# Approximation Algorithms for Clustering with Dynamic Points

**Shichuan Deng**
Institute for Interdisciplinary Information Sciences, Tsinghua University, China
dsc15@mails.tsinghua.edu.cn

**Jian Li**
Institute for Interdisciplinary Information Sciences, Tsinghua University, China
lijian83@mail.tsinghua.edu.cn

**Yuval Rabani**
The Rachel and Selim Benin School of Computer Science and Engineering,
The Hebrew University of Jerusalem, Israel
yrabani@cs.huji.ac.il

## Abstract

In many classic clustering problems, we seek to sketch a massive data set of $n$ points (a.k.a *clients*) in a metric space, by segmenting them into $k$ categories or clusters, each cluster represented concisely by a single point in the metric space (a.k.a. the cluster's *center* or its *facility*). The goal is to find such a sketch that minimizes some objective that depends on the distances between the clients and their respective facilities (the objective is a.k.a. the *service cost*). Two notable examples are the $k$-center/$k$-supplier problem where the objective is to minimize the maximum distance from any client to its facility, and the $k$-median problem where the objective is to minimize the sum over all clients of the distance from the client to its facility.

In practical applications of clustering, the data set may evolve over time, reflecting an evolution of the underlying clustering model. Thus, in such applications, a good clustering must simultaneously represent the temporal data set well, but also not change too drastically between time steps. In this paper, we initiate the study of a dynamic version of clustering problems that aims to capture these considerations. In this version there are $T$ time steps, and in each time step $t \in \{1, 2, \dots, T\}$, the set of clients needed to be clustered may change, and we can move the $k$ facilities between time steps. The general goal is to minimize certain combinations of the service cost and the facility movement cost, or minimize one subject to some constraints on the other. More specifically, we study two concrete problems in this framework: the Dynamic Ordered $k$-Median and the Dynamic $k$-Supplier problem. Our technical contributions are as follows:

- We consider the Dynamic Ordered $k$-Median problem, where the objective is to minimize the weighted sum of ordered distances over all time steps, plus the total cost of moving the facilities between time steps. We present one constant-factor approximation algorithm for $T = 2$ and another approximation algorithm for fixed $T \geq 3$.
- We consider the Dynamic $k$-Supplier problem, where the objective is to minimize the maximum distance from any client to its facility, subject to the constraint that between time steps the maximum distance moved by any facility is no more than a given threshold. When the number of time steps $T$ is 2, we present a simple constant factor approximation algorithm and a bi-criteria constant factor approximation algorithm for the outlier version, where some of the clients can be discarded. We also show that it is NP-hard to approximate the problem with any factor for $T \geq 3$.

**2012 ACM Subject Classification** Theory of computation → Facility location and clustering

**Keywords and phrases** clustering, dynamic points, multi-objective optimization

**Digital Object Identifier** 10.4230/LIPIcs.ESA.2020.37

**Related Version** A full version of the paper is available at `https://arxiv.org/abs/2006.14403`.


28th Annual European Symposium on Algorithms (ESA 2020).
Editors: Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders; Article No. 37; pp. 37:1–37:15
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

**Funding** *Shichuan Deng* and *Jian Li*: Supported in part by the National Natural Science Foundation of China Grant 61822203, 61772297, 61632016, 61761146003, the Zhongguancun Haihua Institute for Frontier Information Technology, Turing AI Institute of Nanjing, and Xi'an Institute for Interdisciplinary Information Core Technology.
*Yuval Rabani*: Supported in part by ISF grant number 2553-17.

**Acknowledgements** We want to thank the reviewers for their insightful and constructive comments.

## 1 Introduction

Clustering a data set of points in a metric space is a fundamental abstraction of many practical problems of interest and has been subject to extensive study as a fundamental problem of both machine learning and combinatorial optimization. In particular, cluster analysis is one of the main methods of unsupervised learning, and clustering often models facility location problems.[1] More specifically, some of the most well-studied clustering problems involve the following generic setting. We are given a set $C$ of points in a metric space, and our goal is to compute a set of $k$ centers that optimizes a certain objective function which involves the distances between the points in $C$ and the computed centers. Two prominent examples are the $k$-median problem and the $k$-center problem. They are formally defined as follows. Let $S$ denote the computed set of $k$ cluster centers, let $d(j, S) = \min_{i\in S} d(i, j)$ be the minimum distance from a point $j \in C$ to $S$, and let $D = (d(j, S))_{j\in C}$ be called the *service cost vector*. The $k$-median problem aims to minimize the $L_1$ objective $\|D\|_1 = \sum_{j\in C} d(j, S)$ over the choices of $S$, and the $k$-center aims to minimize the $L_\infty$ objective $\|D\|_\infty = \max_{j\in C} d(j, S)$. In general metric spaces and when $k$ is not a fixed constant, both problems are APX-hard and exhibit constant factor approximation algorithms [4, 6, 12, 23, 30]. An important generalization is the *ordered $k$-median* problem. Here, in addition to $C$ and $k$, we are given also a non-increasing weight vector $w \in \mathbb{R}^{|C|}_{\geq 0}$. Letting $D^{\downarrow}$ denote the sorted version of $D$ in non-increasing order, the objective of ordered $k$-median is to minimize $w \cdot D^{\downarrow}$. This problem generalizes both $k$-center and $k$-median and has attracted significant attention recently and several constant factor approximation algorithms have been developed [3, 7, 9, 10].

In this paper, we study several dynamic versions of the classical clustering problems, in which the points that need to be clustered may change for each time step, and we are allowed to move the cluster centers in each time step, either subject to a constraint on the distance moved, or by incurring a cost proportional to that distance. These versions are motivated in general by practical applications of clustering, where the data set evolves over time, reflecting an evolution of the underlying clustering model. Consider, for instance, a data set representing the active users of a web service, and a clustering representing some meaningful segmentation of the user base. The segmentation should be allowed to change over time, but if it is changed drastically between time steps, then it is probably meaningless. For a more concrete example, consider the following application scenario. There is a giant construction company with several construction teams working in a city. The company has $k$ movable wireless base stations for their private radio communication, and each construction team also has a terminal device. The teams need to put their devices at a certain energy level, in order to maintain the communication channel between the device and the nearest base station. Some construction team may finish some project and move to another place at

[1] In the facility location literature, points are called clients and centers are called facilities, and we will use these terms interchangeably.

some time. Note that the wireless base stations are also movable at a certain expense. Our high level objective is to have all teams covered by the base stations at all times, meanwhile minimizing the energy cost of all teams plus the cost of moving these base stations.

We study two problems of this flavor. The first problem, a dynamic version of the *ordered $k$-median* problem, is a very general model that captures a wide range of dynamic clustering problems where the objective is to minimize the sum of service cost and movement cost. In particular, it generalizes dynamic versions of $k$-center and $k$-median. The problem is defined as follows. We are given a metric space and there are $T$ time steps. In each time step $t$, there is a set $C_t$ of clients that we need to serve. In each time step, we can also choose the locations for $k$ movable facilities to serve the clients (each client is served by its closest facility). Our goal is to minimize the total ordered service distance (i.e., the ordered $k$-median objective), summed over all times steps, plus the total distances traveled by the $k$ movable facilities. We define the problem formally as follows.

▶ **Definition 1** (Dynamic Ordered $k$-Median)**.** *We are given a metric space $(X, d)$. An instance of Dynamic Ordered $k$-Median is specified by $(\{C_t\}_{t=1}^T, \{F_t\}_{t=1}^T, \{w_t \in \mathbb{R}_{\geq 0}^{|C_t|}\}_{t=1}^T, \gamma > 0, k \in \mathbb{N}_+)$, where $T \geq 2$ is a constant integer, $C_t \subset X$ is the set of clients for time $t$, $F_t \subset X$ is the set of candidate locations where we can place facilities. For a vector $v$, denote by $v^{\downarrow}$ the vector derived from $v$ by sorting its entries in non-increasing order. Also denote by $m(X, Y) = \min_{M_0 \in M(X,Y)} \sum_{(i,i') \in M_0} d(i, i')$ the total weight of minimum-weight perfect matching between two equal-sized multi-sets $X, Y$. We are required to compute a sequence of multi-sets of facilities $\{A_t\}_{t=1}^T$ with $A_t \subset F_t, |A_t| = k$, so that the following sum of ordered service cost and movement cost is minimized:*

$$\sum_{t=1}^{T} w_t \cdot (d(j, A_t))_{j \in C_t}^{\downarrow} + \gamma \cdot \sum_{t=1}^{T-1} m(A_t, A_{t+1}). \tag{1}$$

It is also natural to formulate dynamic clustering problems where the objective is to minimize just the service cost, subject to a constraint on the movement cost. This turns out to be technically very different from Dynamic Ordered $k$-Median. Our second problem, which we call Dynamic $k$-Supplier, is such a concrete problem, motivated by the above-mentioned construction company application. In this problem the service cost is the $k$-supplier objective, i.e. the maximum client service distance over all time steps, and the constraints are that any facility cannot be moved further than a fixed value $B > 0$ between any two consecutive time steps. More formally:

▶ **Definition 2** (Dynamic $k$-Supplier)**.** *We are given a metric space $(X, d)$. An instance of Dynamic $k$-Supplier is specified by $(\{C_t\}_{t=1}^T, \{F_t\}_{t=1}^T, B > 0, k \in \mathbb{N}_+)$, where $T \geq 2$ is the number of time steps, $C_t \subset X$ is the set of clients for time $t$, $F_t \subset X$ is the set of candidate locations where we can place facilities. We are required to compute a sequence of multi-sets of facilities $\{A_t\}_{t=1}^T$, with $A_t \subset F_t$, $|A_t| = k$, minimizing the maximum service cost of any client $\max_t \max_{j \in C_t} d(j, A_t)$, subject to the constraint that there must exist a one-to-one matching between $A_t$ and $A_{t+1}$ for any $t$, such that the distance between each matched pair is at most $B$.*

*In the outlier version, we are additionally given the outlier constraints $\{l_t \in \mathbb{N}\}_{t=1}^T$. We are asked to identify a sequence of multi-sets of facilities $\{A_t\}_{t=1}^T$, as well as a sequence of subsets of served clients $\{S_t \subset C_t\}_{t=1}^T$. The goal is to minimize the maximum service cost of any served client $\max_t \max_{j \in S_t} d(j, A_t)$, with the constraints that $A_t \subset F_t, |A_t| = k, |S_t| \geq l_t$, and there must exist a one-to-one matching between $A_t$ and $A_{t+1}$ for any $t$, such that the distance between each matched pair is at most $B$.*

**Note.** The solutions for both Dynamic Ordered $k$-Median and Dynamic $k$-Supplier may be multi-sets, since we allow multiple centers to travel to the same location.

## 1.1 Our Results

We first study Dynamic Ordered $k$-Median. We assume the number of time steps $T$ is a constant and all entries of the weight vector are larger than some small constant $\epsilon > 0$. We present a polynomial-time approximation on general metrics. Moreover, if $T = 2$ we present a constant-factor approximation algorithm without the assumption on the entries of the weight vectors.

▶ **Theorem 3.** 1. *If $T = 2$, for any constant $\delta > 0$ there exists a polynomial-time $(48 + 20\sqrt{3} + \delta)$-approximation for Dynamic Ordered $k$-Median.*
2. *If $T \geq 3$ is a constant and all entries in $\{w_t\}_{t=1}^T$ are at least $\epsilon > 0$, for any constant $\delta > 0$ there exists a polynomial-time $(48 + 20\sqrt{3} + \delta + 6\gamma/\epsilon)$-approximation algorithm for Dynamic Ordered $k$-Median.*

**Our techniques.** The key idea in our algorithm is to design a surrogate relaxed LP as in [7] and embed the fractional LP solution in a network flow instance. We then proceed to round the fractional flow to an integral flow, thus obtaining the integral solution. The network is constructed based on a filtering process introduced by Charikar and Li [12]. We also adapt the oblivious clustering arguments by Byrka et al. [7], but with a slight increase in approximation factors due to the structure of our network flow.

Our approach can also give a constant approximation to the facility-weighted minimum total movement mobile facility location problem (facility-weighted TM-MFL), with a simpler analysis than the previously known local-search based algorithm [1], which achieves an approximation factor of $3 + O(\sqrt{\log\log p/\log p})$ for a $p$-Swap algorithm. The following result is also an improvement over the previously proven factor 499 in [1] when $p = 1$. For more details, we direct the interested readers to the full version of the paper and [1].

▶ **Theorem 4.** *There exists a polynomial-time 10-approximation algorithm for facility-weighted minimum total movement mobile facility location problem.*

As a second result, we consider Dynamic $k$-Supplier and its outlier version. We show that if $T \geq 3$, it is not possible to obtain efficient approximation algorithms for Dynamic $k$-Supplier with *any* approximation factor, unless P = NP, via a simple reduction from perfect 3D matching [27]. However, for the case of $T = 2$, we present a flow-based 3-approximation, which is the best possible factor since vanilla $k$-supplier is NP-hard to approximate within a factor of $(3 - \epsilon)$ for any constant $\epsilon > 0$ [24].

▶ **Theorem 5.**
1. *There exists a 3-approximation for Dynamic $k$-Supplier when $T = 2$.*
2. *There is no polynomial time algorithm for solving Dynamic $k$-Supplier with any approximation factor if $T \geq 3$, unless $P = NP$.*

We also study the outlier version of the problem for $T = 2$. In the outlier version, we can exclude a certain fraction of the clients as outliers in each time step. We obtain a bi-criteria approximation for the problem.

▶ **Theorem 6.** *For any constant $\epsilon > 0$, there exists a bi-criteria 3-approximation algorithm for Dynamic $k$-Supplier with outliers when $T = 2$, that outputs a solution which covers at least $(1 - \epsilon)l_t$ clients within radius $3R^*$ at time $t$, where $t = 1, 2$ and $R^*$ is the optimal radius.*

**Our techniques.** We first guess a constant-size portion of facilities in the optimal solution, remove these facilities and solve the LP relaxation of the remaining problem. This guessing step is standard as in multi-objective optimizations in [20]. From the LP solution, we form clusters as in Harris et al. [22], cast the outlier constraints as budget constraints over the LP solution, and finally round the fractional LP solution to an integral solution using the budgeted optimization methods by Grandoni et al. [20]. Note that since our outlier constraints translate naturally to budget lower bounds, and our optimization goal is minimization, we are only able to achieve bi-criteria approximations instead of pure approximations. For more details, please refer to the full version of this paper.

## 1.2 Other Related Work

The ordered $k$-median problem generalizes a number of classic clustering problems like $k$-center, $k$-median, $k$-facility $l$-centrum, and has been studied extensively in the literature. There are numerous approximation algorithms known for its special cases. We survey here only the results most relevant to our work (ignoring, for instance, results regarding restricted metric spaces or fixed $k$). Constant approximations for $k$-median can be obtained via local search, Lagrangian relaxations and the primal-dual schema, or LP-rounding [4, 6, 25, 26]. Constant approximations for $k$-center are obtained via greedy algorithms [23]. Aouad and Segev [3] employ the idea of surrogate models and give the first $O(\log n)$-approximation for ordered $k$-median. Later, Byrka et al. [7], Chakrabarty and Swamy [9] both successfully design constant-factor approximations for $k$-facility $l$-centrum and ordered $k$-median. Chakrabarty and Swamy [10] subsequently improve the approximation factor for ordered $k$-median to $(5+\epsilon)$, using deterministic rounding in a unified framework.

The outlier setting of clustering problems, specifically for center-type clustering problems, was introduced by Charikar et al. [11] and later further studied by Chakrabarty et al. [8]. Many other variants of different clustering constraints are also extensively studied, including matroid and knapsack center with outliers [13], and fair center-type problems with outliers [22].

Our problems are closely related to the mobile facility location problems (MFL), introduced by Demaine et al. [16]. In these problems, a static set of clients has to be served by a set of facilities that are given initial locations and can be moved to improve the service cost at the expense of incurring a facility movement cost. For the minimum total movement MFL problem (TM-MFL), Friggstad and Salavatipour [18] give an 8-approximation using LP-rounding, where all facilities have unit weights. Ahmadian et al. [1] give a local search algorithm for TM-MFL with weighted facilities using $p$-swaps with an approximation ratio of $3+O(\sqrt{\log\log p/\log p})$, and specifically show that the approximation ratio is at most 499 for $p=1$.

The dynamic formulations of our problems are closely related to the facility location problem with evolving metrics, proposed by Eisenstat et al. [17]. In this problem, there are also $T$ time steps, while the facilities and clients are fixed, and the underlying metric is changing. The total cost is the sum of facility-opening cost, client-serving cost and additional switching costs for each client. The switching cost is paid whenever a client switches facility between adjacent time steps. In comparison, our problem Dynamic $k$-Supplier considers the cost of moving facilities instead of opening costs, and allows the number of clients to change over time. Eisenstat et al. [17] consider the problem when the open facility set $A$ is fixed, and give a $O(\log(nT))$-approximation, where $n$ is the number of clients. They also show a hardness result on $o(\log T)$-approximations. An et al. [2] consider the case when the open facilities are allowed to evolve as well, and give a 14-approximation.

Our problem is also related to stochastic $k$-server [15] and the page migration problem [5, 32]. Dehghani et al. [15] first study the *stochastic $k$-server* problem. In this problem, we also have $T$ time steps, and the distributions $\{P_t\}_{t\in[T]}$ are given *in advance*. The $t$-th client is drawn from $P_t$, and we can use $k$ movable servers. One variant they consider is that, after a client shows up, its closest server goes to the client's location and comes back, and the optimization objective is the total distance travelled by all servers. They provide an $O(\log n)$-approximation for general metrics, where $n$ is the size of the distribution support. In expectation, their objective is the same as in Dynamic Ordered $k$-Median, if we consider non-ordered weighted clients and total weights sum up to 1 for each time slot. However, we note that our result does not imply a constant approximation for their problem. The difficulty is that if one maps the stochastic $k$-server problem to our problem, the corresponding weight coefficient $\gamma$ is not necessarily a constant and our approximation ratio is proportional to $\gamma$. Obtaining a constant factor approximation algorithm for stochastic $k$-server is still an interesting open problem.

## 2 A Constant Approximation for Dynamic Ordered $k$-Median

We devise an LP-based algorithm, which generalizes the oblivious-clustering argument by Byrka et al. [7]. At the center of our algorithm, a network flow method is used, where an integral flow is used to represent our solution.

### 2.1 Flow-based Rounding of LP Solution

We first formulate the LP relaxation. By adding a superscript to every variable to indicate the time step, we denote $x_{ij}^{(t)} \in [0,1]$ the partial assignment of client $j$ to facility $i$ and $y_i^{(t)} \in [0,1]$ the extent of opening facility location $i$ at time step $t$. Moreover, denote $z_{ii'}^{(t)}$ the fractional movement from facility $i$ to facility $i'$, between neighboring time steps $t$ and $t+1$.

The following surrogate LP is designed using the cost reduction trick by Byrka et al. [7]. When the reduced cost functions are exactly guessed, the LP relaxation has an objective value at most the total cost of the optimal solution, denoted by OPT. Call $d' : X \times X \to R_{\geq 0}$ a reduced cost function (not necessarily a metric) of distance function $d$, if for any $x, y \in X$, $d'(x,y) \geq 0$, $d'(x,y) = d'(y,x)$, and $d(x_1,y_1) \leq d(x_2,y_2) \Rightarrow d'(x_1,y_1) \leq d'(x_2,y_2)$. For a sequence of reduced cost functions $\mathfrak{D} = \{d^t\}_{t=1}^T$ of $d$, the modified LP relaxation is defined as follows.

$$\text{minimize}: \sum_{t=1}^{T} \sum_{j\in C_t} \sum_{i\in F_t} d^t(i,j) x_{ij}^{(t)} + \gamma \sum_{t=1}^{T-1} \sum_{i\in F_t} \sum_{i'\in F_{t+1}} d(i,i') z_{ii'}^{(t)} \tag{LP($\mathfrak{D}$)}$$

$$\text{subject to}: \sum_{i\in F_t} x_{ij}^{(t)} = 1, \quad \forall j \in C_t, t \in [T] \tag{2}$$

$$\sum_{i\in F_t} y_i^{(t)} = k, \quad \forall t \in [T] \tag{3}$$

$$0 \leq x_{ij}^{(t)} \leq y_i^{(t)}, \quad \forall i \in F_t, j \in C_t, t \in [T] \tag{4}$$

$$\sum_{i'\in F_{t+1}} z_{ii'}^{(t)} = y_i^{(t)}, \quad \forall i \in F_t, t \in [T-1] \tag{5}$$

$$\sum_{i\in F_t} z_{ii'}^{(t)} = y_{i'}^{(t+1)}, \quad \forall i' \in F_{t+1}, t \in [T-1] \tag{6}$$

Suppose we have solved the corresponding surrogate $\mathsf{LP}(\mathfrak{D})$. In the optimal solution $(x, y, z)$, we assume that whenever $x_{ij}^{(t)} > 0$, we have $x_{ij}^{(t)} = y_i^{(t)}$, via the standard duplication technique of facility locations (for example, see [12]). Denote $\mathsf{Ball}^o(j, R) = \{x \in X : d(x, j) < R\}$ the open ball centered at $j$ with radius $R$, and $E_j^{(t)} = \{i \in F_t : x_{ij}^{(t)} > 0\}$ the relevant facilities for client $j$. For any specific time step $t$, denote $d_{\mathsf{av}}^{(t)}(j) = \sum_{i \in F_t} d(i, j) x_{ij}^{(t)}$ the average *unweighted* service cost of client $j$ and $y^{(t)}(S) = \sum_{i \in S} y^{(t)}(i)$ the amount of fractional facilities in $S \subset F_t$. We perform a filtering-and-matching algorithm (see the full version of this paper) to obtain a subset $C'_t \subset C_t$ for each $t$, a bundle $\mathcal{U}_j^{(t)} \subset F_t$ for each $j \in C'_t$, as well as $P_t$ a partition of $C'_t$, where

1. $C'_t$ is a subset of "well-separated" clients of $C_t$, such that for any client in $C_t \backslash C'_t$, there exists another relatively close client in $C'_t$. To be more precise, for any $j \neq j'$ in $C'_t$, $d(j, j') \geq 4 \max\{d_{\mathsf{av}}^{(t)}(j), d_{\mathsf{av}}^{(t)}(j')\}$, and for any $j'' \in C_t \backslash C'_t$, there exists $j''' \in C'_t$ such that $d_{\mathsf{av}}^{(t)}(j''') \leq d_{\mathsf{av}}^{(t)}(j''), d(j'', j''') \leq 4 \max\{d_{\mathsf{av}}^{(t)}(j''), d_{\mathsf{av}}^{(t)}(j''')\}$;
2. $\mathcal{U}_j^{(t)}$ is a subset of fractionally open facility locations that are relatively close to client $j$;
3. $P_t$ is a judiciously created partition of $C'_t$, where every subset contains either a pair of clients, or a single client. Each pair $\{j, j'\}$ in $P_t$ is chosen such that either $j$ or $j'$ is the closest neighbor of the other, and we guarantee to open a facility location in $\mathcal{U}_j^{(t)}$ or $\mathcal{U}_{j'}^{(t)}$.

The filtering-and-matching algorithm is fairly standard in several LP-based methods for median-type problems (see e.g. [7, 10, 12]). It is worth noting that, while we define the objective value of $\mathsf{LP}(\mathfrak{D})$ using reduced cost functions $\mathfrak{D}$ with respect to the weights, the filtering algorithm is completely oblivious of the weights and only uses the underlying metric $d$.

**Network construction.** We construct an instance of network flow $\mathcal{N}$, and embed the LP solution as a fractional flow $\tilde{f}$. The network $\mathcal{N}$ consists of a source $\mathsf{s}$, a sink $\mathsf{t}$ and $6T$ intermediate layers $L_1, L_2, \ldots, L_{6T}$ arranged in a linear fashion.

For each time step $t \in [T]$, we create two nodes for every pair $p \in P_t$, every bundle $\mathcal{U}_j^{(t)}$ and every candidate facility location $i \in F_t$. All these nodes are contained in the layers $L_{6t-5}, \ldots, L_{6t}$. To distinguish between the two mirror nodes, we use $\mathscr{L}(\cdot)$ and $\mathscr{R}(\cdot)$ to represent the nodes in $\{L_{6t-5}, L_{6t-4}, L_{6t-3}\}$ and the nodes in $\{L_{6t-2}, L_{6t-1}, L_{6t}\}$, respectively. The network is constructed as follows. An example figure is shown in Figure 1.

1. For all $t \in [T]$, add $\mathscr{L}(i)$ to $L_{6t-5}$ and $\mathscr{R}(i)$ to $L_{6t}$ for each $i \in F_t$.
2. For all $t \in [T]$, add $\mathscr{L}(\mathcal{U}_j^{(t)})$ to $L_{6t-4}$ and $\mathscr{R}(\mathcal{U}_j^{(t)})$ to $L_{6t-1}$ for each $\mathcal{U}_j^{(t)}$.
3. For all $t \in [T]$, add $\mathscr{L}(p)$ to $L_{6t-3}$ and $\mathscr{R}(p)$ to $L_{6t-2}$ for each $p \in P_t$.
4. For all $t \in [T], j \in C'_t, p \in P_t$ such that $j \in p$, connect $(\mathscr{L}(\mathcal{U}_j^{(t)}), \mathscr{L}(p)), (\mathscr{R}(p), \mathscr{R}(\mathcal{U}_j^{(t)}))$ *in neighboring layers* with an edge of capacity $\left[\lfloor y^{(t)}(\mathcal{U}_j^{(t)}) \rfloor, \lceil y^{(t)}(\mathcal{U}_j^{(t)}) \rceil\right]$. Let their initial fractional flow values be $\tilde{f}(\mathscr{L}(\mathcal{U}_j^{(t)}), \mathscr{L}(p)) = \tilde{f}(\mathscr{R}(p), \mathscr{R}(\mathcal{U}_j^{(t)})) = y^{(t)}(\mathcal{U}_j^{(t)})$. The capacity is either $[0, 1]$ or $\{1\}$.
5. For all $t \in [T], p \in P_t$, connect $(\mathscr{L}(p), \mathscr{R}(p))$ with an edge of capacity $\left[\lfloor y^{(t)}(p) \rfloor, \lceil y^{(t)}(p) \rceil\right]$, and define $\tilde{f}(\mathscr{L}(p), \mathscr{R}(p)) = y^{(t)}(p) = \sum_{j \in p} y^{(t)}(\mathcal{U}_j^{(t)})$. If $p$ is a normal pair, the capacity is either $[1, 2]$ or $\{1\}$ or $\{2\}$; if $p$ is a singleton pair, the capacity is either $[0, 1]$ or $\{1\}$.
6. For all $t \in [T], j \in C'_t$ and $i \in \mathcal{U}_j^{(t)}$, connect $(\mathscr{L}(i), \mathscr{L}(\mathcal{U}_j^{(t)})), (\mathscr{R}(\mathcal{U}_j^{(t)}), \mathscr{R}(i))$ *in neighboring layers* with an edge of unit capacity. Let the initial fractional flows be $\tilde{f}(\mathscr{L}(i), \mathscr{L}(\mathcal{U}_j^{(t)})) = \tilde{f}(\mathscr{R}(\mathcal{U}_j^{(t)}), \mathscr{R}(i)) = y_i^{(t)}$.

7. For all $t \in [T]$ but $i \in F_t - \bigcup_{j \in C'_t} \mathcal{U}_j^{(t)}$, connect $(\mathscr{L}(i), \mathscr{R}(i))$ with an edge of unit capacity (*across intermediate layers* $L_{6t-4}, \ldots, L_{6t-1}$). Let its initial fractional flow be $\tilde{f}(\mathscr{L}(i), \mathscr{R}(i)) = y_i^{(t)}$.
8. For all $z_{ii'}^{(t)}, i \in F_t, i' \in F_{t+1}$, connect $(\mathscr{R}(i), \mathscr{L}(i'))$ with an edge of unit capacity. Let its initial fractional flow be $\tilde{f}(\mathscr{R}(i), \mathscr{L}(i')) = z_{ii'}^{(t)}$.

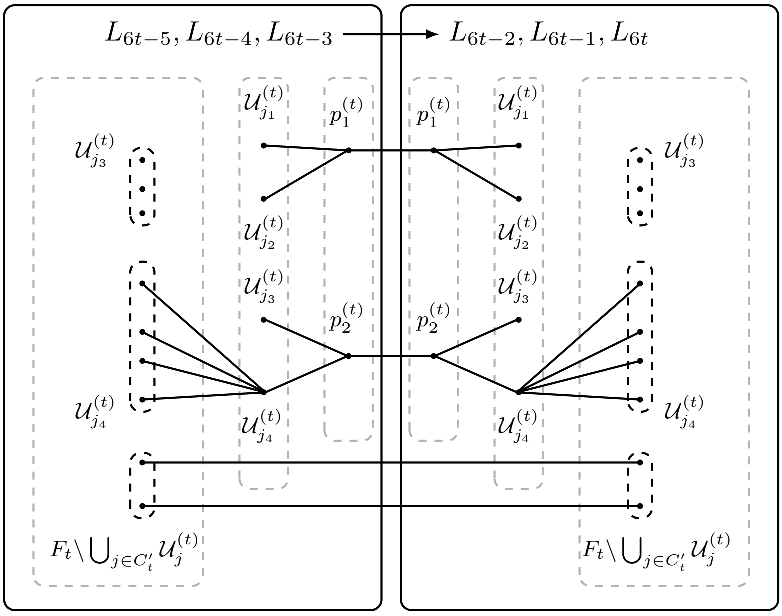


**Figure 1** Some intermediate layers of $\mathcal{N}$ representing a single time step $t$.

Notice $\tilde{f}$ is naturally a flow with value $k$. Since the flow polytope is defined by a totally unimodular matrix, and our capacity constraints are all integers, it is a well-known result (see e.g. [19]) that we can efficiently and stochastically round $\tilde{f}$ to an integral flow $\bar{f}$, such that $\bar{f}$ is guaranteed to have value $k$, and $\mathbb{E}[\bar{f}] = \tilde{f}$. Next, given the integral flow $\bar{f}$, we deterministically construct the facilities to open $\{A_t\}_{t \in [T]}$ as follows.

- If $T = 2$, there are 12 layers $L_1, L_2, \ldots, L_{12}$ in the network. For each link $e = (\mathscr{R}(i_1), \mathscr{L}(i_2))$ between $L_6$ and $L_7$ such that $\bar{f}(e) = 1$, we add the original facility corresponding to $i_1$ to $A_1$, and the original facility of $i_2$ to $A_2$.
- If $T \geq 3$, the integral flow $\bar{f}$ may enter $L_{6t-5}$ and exit from $L_{6t}$ at sets of different facility locations. For illustration, denote $A_{t,1}$ the set in $L_{6t-5}$ and $A_{t,2}$ the set in $L_{6t}$. Notice it may happens that $|A_{t,1} \bigcup A_{t,2}| > k$ and we cannot open them both, so we design an algorithm to find $A_t \subseteq A_{t,1} \bigcup A_{t,2}$ and $|A_t| = k$, and open the facilities in $A_t$ for time $t$. The algorithm looks at each pair $(j_1, j_2) = p \in P_t$, and consider the 1 or 2 units of flow $\bar{f}$ on the link $(\mathscr{L}(p), \mathscr{R}(p))$. For a facility $i$, if there is one unit of flow through $\mathscr{L}(i)$ or $\mathscr{R}(i)$, we call the facility $i$ *activated*. But if $\mathscr{L}(i_1)$ and $\mathscr{R}(i_2)$ are activated and $i_1, i_2 \in \mathcal{U}_{j_1}, i_1 \neq i_2$, we only open one of them. The same is true when $i_1 \in \mathcal{U}_{j_1}, i_2 \in \mathcal{U}_{j_2}, i_1 \neq i_2$.
  For each unit flow, our algorithm either always choose $i_1$ to open where $\mathscr{L}(i_1)$ is activated, or always choose the facility in $\mathcal{U}_{j_2}$ if $j_1$ is not the closest neighbor of $j_2$. As a result, we give the following lemma estimating the movement cost, and the detailed algorithm can be found in the full version of this paper.

▶ **Lemma 7.** *Let $d(A, A')$ denote the cost of minimum weight matching between $A, A'$. If $T = 2$, the expected movement cost of solution $\{A_1, A_2\}$ satisfies*

$$\mathbb{E}[d(A_1, A_2)] = \sum_{i \in F_1} \sum_{i' \in F_2} d(i, i') z^{(1)}_{ii'}.$$

*If $T \geq 3$, the expected movement cost of solution $\{A_t\}_{t=1}^T$ after rerouting satisfies*

$$\mathbb{E}\left[\sum_{t \in [T-1]} d(A_t, A_{t+1})\right] \leq \sum_{t \in [T-1]} \sum_{i \in F_t} \sum_{i' \in F_{t+1}} d(i, i') z^{(t)}_{ii'} + 6 \sum_{t=1}^{T} \sum_{j \in C_t} d^{(t)}_{\mathsf{av}}(j).$$

## 2.2 From Rectangular to General Cases

We first provide a lemma to bound the stochastic $k$-facility $l$-centrum cost of $A_t$ for any fixed time $t$. Consequently, the ordered cost can be nicely bounded as well. The proof of the following lemma can be found in the full version of this paper.

▶ **Lemma 8** (adapted from [7]). *Fix $t \in [T]$ and let $m \in \mathbb{N}_+, h > 0$. Define $\mathsf{rect}(a, b)$ the rectangular vector of length $b$, where the first $a$ elements are 1s and the rest are 0s. For $A_t$ as the (random) set of activated locations returned by our algorithm, and $d(C_t, A_t) = (d(j, A_t))_{j \in C_t}$ as the service cost vector, we have*

$$\mathbb{E}_{A_t}[\mathsf{rect}(m, |C_t|) \cdot d(C_t, A_t)^{\downarrow}] \leq (24 + 10\sqrt{3}) m \cdot h + (24 + 10\sqrt{3}) \sum_{j \in C_t} d^{-h}{}^{(t)}_{\mathsf{av}}(j),$$

*where $d^{-h}(j, j') = 0$ if $d(j, j') < h$ and $d^{-h}(j, j') = d(j, j')$ otherwise. Similar to $d_{\mathsf{av}}(j)$, the average clipped service cost $d^{-h}{}^{(t)}_{\mathsf{av}}(j)$ is defined as $d^{-h}{}^{(t)}_{\mathsf{av}}(j) = \sum_{i \in F_t} d^{-h}(i, j) x^{(t)}_{ij}$.*

Finally we turn to the generally-weighted case, where the weight vectors $w_t, t \in [T]$ are not necessarily rectangular ones like $\mathsf{rect}(m, |C_t|)$. The guessing of underlying reduced cost functions $\mathfrak{D}$ is exactly the same as in Byrka et al. [7], thus omitted here. We solve $\mathsf{LP}(\mathfrak{D})$ using these induced reduced cost functions and proceed accordingly. The following lemma is similar to that of Lemma 5.1 in Byrka et al. [7], and the proof can be found in the full version of this paper.

▶ **Lemma 9.** *When $T = 2$, the procedure described above is a $(48 + 20\sqrt{3})$-approximation for Dynamic Ordered $k$-Median.*

*If $T \geq 3$ is a constant and the smallest entry in $\{w_t\}_{t=1}^T$ is at least some constant $\epsilon > 0$, the above-described procedure is a $(48 + 20\sqrt{3} + 6\gamma/\epsilon)$-approximation for Dynamic Ordered $k$-Median.*

*In both cases, the procedure makes $O\left(\prod_{t=1}^T (|F_t| \cdot |C_t|)^{N_t}\right)$ calls to its subroutines, where $N_t$ is the number of distinct entries in the weight vector $w_t$, $t \in [T]$.*

Fix some positive parameter $\delta > 0$ and recall the distance bucketing trick by Aouad and Segev [3]. When $T$ is a constant, it is possible to guess the largest service distance for each time step by paying a polynomial factor in the running time. Then we make logarithmically many buckets for each time step to hold the service cost values of clients. For each bucket, its average weight is also guessed up to a small multiplicative error $(1 + \delta)$. Since there are at most $O\left(\log_{1+\delta}\left(\frac{n}{\delta}\right)\right) = O\left(\frac{1}{\delta}\log\left(\frac{n}{\delta}\right)\right)$ buckets for each time step, where $n = |F| + |C|$, guessing a non-increasing sequence of the average weights only causes another polynomial factor $\exp\left(O\left(\frac{1}{\delta}\log\left(\frac{n}{\delta}\right)\right)\right) = n^{O(1/\delta)}$. Finally, because $T$ is a constant, the overall number of guesses is still bounded by a polynomial. For more details, see [3, 7].

▶ **Theorem 10.** *When $T = 2$, for any $\delta > 0$ there exists a $(48+20\sqrt{3})(1+\delta)$-approximation algorithm for Dynamic Ordered $k$-Median, with running time $(|F_1| + |C_1|)^{O(1/\delta)} \cdot (|F_2| + |C_2|)^{O(1/\delta)}$.*

*When $T \geq 3$ is a constant, and the smallest entry in $\{w_t\}_{t=1}^T$ is at least some constant $\epsilon > 0$, for any $\delta > 0$ there exists a $\left(48 + 20\sqrt{3} + 6\gamma/\epsilon\right)(1 + \delta)$-approximation algorithm for Dynamic Ordered $k$-Median, with running time $\prod_{t\in[T]} (|F_t| + |C_t|)^{O(1/\delta)}$.*

**Proof.** This is almost a direct consequence of Theorem 5.2 in [7], with the constant factor replaced by our $\mu = 24+10\sqrt{3}$. Notice that we need to slightly modify the way of constructing rounded weights $\{w^*_t\}_{t=1}^T$ in the following way,

$$\forall t \in [T], r \in [|C_t|], w^*_{tr} = \begin{cases} w_{t1} & r = 1, \\ \min\left\{(1+\delta)^{\lceil \log_{1+\delta} w_{tr} \rceil}, w_{t1}\right\} & w_{tr} \geq \epsilon w_{t1}/|C_t|, r \neq 1, \\ \epsilon w_{t1}/|C_t| & w_{tr} < \epsilon w_{t1}/|C_t|, \end{cases}$$

so that the perturbed weight vectors are rounded larger, but at most $(1 + \delta)$ times larger in terms of the overall objective, and there are $O(\log_{1+\delta}(|C_t|/\delta))$ different values in $w^*_t$.

Plugging in the approximations of individual time steps does not affect the analysis of movement costs in the proof of Lemma 9, hence the approximation factor follows. We omit the technical details here due to space limit. They can be be found in Appendix D of [7]. ◀

## 3 Approximating Dynamic $k$-Supplier

We present a flow-based algorithm that gives a 3-approximation for Dynamic $k$-Supplier when $T = 2$, and show it is NP-hard to obtain polynomial-time approximation algorithms for Dynamic $k$-Supplier with any approximation factor when $T \geq 3$. We also briefly introduce our bi-criteria approximation algorithm for Dynamic $k$-Supplier with outliers and $T = 2$, while the detailed algorithm and analysis can be found in the full version of this paper.

### 3.1 A 3-Approximation for Dynamic $k$-Supplier, $T = 2$

In contrast to the NP-hardness of approximating Dynamic $k$-Supplier for $T \geq 3$, we consider Dynamic $k$-Supplier when $T = 2$ on general metrics and present a simple flow-based constant approximation. Suppose we are given the client sets $C_1, C_2$ and $F_1, F_2$ as candidate facility locations and the movement constraint is $B > 0$.

First, since the optimal radius $R^*$ is obviously the distance between some client and some facility location, we assume we have successfully guessed the optimal radius $R^*$ (using binary search). Next, we construct the following network flow instance $\mathcal{G}(\mathcal{V}, \mathcal{E})$. $\mathcal{V}$ consists of 4 layers of vertices (two layers $\mathcal{L}^{11}, \mathcal{L}^{12}$ for $t = 1$, two layers $\mathcal{L}^{21}, \mathcal{L}^{22}$ for $t = 2$), a source $\mathsf{s}$ and sink $\mathsf{t}$. We define the layers and links in $\mathcal{G}$ as follows:

- For each $i \in F_1$, add a vertex in $\mathcal{L}^{12}$. For $i' \in F_2$, add a vertex in $\mathcal{L}^{21}$.
- Repeatedly pick an arbitrary client $j \in C_1$ and remove from $C_1$ every client within distance $2R^*$ from $j$. Denote these clients a new cluster corresponding to $j$. Since we have guessed the optimal radius $R^*$, it is easy to see we can get at most $k$ such clusters. And if there are less than $k$ clusters, we create some extra dummy clusters to obtain exactly $k$ clusters, while dummy clusters do not correspond to any client. For each cluster, add a vertex to $\mathcal{L}^{11}$. Repeat this for $C_2$ and form $\mathcal{L}^{22}$.
- The four layers are arranged in order as $\mathcal{L}^{11}, \mathcal{L}^{12}, \mathcal{L}^{21}, \mathcal{L}^{22}$. With a slight abuse of notation, for $u \in \mathcal{L}^{11}, v \in \mathcal{L}^{12}$, connect them using a link with unit capacity if $d(u, v) \leq R^*$; for $w \in \mathcal{L}^{21}, z \in \mathcal{L}^{22}$, connect them using a link with unit capacity if $d(w, z) \leq R^*$. For $v \in \mathcal{L}^{12}, w \in \mathcal{L}^{21}$, connect them using a link with *unbounded* capacity if $d(v, w) \leq B$.

- Connect every dummy cluster in $\mathcal{L}^{11}$ with every facility location vertex in $\mathcal{L}^{12}$. Connect every dummy cluster in $\mathcal{L}^{22}$ with every facility location vertex in $\mathcal{L}^{21}$. Every such link has unit capacity.
- Finally, the source $\mathsf{s}$ is connected to every vertex in $\mathcal{L}^{11}$ and the sink $\mathsf{t}$ is connected to every vertex in $\mathcal{L}^{22}$, with every edge having unit capacity.

▶ **Lemma 11.** *$\mathcal{G}(\mathcal{V}, \mathcal{E})$ admits a flow of value $k$. Moreover, we can obtain a feasible solution of cost at most $3R^*$ from an integral flow of value $k$ in $\mathcal{G}(\mathcal{V}, \mathcal{E})$.*

**Proof.** As an optimal solution with objective $R^*$, there exist two multi-sets $A_1 \subset F_1, A_2 \subset F_2$ such that $|A_1| = |A_2| = k$ and there exists a perfect matching between them. For any $i \in F_1, i' \in F_2$, if the pair $(i, i')$ appears $m$ times in the perfect matching, define a flow value $f(i, i') = m$ over link $(i, i')$.

Consider the first time step. For any facility location $i$ and clusters $j, j'$, either $d(i, j)$ or $d(i, j')$ is larger than $R^*$, otherwise $d(j, j') \leq 2R^*$, contradicting with our construction. Because $A_1$ also covers all $j \in C_1$ with radius $R^*$, for every $j \in \mathcal{L}^{11}$, we can always find a different element $i \in A_1$ such that $d(i, j) \leq R^*$, and we add a unit flow as $f(j, i) = 1$. The same process is repeated for $\mathcal{L}_{22}$ and $A_2$.

The total flow between $\mathcal{L}^{12}$ and $\mathcal{L}^{21}$ is now obviously $k$, since the perfect matching between $A_1$ and $A_2$ has size $k$. After the construction of unit flows for non-dummy clusters, we arbitrarily direct the remaining flows from facility locations to the dummy clusters, one unit each time. Finally, for any cluster with unit flow, define the flow between it and the source/sink to be 1. This completes an integral flow of value $k$ on $\mathcal{G}$.

For the second part, suppose we have an integral flow $\bar{f}$ of value $k$ on $\mathcal{G}$. For any facility location $i \in F_1$, denote $g(i)$ the total flow through $i$. We place $g(i)$ facilities at location $i$, and repeat the same procedures for $i' \in F_2$. If $\bar{f}(i, i') = m$ for $i \in F_1, i' \in F_2$, move $m$ facilities from $i$ to $i'$ in the transition between 2 time steps.

For any $j' \in C_1$, if $j$ is the cluster center it belongs to, there exists a facility at most $d(j', i) \leq d(j', j) + d(j, i) \leq 3R^*$ away. ◀

▶ **Theorem 12.** *There exists a 3-approximation for Dynamic $k$-Supplier when $T = 2$.*

**Proof.** Consider the network flow instance we construct. It only has integer constraints and the coefficient matrix is totally unimodular. Moreover, there exists a flow of value $k$ due to Lemma 11, hence we can efficiently compute an integral solution $\bar{f}$ of value $k$, thus obtaining a 3-approximation solution. ◀

## 3.2 The Hardness of Approximating Dynamic $k$-Supplier, $T \geq 3$

We show it is NP-hard to design approximation algorithms for Dynamic $k$-Supplier with any approximation factor when $T \geq 3$. The proof is via reduction from the perfect 3D matching problem, which is known to be NP-Complete [27].

▶ **Theorem 13.** *There is no polynomial time algorithm for solving Dynamic $k$-Supplier with any approximation factor if $T \geq 3$, unless $P = NP$.*

**Proof.** We reduce an arbitrary instance of perfect 3D-matching to Dynamic $k$-Supplier to show the NP-hardness. Recall for an instance of perfect 3D-matching, we are given three finite sets $A, B, C$ with $|A| = |B| = |C|$, and a triplet set $\mathcal{T} \subset A \times B \times C$. Suppose $|A| = n$ and $|\mathcal{T}| = m$, and we are asked to decide whether there exists a subset $\mathcal{S} \subset \mathcal{T}$, such that $|\mathcal{S}| = n$, and each element in $A, B, C$ appears exactly once in some triplet in $\mathcal{S}$. We construct the following graph $G = (V, E)$, where $V, E$ are initially empty.

- For each triplet $g = (a, b, c) \in \mathcal{T}$, add three new vertices $a_g, b_g, c_g$ to $V$ correspondingly. Connect $a_g, b_g$ with an edge of length $\alpha$. Connect $b_g, c_g$ with an edge of length $\alpha$.
- Denote $V_A$ all the vertices that correspond to vertices in $A$. Similarly for $V_B$ and $V_C$.
- For any two vertices in $V_A$ corresponding to the same element $a \in A$, connect them with an edge of length 1. Repeat the same procedure for $V_B, V_C$.

Assume we are able to solve Dynamic $k$-Supplier for $T = 3$ with an approximation factor $\alpha$. We solve Dynamic $k$-Supplier for $G$ on its graph metric $d_G$, with $k = n$ and the movement constraint $B = \alpha$, where the client sets are $\{V_A, V_B, V_C\}$ and facility sets are $\{V_A, V_B, V_C\}$ for the three time steps, respectively.

It is easy to see that the reduced Dynamic $k$-Supplier instance has covering radius $R^* = 1$ if and only if there exists a perfect 3D-matching, otherwise the covering radius is at least $2\alpha + 1$. Since our approximation factor is $\alpha$, this concludes the NP-hardness of approximation algorithms with any factor for Dynamic $k$-Supplier when $T \geq 3$. ◀

## 3.3 A Bi-criteria Approximation for Dynamic $k$-Supplier with Outliers

Lastly, we present our bi-criteria approximation algorithm that solves Dynamic $k$-Supplier, when $T = 2$ and outliers are allowed. As a useful ingredient, let us first briefly review the *$m$-budgeted bipartite matching* problem. The input consists of a bipartite graph $G = (V, E)$, and each edge $e \in E$ is associated with a weight $w(e) \geq 0$ and $m$ types of lengths $f_i(e) \geq 0, i = 1, \ldots, m$. The problem asks for a maximum weight matching $M$ with $m$ budget constraints, where the $i$th constraint is that the sum of all $f_i$ lengths in $M$ is no more than $L_i$, i.e. $\sum_{e \in M} f_i(e) \leq L_i$. When the number of constraints $m$ is a constant, a pure $(1 - \epsilon)$-approximation algorithm for any constant $\epsilon > 0$ is devised by Grandoni et al. [20].

**Sketch.** Due to space limit, we provide a sketch here and defer the details to the full version of this paper. Consider Dynamic $k$-Supplier with outliers and $T = 2$. In the solution, we place $k$ facilities for time $t = 1$, serving in total at least $l_1$ clients in $C_1$, then move each of these facilities for a distance at most $B$ to serve at least $l_2$ clients in $C_2$, and the maximum service distance is our minimization goal. Clearly, the optimal solution $R^*$ only has a polynomial number of possible values and can be guessed efficiently, so we assume that $R^*$ is known to us in the following analysis.

For a fixed $R^*$, denote $c_i$ the number of clients that facility location $i$ can serve within distance $R^*$. We assign two lengths $f_1(e) = c_i, f_2(e) = c_{i'}$ and weight $w(e) = 1$ for every candidate edge $e = (i, i')$, where $i \in F_1, i' \in F_2$. By duplicating each possible facility location in $F_1$ and $F_2$ and only allowing vertices within distance $B$ to be matched, the required solution can be fully represented by a $k$-cardinality matching $M$ between $F_1$ and $F_2$. Let us temporarily assume that any client miraculously contribute only once to the total number of clients served. Then the problem naturally translates to deciding whether there exists a bipartite matching $M$ between $F_1$ and $F_2$ (with candidate facility locations duplicated) with weight $k$, such that the sum of all $f_1$ lengths in $M$ is at least $l_1$, and the sum of all $f_2$ lengths in $M$ is at least $l_2$.

This new problem is very similar to 2-*budgeted bipartite matching*, but there are still some major differences. In the approximation algorithm in [20], every integral matching $M$ is obtained by first finding a *feasible* fractional matching $M'$, which has at most $2m$ edges being fractional, and then dropping these fractionally-matched edges completely. Back to our problem where $m = 2$. If we obtain such a fractional solution $M'$ which satisfies the constraints and only has at most 4 fractional edges, we would like to find an integral matching $M$ in a way that uses more "budget" instead of using less, so as to cover at least as many clients as $M'$ does and not violate any budget constraint (in other words, outlier constraints), and we have to drop these fractional edges again from $M'$.

Contrary to 2-*budgeted bipartite matching*, we want to control the portion of budget dropped in this case. We achieve this by guessing a constant number of edges, which has either the top-$\theta$ $f_1$ lengths or top-$\theta$ $f_2$ lengths in the optimal solution, using a suitably chosen constant $\theta > 0$. We are able to devise a bi-criteria approximation algorithm that violates both budget constraints by any small constant $\epsilon$-portion. The bi-criteria method is developed in line with the multi-criteria approximation schemes in [20].

To fully avoid counting any served client multiple times, whenever we duplicate a facility location, we make sure that only one copy induces non-zero lengths on edges that reside on it. We also use a greedy algorithm to remove some facility locations in $F_1, F_2$ and form client clusters around the remaining ones. Now, instead of defining $c_i$ as the number of clients that facility location $i$ can serve within distance $R^*$, we change $c_i$ to the number of clients that are gathered around $i$. More specifically, for client set $C_t$ and facility location set $F_t$, we find a subset $F'_t \subset F_t$ and a corresponding sub-partition $\{K_i\}_{i\in F'_t}$ of $C_t$ (i.e., $K_i$s are pair-wise disjoint and their union is a subset of $C_t$), such that $\forall j \in K_i, d(i,j) \leq 3R^*$ and we define $c_i = |K_i|$ for $i \in F'_t$, $c_i = 0$ for $i \in F_t \backslash F'_t$. Using this method, every client is counted at most once in all $c_i$s, hence its contribution to the total number is always at most 1. The same filtering process can be found in [22]. See the full version of the paper for more details.

## 4 Future Directions

We list some interesting future directions and open problems.

1. It would be very interesting to remove the dependency of $\gamma$ (the coefficient of movement cost) and $\epsilon$ (the lower bound of the weight) from the approximation factor for Dynamic Ordered $k$-Median in Theorem 10, or show such dependency is inevitable. We leave it as an important open problem. We note that a constant approximation factor for Dynamic Ordered $k$-Median without depending on $\gamma$ would imply a constant approximation for stochastic $k$-server, for which only a logarithmic-factor approximation algorithm is known [15].
2. Our approximation algorithm for Dynamic Ordered $k$-Median is based on the technique developed in Byrka et al. [7]. The original ordered $k$-median problem has subsequently seen improved approximation results in [9, 10]. We did not try hard to optimize the constant factors. Nevertheless, it is an interesting future direction to further improve the constant approximation factors by leveraging the techniques from [9, 10] or other ideas.
3. From Theorem 13, we can see that Dynamic $k$-Supplier is hard to approximate when $T \geq 3$. However, it makes sense to relax the hard constraint $B$ (we allow the distance a facility can move be at most $\alpha B$ for some constant $\alpha$).

It is possible to formulate other concrete problems that naturally fit into the dynamic clustering theme and are well motivated by realistic applications, but not yet considered in the paper. For example, one can use the $k$-median objective for the service cost and the maximum distance of any facility movement as the movement cost. One can also consider combining the cost in more general fashion like in [10], or extending the problems to the fault-tolerant version [21, 28, 31] or the capacitated version [14, 29].

## References


1 Sara Ahmadian, Zachary Friggstad, and Chaitanya Swamy. Local-search based approximation algorithms for mobile facility location problems. In *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*, pages 1607–1621, 2013. doi:10.1137/1.9781611973105.115.

2 Hyung-Chan An, Ashkan Norouzi-Fard, and Ola Svensson. Dynamic facility location via exponential clocks. *ACM Transactions on Algorithms*, 13(2):1–20, 2017. doi:10.1145/2928272.

3 Ali Aouad and Danny Segev. The ordered $k$-median problem: surrogate models and approximation algorithms. *Mathematical Programming*, 177(1-2):55–83, 2019. doi:10.1007/s10107-018-1259-3.

4 Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. Local search heuristics for $k$-median and facility location problems. *SIAM Journal on Computing*, 33(3):544–562, 2004. doi:10.1137/S0097539702416402.

5 David L Black, Anoop Gupta, and Wolf-Dietrich Weber. Competitive management of distributed shared memory. *COMPCON Spring 89*, pages 184–185, 1989. doi:10.1109/CMPCON.1989.301925.

6 Jarosław Byrka, Thomas Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. An improved approximation for $k$-median and positive correlation in budgeted optimization. *ACM Transactions on Algorithms*, 13(2):1–31, 2017. doi:10.1145/2981561.

7 Jarosław Byrka, Krzysztof Sornat, and Joachim Spoerhase. Constant-factor approximation for ordered $k$-median. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 620–631, 2018. doi:10.1145/3188745.3188930.

8 Deeparnab Chakrabarty, Prachi Goyal, and Ravishankar Krishnaswamy. The non-uniform $k$-center problem. In *43rd International Colloquium on Automata, Languages, and Programming*, volume 55, page 67, 2016. doi:10.4230/LIPIcs.ICALP.2016.67.

9 Deeparnab Chakrabarty and Chaitanya Swamy. Interpolating between $k$-median and $k$-center: Approximation algorithms for ordered $k$-median. In *45th International Colloquium on Automata, Languages, and Programming*, volume 107, page 29, 2018. doi:10.4230/LIPIcs.ICALP.2018.29.

10 Deeparnab Chakrabarty and Chaitanya Swamy. Approximation algorithms for minimum norm and ordered optimization problems. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 126–137, 2019. doi:10.1145/3313276.3316322.

11 Moses Charikar, Samir Khuller, David M Mount, and Giri Narasimhan. Algorithms for facility location problems with outliers. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 642–651, 2001. URL: http://dl.acm.org/citation.cfm?id=365411.365555.

12 Moses Charikar and Shi Li. A dependent LP-rounding approach for the $k$-median problem. In *Proceedings of the 39th International Colloquium on Automata, Languages, and Programming-Volume Part I*, pages 194–205, 2012. doi:10.1007/978-3-642-31594-7_17.

13 Danny Z Chen, Jian Li, Hongyu Liang, and Haitao Wang. Matroid and knapsack center problems. *Algorithmica*, 75(1):27–52, 2016. doi:10.1007/s00453-015-0010-1.

14 Julia Chuzhoy and Yuval Rabani. Approximating $k$-median with non-uniform capacities. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 952–958, 2005. URL: http://dl.acm.org/citation.cfm?id=1070432.1070569.

15 Sina Dehghani, Soheil Ehsani, MohammadTaghi Hajiaghayi, Vahid Liaghat, and Saeed Seddighin. Stochastic $k$-server: How should Uber work? In *44th International Colloquium on Automata, Languages, and Programming*, volume 80, page 126, 2017. doi:10.4230/LIPIcs.ICALP.2017.126.

16 Erik D Demaine, MohammadTaghi Hajiaghayi, Hamid Mahini, Amin S Sayedi-Roshkhar, Shayan Oveisgharan, and Morteza Zadimoghaddam. Minimizing movement. *ACM Transactions on Algorithms*, 5(3):30, 2009. doi:10.1145/1541885.1541891.

17 David Eisenstat, Claire Mathieu, and Nicolas Schabanel. Facility location in evolving metrics. In *International Colloquium on Automata, Languages, and Programming*, pages 459–470, 2014. doi:10.1007/978-3-662-43951-7_39.

18 Zachary Friggstad and Mohammad R Salavatipour. Minimizing movement in mobile facility location problems. *ACM Transactions on Algorithms*, 7(3):28, 2011. doi:10.1145/1978782.1978783.

19 Rajiv Gandhi, Samir Khuller, Srinivasan Parthasarathy, and Aravind Srinivasan. Dependent rounding and its applications to approximation algorithms. *Journal of the ACM*, 53(3):324–360, 2006. doi:10.1145/1147954.1147956.

20 Fabrizio Grandoni, R Ravi, Mohit Singh, and Rico Zenklusen. New approaches to multi-objective optimization. *Mathematical Programming*, 146(1-2):525–554, 2014. doi:10.1007/s10107-013-0703-7.

21 Mohammadtaghi Hajiaghayi, Wei Hu, Jian Li, Shi Li, and Barna Saha. A constant factor approximation algorithm for fault-tolerant $k$-median. *ACM Transactions on Algorithms*, 12(3):36, 2016. doi:10.1145/2854153.

22 David G Harris, Thomas Pensyl, Aravind Srinivasan, and Khoa Trinh. A lottery model for center-type problems with outliers. *ACM Transactions on Algorithms*, 15(3):1–25, 2019. doi:10.1145/3311953.

23 Dorit S Hochbaum and David B Shmoys. A best possible heuristic for the $k$-center problem. *Mathematics of Operations Research*, 10(2):180–184, 1985. doi:10.1287/moor.10.2.180.

24 Dorit S Hochbaum and David B Shmoys. A unified approach to approximation algorithms for bottleneck problems. *Journal of the ACM*, 33(3):533–550, 1986. doi:10.1145/5925.5933.

25 Kamal Jain and Vijay V Vazirani. Primal-dual approximation algorithms for metric facility location and $k$-median problems. In *40th Annual Symposium on Foundations of Computer Science*, pages 2–13, 1999. doi:10.1109/SFFCS.1999.814571.

26 Kamal Jain and Vijay 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(2):274–296, 2001. doi:10.1145/375827.375845.

27 Richard M Karp. Reducibility among combinatorial problems. *50 Years of Integer Programming 1958-2008*, pages 219–241, 2010. doi:10.1007/978-3-540-68279-0_8.

28 Samir Khuller, Robert Pless, and Yoram J Sussmann. Fault tolerant $k$-center problems. *Theoretical Computer Science*, 242(1-2):237–245, 2000. doi:10.1016/S0304-3975(98)00222-9.

29 Shi Li. Approximating capacitated $k$-median with $(1+\varepsilon)k$ open facilities. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 786–796, 2016. doi:10.1137/1.9781611974331.ch56.

30 Shi Li and Ola Svensson. Approximating $k$-median via pseudo-approximation. In *Proceedings of the forty-fifth annual ACM symposium on Theory of Computing*, pages 901–910, 2013. doi:10.1145/2488608.2488723.

31 Chaitanya Swamy and David B Shmoys. Fault-tolerant facility location. *ACM Transactions on Algorithms*, 4(4):1–27, 2008. doi:10.1145/1383369.1383382.

32 Jeffery Westbrook. Randomized algorithms for multiprocessor page migration. *SIAM Journal on Computing*, 23(5):951–965, 1994. doi:10.1137/S0097539791199796.