Delayed Assignments in Online Non-Centroid Clustering with Stochastic Arrivals

Saar Cohen
Department of Computer Science, Bar Ilan University
Ramat Gan, Israel
Department of Computer Science, University of Oxford
Oxford, United Kingdom
saar30@gmail.com

ABSTRACT

Clustering is a fundamental problem, aiming to partition a set of elements, like agents or data points, into clusters such that elements in the same cluster are closer to each other than to those in other clusters. In this paper, we present a new framework for studying online non-centroid clustering *with delays*, where elements, that arrive one at a time as points in a finite metric space, should be assigned to clusters, but assignments need *not* be immediate. Specifically, upon arrival, each point's location is revealed, and an online algorithm has to *irrevocably* assign it to an existing cluster or create a new one containing, at this moment, only this point. However, we allow decisions to be postponed at a *delay cost*, instead of following the more common assumption of *immediate* decisions upon arrival. This poses a critical challenge: the goal is to minimize both the total distance costs between points in each cluster and the overall delay costs incurred by postponing assignments. In the classic *worst-case arrival model*, where points arrive in an *arbitrary* order, no algorithm has a competitive ratio better than sublogarithmic in the number of points. To overcome this strong impossibility, we focus on a *stochastic arrival model*, where points' locations are drawn independently across time from an *unknown* and *fixed* probability distribution over the finite metric space. We offer hope for beyond worst-case adversaries: we devise an algorithm that is ***constant*** competitive in the sense that, as the number of points grows, the ratio between the expected overall costs of the output clustering and an optimal offline clustering is bounded by a constant.

KEYWORDS

Clustering, Online Algorithms, Stochastic Arrival Models

ACM Reference Format:
Saar Cohen. 2026. Delayed Assignments in Online Non-Centroid Clustering with Stochastic Arrivals. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 10 pages. https://doi.org/10.65109/AVAR7513

1 INTRODUCTION

In multiplayer online gaming platforms, players enter the platform over time and are assigned to teams for cooperative gameplay (e.g., completing quests or participating in tournaments). Players usually prefer teammates with similar skill levels, playstyles, and compatible roles. Once a player enters the platform, the platform's team assignment system may postpone the assignment in the hope that more compatible players will later join the platform. However, players may become highly unsatisfied if they wait for too long. To optimize their gaming experience, the system should assign players to teams so that players in the same team are more similar to each other than to players in other groups, while minimizing their waiting times for assignments.

A widely studied model of clustering is *center-based* metric clustering [31], where elements are points in a metric space whose similarity is measured by distance: closer elements are considered more similar. Clusters are defined by centers, with each point assigned to its nearest center. In contrast, we herein focus on *non-centroid* metric clustering (see, e.g., [11]), with no cluster centers. This captures real-world scenarios where elements (e.g., agents) prefer to be close to others in their cluster, like our gaming platform example, clustered federated learning [55], clustering in social networks [48], and document clustering [43]. Other examples include package-delivery services and ride-sharing platforms. For instance, in last-mile delivery, parcels arrive over time and must be dynamically grouped into vehicle loads, where grouping items with nearby destinations reduces travel distance and improves efficiency. Similarly, in ride-sharing platforms, passengers arrive sequentially and the system must form small groups of riders whose pickup and drop-off locations are mutually compatible. Delaying assignment improves batching opportunities but increases waiting time, exactly the tradeoff modeled in our framework.

Prior work on non-centroid clustering assumes an *offline* setting, where all elements are known in advance. However, in many real-world scenarios such as our gaming platform example, the input is revealed gradually over time. For that reason, various online clustering problems have received increased attention in recent years (see, e.g., [24, 39, 40, 46]), yet online *non-centroid metric* clustering has remained largely overlooked. Further, prior works on online clustering often assume that elements are assigned to clusters *immediately* upon arrival, which may lead to poor solutions in practice. For instance, in our gaming platform example, a perfectly compatible teammate might join the platform just after a player has already been assigned to a team.

In this paper, we thus introduce and study a new model of *online* non-centroid metric clustering *with delays*, where elements that arrive one at a time as points in a finite metric space must be assigned to clusters, but assignments need *not* be immediate. Upon arrival, each point's location is revealed. Then, a central authority (i.e., an

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/AVAR7513

online algorithm) has to *irrevocably* decide whether to assign it to an existing cluster or create a new one containing, at this moment, only this point. However, we allow decisions to be postponed at a *delay cost*, departing from the more common assumption of *immediate* decisions upon arrival. This raises a key challenge: our goal is to minimize both the total distance costs between points in each cluster and the total delay costs incurred by postponing decisions.

If clusters can be of any size, then the non-centroid clustering of singletons is trivially optimal, as it yields zero total cost. The same applies even if we instead seek a *capacitated* clustering, where each cluster must respect an upper bound on its size (see, e.g., [27, 54]). Hence, we focus on clustering with *fixed-size* clusters, where each cluster is required to have a predetermined size (see, e.g., [37]). This reflects many real-world scenarios where lack of size constraints can result in highly skewed clusters with arbitrary sizes, where one cluster is significantly larger than the others with small sizes, thus undermining their practical usefulness. For instance, capacity constraints naturally arise from limitations such as the number of customers an individual salesperson can effectively serve.

When the number of points is restricted to be *even* and each cluster must be of size *exactly* two, our problem reduces to the *Min-cost Perfect Matching with Delays* (MPMD) problem [6, 30]. MPMD is often studied under the classic *worst-case (adversarial) model*, modeling agents' arrival order and locations as controlled by an *adversary*, whose goal is degrading an algorithm's performance. Here, the performance of an online algorithm is often measured in terms of its *competitive ratio* [32], defined as the worst-case ratio between the algorithm's total cost and that of an optimal offline solution. However, the worst-case model is overly pessimistic: even if the metric space is known upfront, no online algorithm has a constant competitive ratio [5].

Contributions. To enable the design of effective algorithms in practice, we consider the *unknown i.i.d.* (UIID) model, where points' locations are drawn independently across time from an *unknown* and *fixed* probability distribution over the finite metric space. Under the UIID model, we measure the performance of an online algorithm in terms of its *ratio-of-expectations* (RoE) [50], evaluating the expected total costs of an online algorithm with that of an offline optimal clustering, as the number of points increases. This asymptotic perspective captures how an algorithm performs in large-scale and practical scenarios, rather than small pathological worst-case cases that may not truly reflect its efficiency. An algorithm's RoE is at least 1, with better performance corresponding to a smaller RoE. We offer hope for beyond worst-case adversaries: we devise a greedy algorithm with a ***constant*** ratio-of-expectations. Finally, we show how our results easily extend to non-metric spaces and cases with lower and upper bounds on cluster sizes.

2 RELATED WORK

Center-based metric clustering and its variants with cluster size constraints have been extensively studied in both *offline* settings [1, 13, 37, 47, 53], and *dynamic* settings [3, 12, 15, 42]. Center-based metric clustering is well-suited for applications such as facility location [4, 29, 35, 41], where elements correspond to agents and facilities, while cluster centers are potential sites for public infrastructure (e.g., parks), with each agent preferring the facility closest to them. However, in this work we focus on *non-centroid* metric clustering (see, e.g., [11]), where there are no cluster centers.

Closely related to our work is the correlation clustering problem introduced by Bansal et al. [8], where the input is a weighted graph so that a weight's sign indicates, for each pair of points, whether they should be placed in the same cluster. The goal is then to find a clustering that minimizes the number of disagreements with these pairwise recommendations. In *offline* settings, the problem is NP-hard and admits several approximation algorithms (see, e.g., [14, 26, 38]). In *online* settings under the classic worst-case model, it has been proven that the competitive ratio of any online algorithm is at least $\Omega(n)$ for a graph with n nodes. To escape this impossibility, online correlation clustering has been explored in either *stochastic* settings [2, 47] or scenarios that allow re-assignments at a cost [24], where algorithms with a constant competitive ratio have been devised. Yet, online non-centroid clustering in *metric* spaces has been largely overlooked. Further, prior works on online clustering typically assume that assignments to clusters are *immediate* upon arrival. To the best of our knowledge, online non-centroid clustering with *delays* has not been studied before.

As noted earlier, the *Min-cost Perfect Matching with Delays* (MPMD) problem presented by Emek et al. [30] is a special case of our setting. In the worst-case model with a finite metric space of size m that is *known* upfront, the best known competitiveness is $O(\log m)$ [6], and any algorithm's competitive ratio is at least $\Omega(\log m / \log\log m)$ [5]. If the metric space is *unknown*, the current best competitive ratio is $O(n^{\log(1.5)+\varepsilon}/\varepsilon)$ for n online arrivals and $\varepsilon > 0$. While MPMD has been studied under a Poisson arrival model with *known* arrival rates [51], we consider the more challenging and general *unknown i.i.d. arrival model*, where neither the distribution, its form nor its parameters are known. Our problem is more closely related to k-way MPMD, where elements must be matched into sets of size *exactly* k. In the worst-case model, any *randomized* algorithm has a generally *unbounded* competitive ratio [44, 45]. Yet, our setting is more challenging, allowing clusters of different sizes. We also study the *unknown i.i.d. model*, giving hope for beyond worst-case adversaries by devising an algorithm with ***constant*** competitiveness.

Our work captures cases where agents prefer to be close to others in their cluster, exemplifying *coalition formation*, where *agents* collaborate in *coalitions* instead of acting alone. A popular model for studying coalition formation is *hedonic games* [28], where agents express preferences for coalitions they belong to by disregarding externalities. Hedonic games have been extensively studied in *offline* settings (see, e.g., [7, 17, 56]), where settings with *bounded* coalition sizes [33, 49, 52] and *fixed-size* coalitions [9, 25] have recently received increased attention. Yet, the above works consider hedonic games in *offline* settings, while we study *online* non-centroid clustering problems with clusters of a fixed and predetermined size. Recently, an *online* variant of hedonic games has been explored [10, 18–23, 34]. However, those studies assume that decisions are *immediate*, while we allow postponing them at a cost. **In Section 5.1, we explain how our results also extend to hedonic games.**

3 PRELIMINARIES

We consider the *Online non-centroid Clustering with Delays* (OCD) problem, where n points, representing elements like agents or data, arrive one at a time over multiple rounds in a finite metric space. Formally, the input to OCD is a sequence σ of n points that arrive online over multiple rounds in a finite *metric space* $\mathcal{M} = (\mathcal{X}, d)$ with a set $\mathcal{X}$ of $|\mathcal{X}| < \infty$ different locations equipped with a distance function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ that satisfies the triangle inequality. Each point $i \in \sigma$ is given by its *location* $\ell_i \in \mathcal{X}$ and *arrival time* $t_i \in \mathbb{N}^+$. Note that points indexed consecutively in the ordering σ do not necessarily arrive in consecutive rounds, i.e., there may exist distinct points $i, i+1 \in \sigma$ such that $t_{i+1} - t_i \geq 2$, and all points arrive after exactly $T := \max_{i \in \sigma} t_i$ rounds. Without loss of generality, no two points arrive at the same time. For $k \in \mathbb{N}$, we hereafter denote $[k] := \{1, \dots, k\}$. To generate the sequence σ of n points, we consider the *unknown i.i.d.* (UIID) model, where arrivals are sampled independently across time from a *unknown*, *fixed* probability distribution $\{p_x\}_{x \in \mathcal{X}}$ satisfying $\sum_{x \in \mathcal{X}} p_x \leq 1$. Specifically, each point i arrives at location $x \in \mathcal{X}$ with probability $\mathbb{P}[\ell_i = x] = p_x$. However, no point arrives between any two consecutive arrivals of points with probability $1 - \sum_{x \in \mathcal{X}} p_x$, which is positive if $\sum_{x \in \mathcal{X}} p_x < 1$. We denote by N^t the set of points that arrive until time $t \in [T]$, and we set $N := N^T = N^t$ for any time $t \geq T$.

At each time $t \in [T]$, an online algorithm $\mathcal{A}$ shall produce a *partial clustering* C^t of the points *which arrived until time t* into disjoint subsets of points (i.e., *clusters*), without any knowledge about future points. Once point i arrives at time t_i, its location ℓ_i is revealed. Then, $\mathcal{A}$ has to *irrevocably* assign point i to an existing cluster in C^{t-1} or create a new cluster $\{t\}$, if possible. Unlike prior research on online clustering and online coalition formation [10, 18–24, 34], in our work an online algorithm $\mathcal{A}$ does *not* have to *immediately* assign an arriving point i to a cluster, but it may postpone the decision to some time $s_i \geq t_i$ with a ***delay cost***, which equals point i's waiting time, denoted as $w_i := s_i - t_i$. However, if no assignment is made at time $t \in [T]$, then $C^t = C^{t-1}$. Moreover, if points remain unassigned once the last point arrives (i.e., at time t_n), then they will keep waiting until the algorithm inserts them into clusters. We thus denote the number of clusters in C^t as $|C^t|$ and the cluster in C^t containing point $i \in N^t$ as $C^t(i)$.

Once two distinct points $i \neq j$ are in the same cluster, a ***connection cost*** is incurred, given by the distance between them plus their delay costs, i.e., $d(\ell_i, \ell_j) + w_i + w_j$. Therefore, we evaluate the quality of the clustering C^t at time $t \in [T]$ by its ***total cost***, defined as the sum of the connection costs in each cluster, i.e., $\text{tc}(C^t, \mathbf{w}) = \sum_{C \in C^t} \sum_{i,j \in C: i \neq j} [d(\ell_i, \ell_j) + w_i + w_j]$, where $\mathbf{w} = (w_i)_{i \in N}$ is the *delay profile*. For any cluster $C \in C^t$ and point $i \in C$, point i's delay cost affects her connection cost to any other agent $j \in C$. This captures scenarios where a point's delay imposes a cost on *each* cluster member, not just the delayed point. For example, in online gaming, all players must be present before the game starts. Hence, the delay of one player slows down *all* players' start time, so each cluster member suffers from the waiting time of late arrivals. It is thus reasonable to view that waiting times impose cost proportional to the number of participants affected.

We also remark that summing connection (distance) and delay (time) costs is a standard objective in the literature on online problems with delays (see, e.g., [30] and subsequent works). Here, the interpretation is that both terms represent the same unit of cost. If one requires a different conversion via a user-supplied conversion function $f : \mathbb{N} \to \mathbb{R}^+$, a point i with waiting time w_i suffers a delay cost of $f(w_i)$. Our analysis extends to some choices of f, but analyzing the general case is an interesting direction for future work. For example, $f(t) = \lambda t$ with $\lambda > 0$ scales time units; here, our analysis carries through with constants depending on λ (proofs are almost identical up to this scaling). **The main goal of $\mathcal{A}$ is thus finding a *cost-optimal* clustering $C^\star$ that *minimizes* the total cost among all possible clusterings of the n points.**

When clusters can be of any size, the clustering of singletons $(\{i\})_{i \in \sigma}$ is trivially cost-optimal, as it incurs a total cost of zero. We therefore focus on realistic cases with capacitated clusters. Namely, given k positive integers $\{n_m\}_{m \in [k]}$ with $n_1 \geq n_2 \geq \cdots \geq n_k$ and $\sum_{m \in [k]} n_m = n$, we seek a final clustering $C = (C_m)_{m \in [k]}$ where each point is assigned to one of k clusters such that the size of the m-th cluster is exactly $|C_m| = n_m$. We require that $2 \leq k < n$ and $2 \leq n_m < n$ for any $m \in [k]$, as cases where either $k = 1$, $n_m = 1$, $k = n$ or $n_m = n$ are trivial. Moreover, we treat the cluster sizes $\{n_m\}_{m \in [k]}$ as *constants*, while allowing the number of clusters k to grow with the number of points n, i.e., k and n are not constants. This assumption is natural in several applications where cluster sizes are intrinsic to the system. In online gaming, tournament systems, and team-based e-sports platforms, team sizes (e.g., 5-player squads) are fixed and cannot be chosen by the algorithm. Similarly, in ride-sharing and package-delivery services, vehicles have predetermined capacities. Further, in balanced document clustering, practical constraints, such as workloads of downstream human reviewers, also require explicit cardinality bounds. As noted in Section 5.2, our analysis also extends to requiring lower and upper bounds on coalition sizes, further generalizing our model beyond the fixed-size assumption.

We measure the performance of an online algorithm $\mathcal{A}$ in terms of its *ratio-of-expectations* [36], comparing the expected cost of $\mathcal{A}$ with the expected cost of the offline cost-optimal clustering. Formally, we denote by $\mathcal{I}$ the distribution over sequences of n points generated by the UIID model. For each sampled sequence $\sigma \sim \mathcal{I}$, let $\text{OPT}(\sigma)$ and $\mathcal{A}(\sigma)$ be the total cost of the *offline* cost-optimal clustering and the clustering generated by running the algorithm $\mathcal{A}$ on σ, respectively. We also denote $\text{Opt}(\mathcal{I}) := \mathbb{E}[\text{OPT}(\sigma)]$ and $\mathcal{A}(\mathcal{I}) := \mathbb{E}[\mathcal{A}(\sigma)]$, where $\mathbb{E}[\cdot]$ is the expectation over $\sigma \sim \mathcal{I}$ and the possible randomness of the algorithm. We thus say that $\mathcal{A}$ is c-competitive for $c \geq 1$ under the *ratio-of-expectations* (RoE) if:

$$\text{RoE}(\mathcal{A}) := \overline{\lim}_{n \to \infty} \frac{\mathcal{A}(\mathcal{I})}{\text{Opt}(\mathcal{I})} \leq c \tag{1}$$

where greater performance is indicated by a smaller RoE. The RoE measures $\mathcal{A}$'s asymptotic behavior and robustness as the number of points grows, thus evaluated at the limit $n \to \infty$. This highlights $\mathcal{A}$'s efficiency in practical, large-scale scenarios instead of small pathological cases that may not truly reflect $\mathcal{A}$'s effectiveness.

4 A *CONSTANT* COMPETITIVE ALGORITHM

We now present the deterministic and greedy Algorithm 1, termed as *Delayed Greedy* (DGreedy), which has a ***constant*** ratio-of-expectations (RoE) and runs as follows. At each time t, if no point arrives at time t, we proceed to treating pending points (line 3-4). Specifically, for any pending point i, we identify two sets of candidate clusters to which point i can be assigned. First, we construct

Algorithm 1 DGreedy

Input: A sequence σ of n points; Cluster sizes $\{n_m\}_{m \in [k]}$.

1: Initialize an empty clustering $C \leftarrow (C_m = \emptyset)_{m \in [k]}$.
2: **for** $t = 1, 2, \ldots, T$ **do**
3:   **if** no point arrives at time t **then**
4:     Proceed to treating pending points.
5:   **for** any pending point i **do**
6:     Set $\mathcal{S}_i \leftarrow \{m \in [k] : |C_m| < n_m$ and
7:     $\forall j \in C_m$ s.t. $d(\ell_i, \ell_j) \le (t - t_i) + w_j\}$.
8:     **if** there are at least two pending points **then**
9:       Set $\mathcal{D}_i \leftarrow \{(j, m) : i \neq j \in N^t$ is pending,
10:       $d(\ell_i, \ell_j) \le (t - t_i) + (t - t_j)$ and $C_m = \emptyset\}$.
11:     **if** $|\mathcal{S}_i| > 0$ **then** ▷ **Best existing cluster**
12:       Breaking ties arbitrarily, pick
13:       $m_1 \in \arg\min_{m \in \mathcal{S}_i} \mathrm{tc}(C_{+\{i\} \to m})$, where
14:       $C_{+C' \to m}$ is the clustering obtained from C
15:       by adding points in some $C' \subseteq N$ to C_m.
16:     **else** $m_1 = 0$
17:     **if** $|\mathcal{D}_i| > 0$ **then** ▷ **Best pending pair**
18:       $\mathcal{D}_i^\star = \arg\min_{(j,m) \in \mathcal{D}_i} \mathrm{tc}(C_{+\{i,j\} \to m})$.
19:       **Pick a pair with max. total waiting time:**
20:       Select $(j, m_2) \in \arg\max_{(j,m) \in \mathcal{D}_i^\star} (t - t_j)$
21:       (ties broken arbitrarily).
22:     **else** $m_2 = 0$
23:     **if** $m_2 \neq 0$ **and** ($m_1 = 0$ **or** ($m_1 \neq 0$ **and**
24:       $\mathrm{tc}(C_{+\{i,j\} \to m_2}) \le \mathrm{tc}(C_{+\{i\} \to m_1})$)) **then**
25:       Add i, j to the currently empty cluster C_{m_2}.
26:     **else if** $m_1 \neq 0$ **and** $m_2 = 0$ **then**
27:       Add point i to the existing cluster C_{m_1}.

$\mathcal{S}_i$, composed of any *existing* cluster $C_m \in C^t$ for some $m \in [k]$ such that $|C_m| \le n_m$ and point i's distance from any point $j \in C_m$ is at most point i's waiting time plus point j's delay cost (lines 6-7). Afterwards, if there are at least two pending points, then we build $\mathcal{D}_i$, consisting of any pair $(j, m) \in (N^t \setminus \{i\}) \times [k]$ such that $C_m = \emptyset$ and j is a pending point for which the total waiting time of points i, j exceeds the distance between them (lines 8-10).

If $|\mathcal{S}_i| > 0$, we pick $m_1 \in \mathcal{S}_i$ such that inserting point i to the cluster C_{m_1} *minimizes* the increase in the total cost of the current clustering (lines 11-15); otherwise, we set $m_1 = 0$ (line 16). Similarly, if $|\mathcal{D}_i| > 0$, we select a pair $(j, m_2) \in \mathcal{D}_i$ such that forming a new cluster $\{i, j\}$ by putting points i, j in the currently empty m_2-th cluster C_{m_2} *minimizes* the increase in the total cost of the current clustering (line 17). Among all such options, we pick the one that *maximizes* the total waiting time of its points (lines 19-21); otherwise, we set $m_2 = 0$ (line 22). Intuitively, by breaking ties based on the total waiting time, we balance immediate connection costs with long-term delay costs, ensuring that highly delayed points are treated sooner, preventing their costs from growing even further.

If we indeed picked a pair $(j, m_2) \in \mathcal{D}_i$ as above (i.e., $m_2 \neq 0$), then we insert points i, j into the currently empty cluster C_{m_2} if either no existing cluster with minimum increase in total cost was also selected (i.e., $m_1 = 0$), or an existing cluster C_{m_1} was found and the increase in total cost incurred by forming the new cluster $\{i, j\}$ is at most that of adding point i to C_{m_1} (lines 24-25). Otherwise, if $m_1 \neq 0$ and $m_2 = 0$, we add point i to the existing cluster C_{m_1} (lines 26-27). In any other case, point i will continue to wait and we proceed to the next pending point.

Note that the algorithm is well-defined. Throughout its execution, DGreedy enforces that the m-th cluster C_m has size at most n_m for each $m \in [k]$, either by adding points to C_m only when it contains fewer than n_m members, or by forming new clusters only of size 2 by assigning two pending points to an empty cluster. As $\sum_{m \in [k]} n_m = n$, every point is eventually assigned to a cluster since the metric space $\mathcal{M}$ contains a *finite* number of points, yielding that the *last* point's waiting time is bounded by the diameter of $\mathcal{M}$. We formulate this in the following observation:

Observation 1. *For any sequence of points σ, the final clusters generated by DGreedy contain exactly k clusters, where the m-th cluster is of size exactly n_m for any $m \in [k]$.*

Geometric Interpretation. To better grasp our algorithm, we consider its geometric interpretation (See Figure 1 for an illustration). When point i arrives, a ball centered at its location ℓ_i expands uniformly over time and stops growing when point i is assigned to a cluster. The ball's radius represents the delay cost for leaving point i unassigned. Thus, once the ball of a pending point i intersects that of either some point within an existing cluster or another pending point, point i is assigned to such an intersecting cluster.

Next, we first provide an upper bound on the (expected) total cost of the clustering generated by DGreedy (Section 4.1). After devising a lower bound on the (expected) total cost of the optimal (offline) clustering (Section 4.2), we infer DGreedy's ***constant*** ratio-of-expectations (Section 4.3).

4.1 Upper Bounding DGreedy's total cost

In this section, we obtain an upper bound on the expected total cost of the clustering produced by DGreedy, and begin with a general bound on the incurred total cost:

Lemma 1. *Given a* finite *metric space $\mathcal{M} = (\mathcal{X}, d)$, for any sequence of n points σ, the total cost of the final clustering C generated by DGreedy satisfies:*

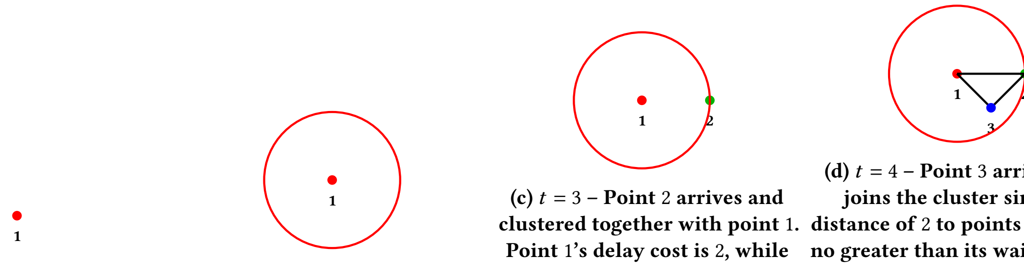
$$\mathrm{tc}(C, \mathbf{w}) \le 2(n_1 - 1) \sum_{i \in \sigma} w_i$$

Proof. For any pair of points i and $j \in C(i)$ with $i \neq j$, as their connection cost is at most the sum of their delay costs, i.e., $d(\ell_i, \ell_j) \le w_i + w_j$, we obtain:

$$\begin{aligned} \mathrm{tc}(C, \mathbf{w}) &= \textstyle\sum_{C \in C} \sum_{i,j \in C: i \neq j} [d(\ell_i, \ell_j) + w_i + w_j] \\ &\le \textstyle\sum_{C \in C} \sum_{i,j \in C: i \neq j} 2[w_i + w_j] \\ &= 2 \textstyle\sum_{C \in C} (|C| - 1) \sum_{i \in C} w_i \le 2(n_1 - 1) \sum_{i \in \sigma} w_i \end{aligned}$$

where the last inequality is by $|C| \le n_1$ for any cluster $C \in C$ (Observation 1) and the m-th cluster's size satisfies $n_m \le n_1$. □

Hence, by finding an upper bound on the expected total delay cost, we can estimate the overall expected total cost of the clustering produced by DGreedy. To this end, we next bound the delay cost of each point. For any subset of locations $\mathcal{Y} \subseteq \mathcal{X}$, if a point arrived at some location in $\mathcal{Y}$ during time t, then we next make the following observation regarding the ***waiting time*** $\tau_t^{\mathcal{Y}}$ for another point to arrive at some location in $\mathcal{Y}$:



(a) $t = 1$ **– Point** 1 **arrives.**

(b) $t = 2$ **– No point arrives.**

(c) $t = 3$ **– Point** 2 **arrives and clustered together with point** 1**. Point** 1**'s delay cost is** 2**, while point** 2**'s delay cost is** 0**.**

(d) $t = 4$ **– Point** 3 **arrives and joins the cluster since its distance of** 2 **to points** 1 **and** 2 **is no greater than its waiting time (**0**) plus their delay cost (**2**).**

Figure 1: An example of how DGreedy works for clusterings with a single cluster of size 3 **on a sequence of three points** 1, 2, 3 **arriving at times** 1, 3, 4 **in a finite metric space consisting of** 3 **locations, where the distance between each pair of locations is** 2**.**

Observation 2. *In the UIID model with a* finite *metric space* $\mathcal{M} = (\mathcal{X}, d)$*, the waiting time* $\tau_t^{\mathcal{Y}}$ *is geometrically distributed with a success probability of* $q_{\mathcal{Y}} := \sum_{y \in \mathcal{Y}} p_y$*, and thus its expectation in the UIID model is* $\mathbb{E}[\tau_t^{\mathcal{Y}}] = \frac{1}{q_{\mathcal{Y}}}$.

To bound each point's delay cost, we distinguish between two types of points that may arrive after it. First, for any possible location $x \in \mathcal{X}$, let $\bar{B}(x,r)$ (resp. $B(x,r)$) be the *closed* (resp. *open*) ball centered at x with radius $r > 0$, i.e., the set of locations y with $d(x,y) \le r$ (resp. $d(x,y) < r$). A point arriving at location x is said to be assigned to a cluster together with a *close point* if the distance between them is at most location x's radius r_x, which is defined as:

$$r_x := \min\left\{r \ge 0 : \frac{1}{\sum_{y \in \bar{B}(x,r)} p_y} \le r\right\} \tag{2}$$

Intuitively, the choice of r_x balances the expected waiting time $\mathbb{E}[\tau_t^{\bar{B}(x,r)}]$ between two arrivals within the ball $\bar{B}(x,r)$ (by Observation 2) and this ball's diameter. It is also well-defined since the function $r \mapsto \mathbb{E}[\tau_t^{\bar{B}(x,r)}]$ is non-increasing, and thus $r_x \in (0, 1/p_x]$.

For any point i, we say that it is an ***early point*** if i is not pending at time $t_i + r_{\ell_i}$ and there is another point j with $t_j - t_i > r_{\ell_i}$ such that $d(\ell_i, \ell_j) \le r_{\ell_i}$; otherwise, we say that point i is a ***late point***. For each *early* point i, we set:

$$\alpha_i^{\text{early}} := \begin{cases} 0, \text{ if } i \text{ is assigned to a cluster at time } t_i + r_{\ell_i} \\ \min_{j \in N}\{t_j - t_i - r_{\ell_i} : t_j - t_i > r_{\ell_i} \ \&\ d(\ell_i, \ell_j) \le r_{\ell_i}\}, \text{ o.w.} \end{cases}$$

Now, we upper bound the waiting time of the ***early*** point i:

Lemma 2. *In the UIID model with a* finite *metric space* $\mathcal{M} = (\mathcal{X}, d)$*, for any sequence of n points* σ *and each* early *point* $i \in \sigma$*, it holds that* $w_i \le r_{\ell_i} + \alpha_i^{early}$.

Proof. Let i be an early point. Let j be the *first* point with $t_j - t_i > r_{\ell_i}$ such that $d(\ell_i, \ell_j) \le r_{\ell_i}$, i.e., $t_j - t_i = r_{\ell_i} + \alpha_i^{\text{early}}$. If point i has already been assigned to a cluster at time t_j, then $s_i \le t_j$ and $w_i = s_i - t_i \le t_j - t_i = r_{\ell_i} + \alpha_i^{\text{early}}$. Otherwise, we show that a cluster containing points i, j is created at time t_j. Note that $(t_j - t_j) + (t_j - t_i) = r_{\ell_i} + \alpha_i^{\text{early}} > r_{\ell_i} \ge d(\ell_i, \ell_j)$. As points i and j are pending points at time t_j and the greedy criteria is met, then clusters of pending points that contain i, j are considered eligible by DGreedy. Assume, towards contradiction, that there is another pending point j' such that a cluster containing points j and j' has already been formed at time t_j, i.e., $d(\ell_j, \ell_{j'}) \le t_j - t_{j'}$. By the triangle inequality, we have: $(t_j - t_i) + (t_j - t_{j'}) > d(\ell_i, \ell_j) + d(\ell_j, \ell_{j'}) \ge d(\ell_i, \ell_{j'})$. That is, a cluster containing the points i, j' should have been created before the arrival of point j, which is a contradiction. We infer that $w_i = s_i - t_i = r_{\ell_i} + \alpha_i^{\text{early}}$. □

In the following, we bound the total delay cost of ***late*** points. Sadly, the waiting time of a late point may be the highest value possible, i.e., the diameter $d_{\max} := \max_{x,y \in \mathcal{X}} d(x,y)$ of the metric space $\mathcal{M}$. However, we prove that only a few such points exist. Recalling that t_n is the arrival time of the last point, then we define:

$$\alpha_i^{\text{late}} := \begin{cases} 0, \text{ if } t_n \le t_i + d_{\max} \\ \min_{j \in N}\{t_j - (t_i + d_{\max}) : t_j > t_i + d_{\max}\}, \text{ o.w.} \end{cases}$$

Therefore, we obtain that:

Lemma 3. *In the UIID model with a* finite *metric space* $\mathcal{M} = (\mathcal{X}, d)$*, for any sequence* σ *and each location* $x \in \mathcal{X}$*:*

1. *There is at most one* late *point at* x*, i.e., there are at most* $|\mathcal{X}| < \infty$ *late points.*
2. *For each* late *point i, it holds that* $w_i \le d_{\max} + \alpha_i^{late}$.

Proof. We prove each part separately:

1. Consider two points i, j with $\ell_i = \ell_j$ and $t_i < t_j$. Assume, towards contradiction, that points i and j are late. By the definition of a late point, we have that $t_j - t_i \le r_{\ell_i}$ and point i remains unassigned to a cluster at time t_j. As point j is also late, it is not assigned to a cluster upon its arrival and DGreedy does not form the cluster $\{i, j\}$. However, $0 = d(\ell_i, \ell_j) \le r_{\ell_i} + r_{\ell_j}$, and thus DGreedy should create the cluster $\{i, j\}$ at time t_j, which is a contradiction. As such, point i cannot remain a pending point so long as at least one point arrives at its location, yielding that it is not a late point. That is, at most one late point is located at each location $x \in \mathcal{X}$, i.e., there are at most $|\mathcal{X}| < \infty$ late points (as the metric space is *finite*).
2. Given a late point i, we distinguish between the following:
   (a) If $t_n \le t_i + d_{\max}$, then all the points have already arrived by time $t = t_i + d_{\max}$. Thus, either point i has already been assigned to a cluster or not. In the latter case, the following cases are possible:
      (i) If all clusters are non-empty when DGreedy processes point i, then there is at least one cluster of index $m \in [k]$ such that $|C_{m'}| < n_{m'}$, as the m-th cluster should be

eventually of size n_m and $\sum_{m\in[k]} n_m = n$. Since $t - t_i = d_{\max} \geq d(\ell_i, \ell_j)$ for any point j in the m-th cluster C_m for every $m \in [k]$, our algorithm assigns point i to such a cluster at time t, yielding $w_i = d_{\max}$.

(ii) Otherwise, when our algorithm processes point i, there exists at least one empty cluster of index $m \in [k]$, i.e., $C_m = \emptyset$. Since the m-th cluster should eventually contain exactly $n_m \geq 2$ points, there is at least one other pending point $j \neq i$ at time t. Because $t - t_i = d_{\max} \geq d(\ell_i, \ell_j)$ for any other pending point $j \neq i$, a cluster of pending points containing i, j is considered eligible by DGreedy at time t when point i is being processed by DGreedy. At this time, if all non-empty clusters are full, then our algorithm assigns point i together with other pending points to a currently empty cluster, meaning that $w_i = d_{\max}$. Otherwise, when our algorithm processes point i, there exists at least one partially filled cluster of index $m' \in [k]$ with $1 \leq |C_{m'}| < n_{m'}$. For any such $m' \in [k]$, since $t - t_i = d_{\max} \geq d(\ell_i, \ell_j)$ for any point $i \neq j \in C_{m'}$, then the algorithm identifies a cluster C_{m_1} with $1 \leq |C_{m_1}| < n_{m_1}$ such that inserting point i to the cluster C_{m_1} minimizes the increase in the total cost of the current clustering. Thus, at time t, DGreedy will assign point i to either the m_1-th cluster or a currently empty cluster, depending on which yields a lower total cost. In both cases, $w_i = d_{\max}$.

(b) Otherwise, if $t_n > t_i + d_{\max}$, then the point j that arrives right after point i appears at time $t = t_i + d_{\max} + \alpha_i^{\text{late}}$ by the definition of α_i^{late}. If point i has already been assigned to a cluster at time t, then its waiting time is at most $w_i \leq d_{\max} + \alpha_i^{\text{late}}$. Otherwise, note that $t - t_i = d_{\max} + \alpha_i^{\text{late}} > d_{\max} \geq d(\ell_i, \ell_j)$. As points i and j are pending points at time t and the greedy criteria is met, then a cluster of pending points containing i, j is considered eligible by DGreedy. By arguments similar to the proof of Lemma 2, such a cluster is created. In any case, we conclude that $w_i \leq d_{\max} + \alpha_i^{\text{late}}$, as desired. □

We are now ready to derive an upper bound on the expected total cost of the clustering generated by DGreedy. It will later enable us to obtain DGreedy's ***constant*** ratio-of-expectations in Section 4.3.

Theorem 1. *In the UIID model with a* finite *metric space* $\mathcal{M} = (\mathcal{X}, d)$*, DGreedy's expected total cost over all sequences of n points sampled from a distribution* $\mathcal{I}$ *satisfies:*

$$DGreedy(\mathcal{I}) \leq 2(n_1 - 1)\left[n \sum_{x\in\mathcal{X}} p_x r_x + |\mathcal{X}| \cdot d_{\max}\right] + \frac{2(n_1-1)|\mathcal{X}|}{\sum_{x\in\mathcal{X}} p_x}$$

Proof. First, we bound the expected delay cost of each *late* point i. If $t_n > t_i + d_{\max}$, then the point j that arrives right after point i appears at time $t = t_i + d_{\max} + \alpha_i^{\text{late}}$ by the definition of α_i^{late}. Namely, the waiting time between the arrivals of points i and j is $\tau_{t_i}^{\mathcal{X}} = t - t_i = d_{\max} + \alpha_i^{\text{late}}$. Since $\sum_{x\in\mathcal{X}} p_x < 1$, then by Observation 2 and since $\alpha_i^{\text{late}} = 0$ if $t_n \leq t_i + d_{\max}$, we infer that:

$$\begin{aligned} &\mathbb{E}[\alpha_i^{\text{late}} | i \text{ is a late point}] \\ &\leq \mathbb{E}[\alpha_i^{\text{late}} | i \text{ is a late point and } t_n > t_i + d_{\max}] \\ &= -d_{\max} + \frac{1}{\sum_{x\in\mathcal{X}} p_x} \end{aligned} \tag{3}$$

Since the waiting time of a late point is at most $d_{\max}$:

$$\begin{aligned} &\mathbb{E}[w_i | i \text{ is a late point}] \\ &\leq \mathbb{E}[w_i | i \text{ is a late point and } t_n > t_i + d_{\max}] \\ &+\mathbb{E}[w_i | i \text{ is a late point and } t_n \leq t_i + d_{\max}] \\ &\leq \mathbb{E}[w_i | i \text{ is a late point and } t_n > t_i + d_{\max}] + d_{\max} \end{aligned} \tag{4}$$

Combining (3) and (4) with Lemma 3, $\mathbb{E}[w_i | i \text{ is a late point}] \leq d_{\max} + 1/\sum_{x\in\mathcal{X}} p_x$ due to $\sum_{x\in\mathcal{X}} p_x < 1$. Since we consider finite metric spaces and there are at most $|\mathcal{X}| < \infty$ late points by Lemma 3, the expected total delay cost of the late points satisfies:

$$\mathbb{E}\left[\sum_{i\in N: i \text{ is late}} w_i\right] \leq |\mathcal{X}| \left[d_{\max} + \frac{1}{\sum_{x\in\mathcal{X}} p_x}\right] \tag{5}$$

Next, we analyze the expected delay cost of an *early* point i. Let j be the *first* point with $t_j - t_i > r_{\ell_i}$ such that $d(\ell_i, \ell_j) \leq r_{\ell_i}$. Namely, the waiting time between the arrivals of points i and j within $\bar{B}(\ell_i, r_{\ell_i})$ is $\tau_{t_i}^{\bar{B}(\ell_i, r_{\ell_i})} = t_j - t_i = r_{\ell_i} + \alpha_i^{\text{early}}$. By Observation 2 and (2), since $\sum_{x\in\bar{B}(\ell_i, r_{\ell_i})} p_x < \sum_{x\in\mathcal{X}} p_x < 1$, then we have:

$$\mathbb{E}\left[\alpha_i^{\text{early}} \middle| i \text{ is an early point}\right] = -r_{\ell_i} + \frac{1}{\sum_{x\in\bar{B}(\ell_i, r_{\ell_i})} p_x} \leq -r_{\ell_i} + r_{\ell_i} = 0$$

Therefore, we obtain that $\mathbb{E}[w_i | i \text{ is an early point}] \leq r_{\ell_i}$ by Lemma 2. Thus, the expected total delay cost of early points satisfies:

$$\begin{aligned} &\mathbb{E}\left[\sum_{i\in N: i \text{ is early}} w_i\right] \\ &\leq \sum_{i\in N} \sum_{x\in\mathcal{X}} \mathbb{P}[i \text{ is early and } \ell_i = x] \cdot \mathbb{E}[w_i | i \text{ is early and } \ell_i = x] \\ &\leq \sum_{i\in N} \sum_{x\in\mathcal{X}} p_x r_x = n \sum_{x\in\mathcal{X}} p_x r_x \end{aligned} \tag{6}$$

where we used $r_x \leq 1/p_x$ for $p_x > 0$. We conclude the desired by combining (5) with $\text{DGreedy}(\mathcal{I}) \leq \mathbb{E}\left[2(k-1)\sum_{i\in N} w_i\right] = 2(n_1 - 1)\left[\mathbb{E}\left[\sum_{i\in N: i \text{ is early}} w_i\right] + \mathbb{E}\left[\sum_{i\in N: i \text{ is late}} w_i\right]\right]$, where the latter holds by Lemma 1. □

4.2 Lower Bounding the Optimal total cost

In this section, we obtain a lower bound on the total cost of the (offline) cost-optimal clustering. We begin with a general lower bound on the minimal total cost. For any sequence of points σ, we denote the *minimal delay cost* of each point $i \in \sigma$ by $w_i^\sigma = s_i^\sigma - t_i$ and its *minimum cost* as:

$$c_i(\sigma) = \min_{i\neq j\in\sigma}\{d(\ell_i, \ell_j) + w_i^\sigma + w_j^\sigma\}$$

Letting $\mathbf{w}^\sigma = (w_i^\sigma)_{i\in\sigma}$ be the *optimal delay profile*, we obtain the following lower bound on the minimal total cost:

Lemma 4. *In the UIID model with a* finite *metric space* $\mathcal{M} = (\mathcal{X}, d)$*, for any sequence of n points* σ*, the minimum total cost satisfies:*

$$\text{OPT}(\sigma) \geq \frac{n_k - 1}{2} \sum_{i\in\sigma} c_i(\sigma)$$

Proof. Fix a cost-optimal clustering $C^\star$ with an optimal delay profile $\mathbf{w}^\sigma = (w_i^\sigma)_{i\in\sigma}$. For any pair of distinct points $i, j \in \sigma$, $c_i(\sigma)$ and $c_j(\sigma)$ are both at most $d(\ell_i, \ell_j) + w_i^\sigma + w_j^\sigma$, and thus

$d(\ell_i, \ell_j) + w_i^\sigma + w_j^\sigma \geq \frac{c_i(\sigma)+c_j(\sigma)}{2}$. As $|C| \geq n_k$ for any $C \in C^\star$ since the m-th cluster's size satisfies $n_m \geq n_k$ for any $m \in [k]$, we have:

$$\begin{aligned} \text{tc}(C^\star, \mathbf{w}^\sigma) &= \sum_{C \in C^\star} \sum_{i,j \in C: i \neq j} [d(\ell_i, \ell_j) + w_i^\sigma + w_j^\sigma] \\ &\geq \sum_{C \in C^\star} \sum_{i,j \in C: i \neq j} \frac{c_i(\sigma)+c_j(\sigma)}{2} \\ &= \frac{1}{2} \sum_{C \in C^\star} \sum_{i \in C} (|C| - 1) c_i(\sigma) \geq \frac{n_k - 1}{2} \sum_{i \in \sigma} c_i(\sigma) \end{aligned}$$

□

Thus, we can now estimate the expected minimum total cost by deriving a lower bound on the minimum cost of each point:

Lemma 5. *In the UIID model with a* finite *metric space* $\mathcal{M} = (\mathcal{X}, d)$*, for any sequence of n locations* σ *and any location* $x \in \mathcal{X}$*, the expected minimum cost of each point* $i \in \sigma$*, given that it is located at* x*, satisfies:*

$$\mathbb{E}[c_i(\sigma) | \ell_i = x] \geq \frac{1 - e^{-2}}{4 q_x}$$

Proof. First, we denote by $\mathbb{E}_\sigma[\cdot]$ the expectation over the randomness of any sequence of points σ. By renaming the points, consider a random sequence of points $\sigma = (1, \ldots, n)$ such that the points are ordered according to their arrival times. We then extend σ by introducing a dummy random point j for each $j \leq 0$ and $j \geq n+1$ such that point j is also sampled according to $\{p_x\}_{x \in \mathcal{X}}$, and we thus obtain an extended random sequence $\bar{\sigma} = (\ldots, -1, 0, 1, \ldots, n, n+1, n+2, \ldots)$. Hence, in the extended random sequence $\bar{\sigma}$, with probability one there are points $j \leq 0$ and $j' \geq n+1$ such that $\ell_j = \ell_{j'} = x$ for any location $x \in \mathcal{X}$.

Note that the distribution of the truncation $\bar{\sigma}_n := (1, \ldots, n)$ is the same as that of σ, but the minimum cost of each point $i \in \sigma^p$ can only *decrease*, i.e., $c_i(\bar{\sigma}) \leq c_i(\sigma)$, yielding $\mathbb{E}_{\bar{\sigma}}[c_i(\bar{\sigma}) | \ell_i = x] \leq \mathbb{E}_\sigma[c_i(\sigma) | \ell_i = x]$ for each location $x \in \mathcal{X}$. Now, observe that the (conditional) expected minimum cost of each pair of distinct points $i, j \in \bar{\sigma}$ in an extended random sequence arriving at any location $x \in \mathcal{X}$ is the same, i.e., $\mathbb{E}_{\bar{\sigma}}[c_i(\bar{\sigma}) | \ell_i = x] = \mathbb{E}_{\bar{\sigma}}[c_j(\bar{\sigma}) | \ell_j = x]$.

Hence, we focus on point 0 and analyze $\mathbb{E}_{\bar{\sigma}}[c_0(\bar{\sigma}) | \ell_0 = x]$ for some $x \in \mathcal{X}$. Let $\bar{\sigma}$ be an extended sequence with $\ell_0 = x$. Without loss of generality, we assume $t_0 = 0$ by shifting the points' arrival times by a constant. For the extended random sequence $\bar{\sigma}$, we denote by $\mathcal{E}^0$ the event that point 0's cluster is first formed during the arrival of a point $i \geq 0$ by grouping pending points. We aim to give a lower bound on point 0's expected minimum cost conditioned on the event $\mathcal{E}^0$. For any other point $0 \neq j \in \bar{\sigma}$, consider the case where points 0 and j are eventually in the same cluster. Under the event $\mathcal{E}^0$, by Claim 1 in the full version [16], point 0's cluster $C_{\bar{\sigma}}^{s_0^{\bar{\sigma}}}(0)$ is initially formed by grouping pending points at time $s_0^{\bar{\sigma}} = t_{j'}$ for $j' = \arg\max_{\tilde{j} \in C_{\bar{\sigma}}^{s_0^{\bar{\sigma}}}(0)} \{t_{\tilde{j}}\}$. If point j is one of those pending points (i.e., $s_j^\sigma = t_{j'}$), then we prove that $w_0^{\bar{\sigma}} + w_j^{\bar{\sigma}} \geq |t_0 - t_j|$. Otherwise, if point j is assigned to point i's cluster after its creation (i.e., $s_j^\sigma > t_{j'}$), then $w_0^{\bar{\sigma}} + w_j^{\bar{\sigma}} = t_{j'} - t_0 + s_j^\sigma - t_j > t_{j'} - t_0 + t_{j'} - t_j$, which yields that $w_0^{\bar{\sigma}} + w_j^{\bar{\sigma}} \geq |t_0 - t_j|$ by arguments similar to the previous case.

In any case, we have obtained that $w_0^{\bar{\sigma}} + w_j^{\bar{\sigma}} \geq |t_0 - t_j| \forall 0 \neq j \in \bar{\sigma}$, by which we next lower bound $\mathbb{E}_{\bar{\sigma}}[c_0(\bar{\sigma}) | \ell_0 = x \wedge \mathcal{E}^0]$. Let t^- (resp. t^+) be the (finite) random waiting time between the arrival of point 0 and the arrival of the *last* point *before* point 0 (resp. the *first* point *after* point 0), arriving at some location $y \in B(x, r_x)$, i.e.:

$$t^- := \min_{j<0} \{-t_j : d(\ell_j, x) < \rho_x\} \text{ and } t^+ := \min_{j>0} \{t_j : d(\ell_j, x) < \rho_x\}$$

Under the event $\mathcal{E}^0$, note that, for any point $j \neq 0$, the quantity $d(\ell_i, \ell_j) + |t_j|$ can be lower bounded by r_x when $d(\ell_i, \ell_j) \geq r_x$ and by $|t_j|$ otherwise. Thus, in [16], we prove that $c_0(\bar{\sigma}) \geq \min\{\min\{t^-, t^+\}, r_x\}$ the following holds under the event $\mathcal{E}^0$.

To conclude the desired bound, note that the waiting times t^- and t^+ are mutually independent. Since t^- is the (finite) random waiting time between the arrival of point 0 and the arrival of the *last* point *before* point 0, then t^- is geometrically distributed with a success probability of $q_x := \sum_{y \in B(x, r_x)} p_y$ by arguments similar to Observation 1 in the main text. That is, $t^- \sim G(q_x)$. Similarly, $t^+ \sim G(q_x)$. We thereby require the following lemmas in order to derive the distribution of $\min\{t^-, t^+\}$ and the expectation of $\min\{\min\{t^-, t^+\}, r_x\}$. Combined with $c_0(\bar{\sigma}) \geq \min\{\min\{t^-, t^+\}, r_x\}$, this will aid us in deriving our desired lower bound on point 0's expected minimum cost conditioned on the event $\mathcal{E}^0$.

Lemma 6. *Let* $Y_1 \sim G(q_1)$ *and* $Y_2 \sim G(q_2)$ *be two independent geometric random variables. Let* $Z = \min\{Y_1, Y_2\}$*. Then,* Z *is a geometric random variable with a success probability of* $1 - (1 - q_1)(1 - q_2)$*, i.e.,* $Z \sim G(1 - (1 - q_1)(1 - q_2))$ *(see the full version [16] for a proof).*

Lemma 7. *For any* $s \in \mathbb{N}$ *and geometric random variable* $Y \sim G(q)$*:* $\mathbb{E}[\min\{Y, s\}] = \frac{1 - (1-q)^s}{q}$ *(see the full version [16] for a proof).*

As $t^- \sim G(q_x)$ and $t^+ \sim G(q_x)$, then $\min\{t^-, t^+\} \sim G(1 - (1 - q_x)^2)$ by Lemma 6. Together with Lemma 7, we prove in the full version [16] that $\mathbb{E}_{\bar{\sigma}}[c_0(\bar{\sigma}) | \ell_0 = x \wedge \mathcal{E}_0] \geq \frac{1 - (1 - q_x)^{2 r_x}}{q_x [2 - q_x]}$. As $r_x \geq \frac{1}{q_x}$ by r_x's definition in equation (2) within the main text, note that $(1 - q_x)^{2 r_x} \leq (1 - \frac{1}{r_x})^{2 r_x} \leq e^{-2}$. Combining the above with $q_x[2 - q_x] \leq 2 q_x$ due to $q_x \in [0, 1]$, we have that $\mathbb{E}_{\bar{\sigma}}[c_0(\bar{\sigma}) | \ell_0 = x \wedge \mathcal{E}_0] \geq \frac{1 - e^{-2}}{2 q_x}$. Now, let $\bar{\mathcal{E}}^0$ be the complement of the event $\mathcal{E}^0$, i.e., the event that point 0 is assigned to a cluster formed during the arrival of a point $i < 0$. By symmetry, note that $Pr[\mathcal{E}^0] = Pr[\bar{\mathcal{E}}^0] = \frac{1}{2}$ and $\mathbb{E}_{\bar{\sigma}}[c_0(\bar{\sigma}) | \ell_0 = x \wedge \bar{\mathcal{E}}^0] \geq 0$. Thus, using $\mathbb{E}_{\bar{\sigma}}[c_0(\bar{\sigma}) | \ell_0 = x \wedge \mathcal{E}_0] \geq \frac{1 - e^{-2}}{2 q_x}$, we obtain $\mathbb{E}_{\bar{\sigma}}[c_0(\bar{\sigma}) | \ell_0 = x] \geq \frac{1 - e^{-2}}{4 q_x}$, as desired. □

Finally, we establish a lower bound on the expected minimum total cost, which will soon allow us to establish DGreedy's ***constant*** ratio-of-expectations in Section 4.3.

Theorem 2. *In the UIID model with a* finite *metric space* $\mathcal{M} = (\mathcal{X}, d)$*, the expected minimum total cost over all sequences of n points sampled from a distribution* $\mathcal{I}$ *is at least:*

$$\text{Opt}(\mathcal{I}) \geq n(n_k - 1) \frac{1 - e^{-2}}{4} \sum_{x \in \mathcal{X}} \frac{p_x}{q_x}$$

where $q_x := 1 - \sum_{y \in B(x, r_x)} p_y > 1 - \sum_{x \in \mathcal{X}} p_x > 0$.

Proof. By Lemmas 4-5, $\text{Opt}(\mathcal{I}) = \mathbb{E}[\text{OPT}(\sigma)]$ satisfies:

$$\begin{aligned} \mathbb{E}[\text{OPT}(\sigma)] &\geq \frac{n_k - 1}{2} \sum_{i \in N} \mathbb{E}[c_i(\sigma)] \\ &= \frac{n_k - 1}{2} \sum_{i \in N} \sum_{x \in \mathcal{X}} \mathbb{P}[\ell_i = x] \mathbb{E}[c_i(\sigma) | \ell_i = x] \\ &\geq n(n_k - 1) \frac{1 - e^{-2}}{4} \sum_{x \in \mathcal{X}} \frac{p_x}{q_x} \end{aligned}$$

□

4.3 DGreedy's Constant Ratio-of-Expectations

We herein prove that the performance guarantee within the UIID model is significantly better compared with the current best competitiveness in the (worst-case) adversarial model. Particularly, we show that DGreedy achieves a ***constant*** ratio-of-expectations:

THEOREM 3. *In the UIID model with a* finite *metric space* $\mathcal{M} = (\mathcal{X}, d)$*, if the cluster sizes* $\{n_m\}_{m\in[k]}$ *are constants and the number of clusters* k *grows with the number of points* n *(*n, k *are not constants), DGreedy (Algorithm 1) has a* **constant** *ratio-of-expectations of:*

$$\lim_{n\to\infty} \frac{\text{DGreedy}(\mathcal{I})}{\text{Opt}(\mathcal{I})} \leq \frac{8(n_1-1)}{(n_k-1)(1-e^{-2})}$$

If all cluster sizes are equal (i.e., $n_m = n_{m'}$ *for any* $m, m' \in [k]$*), then:*

$$\lim_{n\to\infty} \frac{\text{DGreedy}(\mathcal{I})}{\text{Opt}(\mathcal{I})} \leq \frac{8}{1-e^{-2}}$$

PROOF. By Theorem 1:

$$\lim_{n\to\infty} \frac{\text{DGreedy}(\mathcal{I})}{\text{Opt}(\mathcal{I})} \leq \lim_{n\to\infty} \frac{2n(n_1-1)\sum_{x\in\mathcal{X}} p_x r_x}{\text{Opt}(\mathcal{I})} + \lim_{n\to\infty} \frac{2(n_1-1)|\mathcal{X}|\cdot\left[d_{\max} + \frac{1}{\sum_{x\in\mathcal{X}} p_x}\right]}{\text{Opt}(\mathcal{I})} \quad (7)$$

By Theorem 2, the expected minimal total cost $\text{Opt}(\mathcal{I})$ is lower bounded by a quantity linear in the number of points n. Hence, since the numerator of the last term in (7) does not depend on n, while n_1 is constant and we consider finite metric spaces (i.e., $|\mathcal{X}| < \infty$), the limit in the last term in (7) equals 0. By also applying Theorem 2 to the first term on the right-hand-side of (7):

$$\begin{aligned}\lim_{n\to\infty} \frac{\text{DGreedy}(\mathcal{I})}{\text{Opt}(\mathcal{I})} &\leq \lim_{n\to\infty} \frac{2n(n_1-1)\sum_{x\in\mathcal{X}} p_x r_x}{n(n_k-1)\frac{1-e^{-2}}{4}\sum_{x\in\mathcal{X}} \frac{p_x}{q_x}} \\ &= \frac{2(n_1-1)\sum_{x\in\mathcal{X}} p_x r_x}{(n_k-1)\frac{1-e^{-2}}{4}\sum_{x\in\mathcal{X}} \frac{p_x}{q_x}} \\ &\leq \frac{8(n_1-1)\sum_{x\in\mathcal{X}} \frac{p_x}{q_x}}{(n_k-1)(1-e^{-2})\sum_{x\in\mathcal{X}} \frac{p_x}{q_x}} = \frac{8(n_1-1)}{(n_k-1)(1-e^{-2})}\end{aligned}$$

where the last inequality follows from $\frac{1}{\sum_{y\in\bar{B}(x,r)} p_y} \leq r_x \leq \frac{1}{q_x} = \frac{1}{\sum_{y\in B(x,r)} p_y}$ due to r_x's definition in (2). If all cluster sizes are equal (i.e., $n_m = n_{m'}$ for any $m, m' \in [k]$), then the above yields $\lim_{n\to\infty} \frac{\text{DGreedy}(\mathcal{I})}{\text{Opt}(\mathcal{I})} \leq \frac{8}{1-e^{-2}}$. □

REMARK 1 (FINITE n). *As noted, e.g., after (1), the RoE measures an algorithm's asymptotic behavior as* $n \to \infty$ *captures its efficiency in practical large-scale scenarios instead of small pathological worst-case situations that may not truly reflect its effectiveness [50]. However, in practical online settings, systems operate with finite, often bounded, inputs. One can derive concrete finite-n bounds directly from our analysis: Theorems 1 and 2 already give finite-n bounds, while DGreedy's RoE for finite n easily follows from the proof of Theorem 3.*

5 EXTENSIONS

In this section, we show that our theoretical guarantees easily extend to non-metric spaces that encompass certain subclasses of hedonic games (Section 5.1), and scenarios with lower and upper bounds on cluster sizes rather than fixed-size clusters (Section 5.2).

5.1 Non-Metric Spaces and Hedonic Games

To the best of our knowledge, online non-centroid clustering with delays has not been studied before, let alone in (non-)metric spaces [10, 18–23, 34]. Yet, while our algorithm's analysis relies on the metric assumption, it employs several non-trivial components that also extend to non-metric domains. Indeed, consider a *non-metric* space $\mathcal{M} = (\mathcal{X}, d)$. First, assume that $d(x, y) > 0$ for any $x \neq y$.

(1) If d satisfies the triangle inequality but is asymmetric (i.e., $d(x, y) \neq d(y, x)$ for some x, y), then we can run our algorithm in the symmetrized metric space $\mathcal{M}^S = (\mathcal{X}, d^S)$ where $d^S(x, y) = \frac{d(x,y)+d(y,x)}{2}$ for any x, y, replacing each occurrence of d in the connection cost, total cost, the algorithm, and its analysis with d^S. As $\mathcal{M}^S$ is a metric space, all guarantees continue to hold.
(2) If d does not satisfy the triangle inequality but is still symmetric (i.e., $d(x, y) = d(y, x)$ for any x, y), then construct the weighted complete directed graph $G = (\mathcal{X}, \mathcal{X} \times \mathcal{X}, d)$ with weight $d(x, y)$ on arc (x, y) and let $d_G(x, y)$ be the length of the shortest path between x, y. As d is symmetric, d_G is a metric. Thus, running our algorithm in the metric space $\mathcal{M}_G = (\mathcal{X}, d_G)$ maintains our guarantees as before.
(3) If d is asymmetric and does not satisfy the triangle inequality, we consider the symmetrization from case (1) of $\mathcal{M}_G$.
(4) Now, if $d(x, y) < 0$ for some $x \neq y$, replace d with $d'(x, y) = d(x, y) - \min_{x,y} d(x, y) + \epsilon$ for some tiny $\epsilon > 0$ to ensure $d'(x, y) > 0$, shifting to nonnegatives while preserving ordering of similarity. Then, apply case (3) above.

Both a metric space and a non-metric space can model online additively separable hedonic games with agent types (see the full version [16] for more details). Here, each location can be seen as a possible agent type, where the preferences of an agent of type x are encoded by a cardinal disutility function $d(x, \cdot)$ (i.e., the negation of her utility), specifying that an agent of type x assigns a cardinal disutility of $d(x, y)$ to each agent of type y.

5.2 Lower and Upper Bounds on Cluster Sizes

Our results can be extended to cases where each cluster m may have any size in an interval $[l_m, u_m]$, provided that $\sum_m l_m \leq n \leq \sum_m u_m$ for feasibility and $l_{\min} := \min_m l_m \geq 2$ with l_m, u_m treated as constant integers. To enforce the lower bounds, we first run DGreedy treating l_m as the size of the cluster m, yielding a clustering $C = (C_m)$ with $|C_m| = l_m$. If $n = \sum_m l_m$, we stop; otherwise, we rerun DGreedy treating u_m as the size of the cluster m with C as the initial clustering; the algorithm already enforces the upper bounds. Our proofs then easily extend while replacing n_1 with $u_{\max} := \max_m u_m$ and n_k with $l_{\min}$ in Lemmas 1 and 4, respectively, including Theorems 2 and 3 that are derived from them.

6 CONCLUSIONS AND FUTURE WORK

We presented a new model for studying online non-centroid clustering *with delays*, where elements that arrive sequentially as points in a finite metric space must be assigned to clusters, but decisions may be delayed at a cost. While the classic *worst-case model* is too pessimistic even in restricted cases, we studied the *unknown i.i.d. model*, under which we developed an online algorithm with a ***constant*** ratio-of-expectations, offering hope for beyond worst-case analysis and the design of effective algorithms in practice.

Our research paves the way for many future works. Immediate directions are studying general delay costs as well as cases where the distribution of the arriving points' locations is known and/or may change over time. Finally, it is worth examining settings where clusterings can be modified with a penalty.

ACKNOWLEDGMENTS

The author acknowledges travel support from a Schmidt Sciences 2025 Senior Fellows award to Michael Wooldridge.

REFERENCES

[1] Sara Ahmadian, Ashkan Norouzi-Fard, Ola Svensson, and Justin Ward. 2019. Better Guarantees for k-Means and Euclidean k-Median by Primal-Dual Algorithms. *SIAM J. Comput.* 49, 4 (2019), FOCS17–97.

[2] Nir Ailon, Moses Charikar, and Alantha Newman. 2008. Aggregating inconsistent information: ranking and clustering. *Journal of the ACM (JACM)* 55, 5 (2008), 1–27.

[3] Nir Ailon, Ragesh Jaiswal, and Claire Monteleoni. 2009. Streaming k-Means Approximation. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 22.

[4] Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. 2001. Local search heuristic for k-median and facility location problems. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*. 21–29.

[5] Itai Ashlagi, Yossi Azar, Moses Charikar, Ashish Chiplunkar, Ofir Geri, Haim Kaplan, Rahul Makhijani, Yuyi Wang, and Roger Wattenhofer. 2017. Min-Cost Bipartite Perfect Matching with Delays. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2017) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 81)*. 1:1–1:20.

[6] Yossi Azar, Ashish Chiplunkar, and Haim Kaplan. 2017. Polylogarithmic Bounds on the Competitiveness of Min-Cost Perfect Matching with Delays. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 1051–1061.

[7] Haris Aziz and Rahul Savani. 2016. Hedonic Games. In *Handbook of Computational Social Choice*, Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia (Eds.). Cambridge University Press, Chapter 15, 356–376.

[8] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. 2004. Correlation clustering. *Machine learning* 56, 1 (2004), 89–113.

[9] Vittorio Bilò, Gianpiero Monaco, and Luca Moscardelli. 2022. Hedonic games with fixed-size coalitions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 9287–9295.

[10] Martin Bullinger and René Romen. 2025. Online coalition formation under random arrival or coalition dissolution. *ACM Transactions on Algorithms* 22, 1 (2025), 1–43.

[11] Ioannis Caragiannis, Evi Micha, and Nisarg Shah. 2024. Proportional Fairness in Non-Centroid Clustering. In *Advances in Neural Information Processing Systems*, Vol. 37. 19139–19166.

[12] T. H. Hubert Chan, Arnaud Guérqin, and Mauro Sozio. 2018. Fully Dynamic k-Center Clustering. In *Proceedings of the 2018 World Wide Web Conference (WWW)*. International World Wide Web Conferences Steering Committee, 579–587.

[13] Moses Charikar, Sudipto Guha, Éva Tardos, and David B. Shmoys. 1999. A Constant-Factor Approximation Algorithm for the k-Median Problem. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing (STOC)*. ACM, 1–10.

[14] Shuchi Chawla, Konstantin Makarychev, Tselil Schramm, and Grigory Yaroslavtsev. 2015. Near optimal LP rounding algorithm for correlation clustering on complete and complete k-partite graphs. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*. 219–228.

[15] Dimitrios Christou, Stratis Skoulakis, and Volkan Cevher. 2023. Efficient online clustering with moving costs. *Advances in Neural Information Processing Systems* 36 (2023), 25325–25364.

[16] Saar Cohen. 2026. Delayed Assignments in Online Non-Centroid Clustering with Stochastic Arrivals. *arXiv preprint arXiv:2601.16091* (2026).

[17] Saar Cohen and Noa Agmon. 2023. Complexity of probabilistic inference in random dichotomous hedonic games. *Proceedings of the AAAI Conference on Artificial Intelligence* 37, 5 (2023), 5573–5581.

[18] Saar Cohen and Noa Agmon. 2023. Online Coalitional Skill Formation. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems, AAMAS*. 494–503.

[19] Saar Cohen and Noa Agmon. 2024. Online Friends Partitioning Under Uncertainty. In *ECAI 2024*. IOS Press, 3332–3339.

[20] Saar Cohen and Noa Agmon. 2024. Online Learning of Partitions in Additively Separable Hedonic Games. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*. 2722–2730.

[21] Saar Cohen and Noa Agmon. 2025. Decentralized Online Learning by Selfish Agents in Coalition Formation. In *Proceedings of the 34th International Joint Conference on Artificial Intelligence (IJCAI)*.

[22] Saar Cohen and Noa Agmon. 2025. Online Learning of Coalition Structures by Selfish Agents. *Proceedings of the AAAI Conference on Artificial Intelligence* 39, 13 (2025), 13709–13717.

[23] Saar Cohen and Noa Agmon. 2025. Online Learning of Fair Coalition Structures. In *Proceedings of the 28th European Conference on Artificial Intelligence, ECAI-25*. 4937–4944.

[24] Vincent Cohen-Addad, Silvio Lattanzi, Andreas Maggiori, and Nikos Parotsidis. 2022. Online and consistent correlation clustering. In *International Conference on Machine Learning*. PMLR, 4157–4179.

[25] Agnes Cseh, Tamás Fleiner, and Petra Harján. 2019. Pareto optimal coalitions of fixed size. *Journal of Mechanism and Institution Design* 4 (2019), 1.

[26] Erik D Demaine, Dotan Emanuel, Amos Fiat, and Nicole Immorlica. 2006. Correlation clustering in general weighted graphs. *Theoretical Computer Science* 361, 2-3 (2006), 172–187.

[27] Derya Dinler and Mustafa Kemal Tural. 2016. *A Survey of Constrained Clustering*. Springer International Publishing, 207–235.

[28] Jacques H Dréze and Joseph Greenberg. 1980. Hedonic coalitions: Optimality and stability. *Econometrica: Journal of the Econometric Society* (1980), 987–1003.

[29] David Eisenstat, Claire Mathieu, and Nicolas Schabanel. 2014. Facility location in evolving metrics. In *International Colloquium on Automata, Languages, and Programming*. Springer, 459–470.

[30] Yuval Emek, Shay Kutten, and Roger Wattenhofer. 2016. Online matching: haste makes waste!. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*. 333–344.

[31] Absalom E. Ezugwu, Abiodun M. Ikotun, Olaide O. Oyelade, Laith Abualigah, Jeffery O. Agushaka, Christopher I. Eke, and Andronicus A. Akinyelu. 2022. A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. *Engineering Applications of Artificial Intelligence* 110 (2022), 104743.

[32] Amos Fiat and Gerhard J Woeginger. 1998. *Online algorithms: The state of the art*. Vol. 1442. Springer.

[33] Foivos Fioravantes, Harmender Gahlawat, and Nikolaos Melissinos. 2025. Exact Algorithms and Lower Bounds for Forming Coalitions of Constrained Maximum Size. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 13847–13855.

[34] Michele Flammini, Gianpiero Monaco, Luca Moscardelli, Mordechai Shalom, and Shmuel Zaks. 2021. On the online coalition structure generation problem. *Journal of Artificial Intelligence Research* 72 (2021), 1215–1250.

[35] Dimitris Fotakis. 2006. Incremental algorithms for Facility Location and k-Median. *Theoretical Computer Science* 361, 2 (2006), 275–313. Approximation and Online Algorithms.

[36] N Garg, A Gupta, Stefano Leonardi, Piotr Sankowski, et al. 2008. Stochastic analyses for online combinatorial optimization problems.. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2008*. 942–951.

[37] S. Geetha, G. Poonthalir, and P. Vanathi. 2009. Improved k-Means Algorithm for Capacitated Clustering Problem. *INFOCOMP Journal of Computer Science* 8, 4 (2009), 52–59.

[38] Ioannis Giotis and Venkatesan Guruswami. 2006. Correlation Clustering with a Fixed Number of Clusters. *Theory OF Computing* 2 (2006), 249–266.

[39] Xiangyu Guo, Janardhan Kulkarni, Shi Li, and Jiayi Xian. 2020. On the Facility Location Problem in Online and Dynamic Models. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2020) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 176)*. 42:1–42:23.

[40] Xiangyu Guo, Janardhan Kulkarni, Shi Li, and Jiayi Xian. 2021. Consistent k-Median: Simpler, Better and Robust. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 130)*. PMLR, 1135–1143.

[41] Anupam Gupta, Guru Guruganesh, and Melanie Schmidt. 2016. Approximation Algorithms for Aversion k-Clustering via Local k-Median. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 55)*. 66:1–66:13.

[42] Shivam Gupta, Shweta Jain, Narayanan C Krishnan, Ganesh Ghalme, and Nandyala Hemachandra. 2024. Capacitated Online Clustering Algorithm. In *ECAI*. 1880–1887.

[43] R. Janani and S. Vijayarani. 2019. Text document clustering using Spectral Clustering algorithm with Particle Swarm Optimization. *Expert Systems with Applications* 134 (2019), 192–200.

[44] Naonori Kakimura and Tomohiro Nakayoshi. 2025. Deterministic primal-dual algorithms for online k-way matching with delays. *Theoretical Computer Science* 1026 (2025), 114988.

[45] Yasushi Kawase and Tomohiro Nakayoshi. 2025. Online Matching with Delays and Size-Based Costs. In *42nd International Symposium on Theoretical Aspects of Computer Science (STACS 2025) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 327)*. 59:1–59:18.

[46] Silvio Lattanzi, Benjamin Moseley, Sergei Vassilvitskii, Yuyan Wang, and Rudy Zhou. 2021. Robust online correlation clustering. *Advances in Neural Information Processing Systems* 34 (2021), 4688–4698.

[47] Silvio Lattanzi and Sergei Vassilvitskii. 2017. Consistent k-Clustering. In *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR, 1975–1984.

[48] Hsuan-Wei Lee, Nishant Malik, Feng Shi, and Peter J. Mucha. 2019. Social clustering in epidemic spread on coevolving networks. *Phys. Rev. E* 99 (2019), 062301. Issue 6.

[49] Chaya Levinger, Noam Hazon, Sofia Simola, and Amos Azaria. 2024. Coalition Formation with Bounded Coalition Size. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*. 1119–1127.

[50] Mathieu Mari, Michał Pawłowski, Runtian Ren, and Piotr Sankowski. 2024. Online Multi-Level Aggregation with Delays and Stochastic Arrivals. In *35th International Symposium on Algorithms and Computation (ISAAC 2024)*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 49–1.
[51] Mathieu Mari, Michał Pawłowski, Runtian Ren, and Piotr Sankowski. 2025. Online matching with delays and stochastic arrival times. *Theory of Computing Systems* 69, 1 (2025), 1–46.
[52] Gianpiero Monaco and Luca Moscardelli. 2023. Nash Stability in Fractional Hedonic Games with Bounded Size Coalitions. In *International Conference on Web and Internet Economics*. Springer, 509–526.
[53] J. M. Mulvey and M. P. Beck. 1984. Solving Capacitated Clustering Problems. *European Journal of Operational Research* 18, 3 (1984), 339–348.
[54] Tai Le Quy, Arjun Roy, Gunnar Friege, and Eirini Ntoutsi. 2021. Fair-Capacitated Clustering. *International Educational Data Mining Society* (2021).
[55] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. 2021. Clustered Federated Learning: Model-Agnostic Distributed Multitask Optimization Under Privacy Constraints. *IEEE Transactions on Neural Networks and Learning Systems* 32, 8 (2021), 3710–3722.
[56] Gerhard J. Woeginger. 2013. Core Stability in Hedonic Coalition Formation. In *SOFSEM 2013: Theory and Practice of Computer Science*. Springer, 33–50.