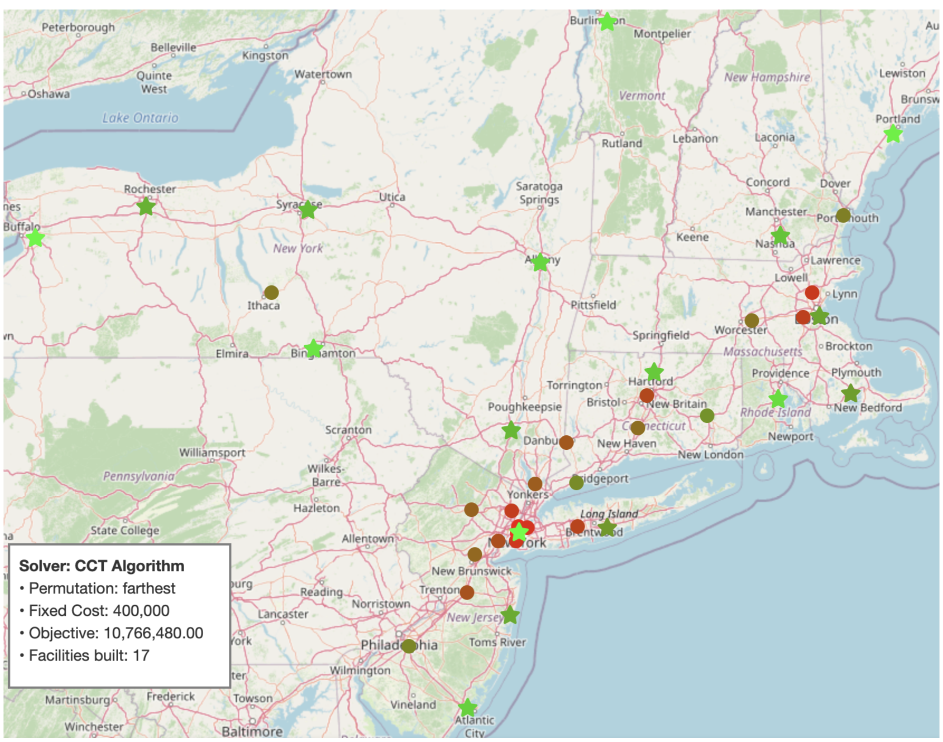


(b) CCT, nearest arrival order

Figure 10: Northeastern United States with optimal fully offline and CCT solutions for the dataset with DHL facilities under nearest arrival order. Points in a star shape are hub facilities. Points are colored according to a linear gradient: earlier arrival points are colored more green, and later arrival points are colored more red.



(a) Fully offline model, farthest arrival order



(b) CCT, farthest arrival order

Figure 11: Northeastern United States with optimal fully offline and CCT solutions for the dataset with DHL facilities under farthest arrival order. Points in a star shape are hub facilities. Points are colored according to a linear gradient: earlier arrival points are colored more green, and later arrival points are colored more red.