Dynamic Clustering to Minimize the Sum of Radii[*]

Monika Henzinger[1], Dariusz Leniowski[2], and Claire Mathieu[3]

1   University of Vienna, Faculty of Computer Science, Vienna, Austria
    `monika.henzinger@univie.ac.at`
2   University of Vienna, Faculty of Computer Science, Vienna, Austria
    `dariusz.leniowski@univie.ac.at`
3   ENS, CNRS, PSL Research University, Paris, France
    `cmathieu@di.ens.fr`

―――― **Abstract** ――――――――――――――――――――――――――――――――――

In this paper, we study the problem of opening centers to cluster a set of clients in a metric space so as to minimize the sum of the costs of the centers and of the cluster radii, in a dynamic environment where clients arrive and depart, and the solution must be updated efficiently while remaining competitive with respect to the current optimal solution. We call this *dynamic sum-of-radii clustering* problem.

We present a data structure that maintains a solution whose cost is within a constant factor of the cost of an optimal solution in metric spaces with bounded doubling dimension and whose worst-case update time is logarithmic in the parameters of the problem.

1   Introduction

The main goal of clustering is to partition a set of objects into homogeneous and well separated subsets (clusters). Clustering techniques have long been used in a wide variety of application areas, see for example the excellent surveys [25, 21].

There are several ways to model clustering. Among them, the problem of sum-of-radii (or sum-of-diameter) clustering has been extensively studied: the clients are located in a metric space and one must open facilities to minimize facility opening cost (or keep the number of open facilities limited to at most k) plus the sum of the cluster radii (or, in other applications, cluster diameters). To give a concrete example, imagine a telecommunications agency setting up mobile towers that provide wireless access to selected clients, incurring costs for setting up towers as well as for configuring a tower to serve the customers lying within a certain distance, where that latter contribution to the cost increases with the maximum distance served by the tower.

Assume the number of facilities is limited to k. For sum-of-diameter clustering, Doddi, Marathe, Ravi, Taylor and Widmayer [15] prove hardness of approximation to within better than a factor of 2. More recently, NP-hardness was proved for the sum-of-radii problem even for shortest path metrics on weighted planar graphs [24], or, in the case of sum-of-diameters, even for metrics of constant doubling dimension [17]. Turning to approximation algorithms,

―――――――――――

Charikar and Panigraphy [13] design and analyze an $O(1)$ approximation algorithm for sum-of-radii and for sum-of-diameter clustering with k clusters. They start from a linear-programming relaxation, use a primal-dual type approach and, along the way, design a bicriteria algorithm. They also design an incremental algorithm that handles arrivals of clients, merging clusters as needed so that at any time the clustering has $O(k)$ clusters and the cost is $O(1)$ times the optimal cost for k clusters.

There have been many other papers on sum-of-radii or sum-of-diameters clustering. A few papers focus on the problems of partitioning the clients into a constant number of clusters as quickly as possible [20, 24]. Some papers concern themselves with bicriteria results such as [2]. Consider the special case of a metric that is Euclidean in two dimensions. Lev-Tov and Peleg [23] give a polynomial time approximation scheme (PTAS) for the related problem of covering input clients by a min-cost set of disks centered at the servers, where both clients and potential servers are located in the Euclidean plane and are part of the input. Recently, Behsaz and Salavatipour [4] gave a PTAS for the minimum sum of diameters problem on the plane with Euclidean distances. See also [1, 18] for other work on the two-dimensional geometric setting.

The problems are complicated by situations where the set of clients may change over time, for example documents in a very large database that must be efficiently searchable and maintained. This then leads to various models: online [14, 16], incremental[13], streaming, or dynamic. The dynamic setting, where clients may not only arrive but also depart, has been empirically studied at least since 1993 [12], and is the focus of the present paper, with the joint goals of maintaining clusterings whose objective value is close to optimal, and of updating the cluster quickly after each event.

This paper can be interpreted as part of a recent focus on exploiting primal-dual techniques in the dynamic setting. In the online setting (where new elements arrive but never depart), primal-dual techniques are extremely successful [11]. Initially it seemed that such techniques were inherently restricted to settings with arrivals only, and no departures, but there recently has been exciting progress to handle the dynamic setting as well, starting with [8, 7] and continuing with [3, 9, 26, 10]. Of particular notice for us is recent work by Gupta et al [19] for the set cover problem (and Bhattacharya et al [6], restricted to vertex cover) in the dynamic setting where elements arrive and depart.

In this paper, we study the *dynamic sum-of-radii clustering problem*, defined as follows: The original input consists of a (possibly infinite) set V of potential *clients* or *points*, a finite set $F \subseteq V$ of *facilities* with an *opening cost* f_j for each facility j, and a metric d over V. For the online input, a set C of live clients evolves over time: at each timestep t, either a new client arrives and is added to C, or a client from C departs and is removed from C, or a query is made for the approximate cost of an optimal solution (*cost query*), or a query asks for the entire current solution (*solution query*). For the output, at each timestep the algorithm maintains a set of *open* facilities, each open facility j being associated to a radius R_j, such that every client of C is *covered*, i.e. belongs to some open ball $B(j, R_j)$, and the goal is to minimize the *cost*, namely, the sum over open facilities j of $f_j + R_j$.

The dynamic sum-of-radii clustering problem can actually be interpreted as a special case of dynamic set cover: our metric space is the universe, our clients are the elements, and for each center c and radius r, the ball $B(c, r)$ defines a set of cost r consisting of those clients covered by the ball. The dynamic set cover algorithm from [19], specialized to our setting, maintains competitive ratio $O(\log n)$ and has update time $O(f \log n)$, where f is the maximum number of sets containing any element; [19] also gives an $O(f^3)$ approximation in time $O(f^2)$ for set cover, and for the related dynamic k-coverage problem, they give a constant approximation fully dynamic algorithm with $O(f \log n)$ update time.

The *doubling dimension* of a metric space (V, d) is said to be bounded by κ if any ball $B(x, r)$ in (V, d) can be covered by 2^κ balls of radius $r/2$ [22]. For example, D-dimensional Euclidean space has doubling dimension $\Theta(D)$.

In this paper we give an algorithm for constant doubling dimension, that maintains a solution whose cost is within a constant factor of the optimal cost, and with logarithmic update times for arrival or departure events. The algorithm answers cost queries in constant time and solution queries in linear time in the size of the solution, up to a factor of $\log(W/f_{min})$, where W is the diameter of the metric space and f_{min} is the minimum opening cost of any facility (which is strictly positive without loss of generality, since facilities of cost 0 can remain open at all times). The universe is known ahead of time, and only the collection of active clients changes dynamically. Note that for the above mentioned algorithm by [19] f would be $\Theta(2^{2\kappa} \log(W/f_{min}))$.

▶ **Theorem 1.** *There exists an algorithm for the dynamic sum-of-radii clustering problem, when clients and facilities live in a metric space with doubling dimension κ, such that at every timestep the solution has cost at most $\mathrm{O}(2^{2\kappa})$ times the cost of an optimal solution at that time, and such that the update time is $\mathrm{O}(2^{6\kappa} \log(W/f_{min}))$, where W is the diameter of the space and in the current number of clients and f_{min} is the minimum opening cost. A cost query can be answered in constant time, a solution query in time $\mathrm{O}(s \log(W/f_{min}))$, where s is the size of the output.*

The $2^{6\kappa}$ factor in the update time is due to fixed-radius nearest neighbor query that we solve using only the basic data structure already present in the algorithm. However, it can be improved – for example in case of finite metric spaces where a $\mathrm{O}(1)$ lookup table over the space is possible (e.g., graphs with the shortest path distance), the update time reduces to $\mathrm{O}(\log(W/f_{\min}))$ at the cost of additional preprocessing time. More generally, if the metric space allows to answer fixed-radius nearest neighbor queries in time $\mathrm{O}(d)$, then the update time becomes $\mathrm{O}(d \log(W/f_{\min}))$, at the cost of preprocessing time necessary to construct the oracle.

We show the following structural property for the sum-of-radii clustering problem (Theorem 8): there exists a collection Π of pairs $\langle j, r \rangle$ where $j \in F$ and r is a non-negative integer, each with an associated area $A(j, r)$ of V, and an abstract tree $\mathcal{T}$ over Π, with the following properties:

1. $\mathcal{T}$ has height $\mathrm{O}(\log(W/f_{\min}))$ and degree at most $2^{4\kappa}$.
2. The collection $\mathcal{A}$ of areas is a laminar family, its laminar structure is given by $\mathcal{T}$, and for each area, $A(j, r) \subseteq B(j, 7 \cdot 5^r)$
3. For any subset $\mathcal{C}$ of V, there exists a collection $\mathcal{S}$ of areas covering $\mathcal{C}$ and whose cost, $\sum_{\langle j, r \rangle \in \mathcal{S}} f_j + 7 \cdot 5^r$, is $O(2^{2\kappa})$ times the optimal cost for $\mathcal{C}$.

Our algorithm has two phases. First, in the preprocessing phase (Section 2), the algorithm constructs Π, the laminar family of areas $\mathcal{A}$ and corresponding abstract tree $\mathcal{T}$. Thanks to the last property above, it suffices to restrict attention to solutions that use only areas $A(j, r)$ for coverage, with $\langle j, r \rangle \in \Pi$. Second, in the dynamic phase (Section 3), while clients arrive and depart, the algorithm maintains an optimal set of pairs $\langle j, r \rangle$ of Π such that the corresponding areas $A(j, r)$ cover all current clients. The hierarchical structure of $\mathcal{T}$ makes this simple, so that each update takes time proportional to the height times $2^{6\kappa}$.

The main contribution of the paper is the definition of Π and the corresponding laminar family of areas $\mathcal{A}$. The latter is reminiscent of the cover tree data structure of [5]. However, the cover tree is tailored to the nearest neighbor problem and its covers, to the best of our knowledge, lack the structural properties of areas that we need to prove the approximation

factor for the sum-of-radii clustering problem. We expect that our new structure can be used for other clustering-type problems.

2   Preprocessing phase

2.1   Discretization of radii

Given the set of clients $\mathcal{C} \subseteq V$, let OPT denote the cost of an optimum solution for $\mathcal{C}$.

▶ **Lemma 2.** *For all $\mathcal{C} \subseteq V$, there exists a solution such that every ball $B(j, R)$ has $f_{\min} \leq f_j \leq R \leq 5 \cdot W$, the radius R is an integer power of 5, and the cost is $O(OPT)$.*

Proof. Consider the unknown optimal solution. If some ball is such that $\max(f_j, R) > W$ then replace the entire solution by a ball centered at the facility of cost $f_{\min}$ and of radius W. Else, for each ball $B(j, R)$ of the optimal solution:

- if $f_j > R$ then increase the radius of the ball from R to f_j
- Increase R to the smallest integer power of 5 that is greater than or equal to R.

The new solution satisfies the desired constraints, and the cost has increased by a factor of 10 at most. ◀

A *logradius* is an integer r such that $f_{\min} \leq 5^r \leq 5 \cdot W$. Let $\rho_{\min} = \lfloor \log_5 f_{\min} \rfloor$ and $\rho_{\max} = \lceil \log_5 W \rceil$. Then the number of different logradii, $\rho_{\max} - \rho_{\min} + 1$, is $O(\log(W/f_{\min}))$.

2.2   Maximal subsets of distant facilities

We construct a set Π of *pairs* $\langle j, r \rangle$ where j is a facility and r is a logradius, satisfying the following properties:

1. (Covering) For every facility $j \in J$ and every logradius r such that $f_j \leq 5^r$, there exists a facility $j' \in J_r$ with $d(j, j') \leq 5^{r+1}$ and $\langle j', r \rangle \in \Pi$.
2. (Separating) For all distinct $\langle j', r \rangle, \langle j', r \rangle \in \Pi$, we have $d(j, j') > 5^{r+1}$.

For each logradius $r \in [\rho_{\min}, \rho_{\max}]$:
- let $J'_r = \{j \in F \mid f_j \leq 5^r\}$.
- let J_r be a maximal subset of J'_r such that any two facilities in J_r are at distance greater than 5^{r+1}.

$\Pi \leftarrow \bigcup_r \{\langle j, r \rangle \mid j \in J_r\}$.

Note that for $r = \rho_{\max}$, the set J_r contains just one facility.

2.3   Hierarchical decomposition of Π

Construct an abstract tree $\mathcal{T}$ over Π as follows (with ties broken arbitrarily):
- the root of $\mathcal{T}$ is the unique pair $\langle j, \rho_{\max} \rangle$.
- for all $r < \rho_{\max}$ and $j \in J_r$:
  - let j' be the facility of J_{r+1} closest to j
  - $\text{parent}(j, r) \leftarrow \langle j', r + 1 \rangle$

By construction, $\mathcal{T}$ has height at most $\rho_{\max} - \rho_{\min} + 1$ and the parent of a pair $\langle j, r \rangle$ is a pair of the form $\langle j', r + 1 \rangle$.

The following Lemma is simple, but it captures the essential way in which using larger balls will greatly simplify the structure, and is the main step towards constructing a laminar set of areas for covering clients.

▶ **Lemma 3.** *(Nesting of balls) If* $\text{parent}(j, r) = \langle j', r+1 \rangle$, *then* $B(j, 7 \cdot 5^r) \subseteq B(j', 7 \cdot 5^{r+1})$.

Proof. We have $\langle j, r \rangle \in \Pi$, so $j \in J'_r \subseteq J'_{r+1}$. By the Covering property of Π the maximum distance from any point in $B(j, 7 \cdot 5^r)$ to j' is $d(j, j') + 7 \cdot 5^r \leq 5^{r+2} + 7 \cdot 5^r \leq 7 \cdot 5^{r+1}$. ◄

▶ **Lemma 4.** *For any point p and radius r, the set of pairs*

$$\Pi(p, r) = \{\langle j, r \rangle \in \Pi \mid d(p, j) < 2^\alpha 5^{r+1}\}$$

has at most $2^{(\alpha+1)\kappa}$ elements, where κ is the doubling dimension of the metric space.

Proof. By definition of doubling dimension, $B(p, 2^\alpha \cdot 5^{r+1})$ can be covered by a set of at most $(2^\kappa)^{\alpha+1}$ balls of radius $(1/2) \cdot 5^{r+1}$. By the Separating property of Π, any two pairs $\langle j, r \rangle$ of $\Pi(p, r)$ are at distance greater than 5^{r+1} from each other, hence must belong to different balls of the set, and so $\Pi(p, r)$ has cardinality at most $(2^\kappa)^{\alpha+1}$. ◄

▶ **Lemma 5.** *A node $\langle j, r \rangle$ of $\mathcal{T}$ has at most $2^{4\kappa}$ children*

Proof. Children of $\langle j, r \rangle$ have logradius $r - 1$, so by the Covering property of Π their distance to j is at most 5^{r+1}, so they belong to $\Pi(j, r-1)$ for $\alpha = 3$, and so Lemma 4 applies. ◄

2.4 Hierarchical decomposition of V into a laminar family of areas

Recall that a collection $\mathcal{A}$ of sets is *laminar* if for any two $A, B \in \mathcal{A}$, either $A \cap B = \emptyset$ or $A \subseteq B$ or $B \subseteq A$. We partition V into a laminar family of *areas*, denoted by $\mathcal{A}$, such that no two same-logradius areas overlap.

For each $\langle j, r \rangle \in \Pi$, initialize $A(j, r) \leftarrow \emptyset$.
For each point $p \in V$:
- let r^* be minimum such that there exists pairs $\langle j, r^* \rangle$ with $p \in B(j, 7 \cdot 5^{r^*})$.
- Among all such pairs, let $\langle j^*, r^* \rangle$ denote the one minimizing $d(p, j^*)$.
- Add p to the set $A(j^*, r^*)$ and to every set $A(j', r')$ with (j', r') ancestor of (j^*, r^*) in $\mathcal{T}$.

▶ **Lemma 6.** *For every $\langle j, r \rangle \in \Pi$, $A(j, r) \subseteq B(j, 7 \cdot 5^r)$.*

Proof. Let $p \in A(j, r)$. Either it's been added directly, in which case it belongs to $B(j, 7 \cdot 5^r)$, or it's been inherited, in which case it also belongs to it by Lemma 3. ◄

▶ **Lemma 7.** *For every subset $\mathcal{C} \subseteq V$ of clients there exists $S \subseteq \Pi$ such that $\mathcal{C}$ is covered by $\cup\{A(j, r) : \langle j, r \rangle \in S\}$ and $\sum_{\langle j, r \rangle \in S}(f_j + 7 \cdot 5^r) = \mathrm{O}(2^{2\kappa} \cdot OPT)$.*

Proof. Let S^* be a solution of cost $\mathrm{O}(OPT)$ satisfying the properties of Lemma 2. For each ball $B(j, 5^r)$ of S^*, put in S all the pairs $\langle j', r \rangle \in \Pi$ such that $d(j, j') \leq 8 \cdot 5^r$.

We claim that $\mathcal{C}$ is covered by $\cup\{A(j', r) : \langle j', r \rangle \in S\}$. Indeed, consider a client $p \in \mathcal{C}$ and a ball $B(j, 5^r)$ of S^* containing p. By the Covering property of Π, there exists $\langle j', r \rangle \in \Pi$ with $d(j, j') \leq 5^{r+1}$. Then $d(p, j') \leq 5^{r+1} + 5^r < 7 \cdot 5^r$, and so in the definition of areas covering p we must have $r^* \leq r$. Along the path from $\langle j^*, r^* \rangle$ to the root of $\mathcal{T}$, there exists a pair for logradius r, $\langle j'', r \rangle$. By definition of areas and by Lemma 6, $p \in A(j'', r) \subseteq B(j'', 7 \cdot 5^r)$, so $d(j, j'') \leq d(j, p) + d(p, j'') \leq 8 \cdot 5^r$, and therefore $\langle j'', r \rangle \in S$ and p is covered.

In terms of costs, since all these areas are associated to pairs within distance $8 \cdot 5^r < 2 \cdot 5^{r+1}$ from j, by Lemma 4 for $\alpha = 1$, there are at most $2^{2\kappa}$ of them. ◄

By Lemma 6 and the definition of areas, we note that $\cup_{J_r} A(j,r) = \cup_{J_r} B(j, 7 \cdot 5^r)$, so we also give hereafter an equivalent description of the same laminar family, illustrating the way in which the parent-child relations in tree $\mathcal{T}$ and the proximity relations in the metric space are balanced against one another. (This also has the advantage of being constructive even if V is infinite).

Partition $\cup_{J_{\rho_{\min}}} B(j, 7 \cdot 5^{\rho_{\min}})$, using the facilities of $J_{\rho_{\min}}$ as centers, into Voronoi cells $A(j, \rho_{\min})$.

For $r \in (\rho_{\min}, \rho_{\max}]$:

- Partition $\cup_{j \in J_r} B(j, 7 \cdot 5^r) \setminus \cup_{j \in J_{r-1}} B(j, 7 \cdot 5^{r-1})$, using the facilities of J_r as centers, into Voronoi cells $A(j, r)$.
- For each $\langle j, r \rangle \in \Pi$, $A(j, r) \leftarrow A(j, r) \cup \bigcup \{A(j', r-1) : \mathrm{parent}(j', r-1) = \langle j, r \rangle\}$

The construction of this section can be summarized in the following structural Theorem.

▶ **Theorem 8.** *Let a metric space (V, d) of doubling dimension κ be given, as well as a subset F of elements of V called facilities, with an associated cost f_j for each $j \in F$. Then there exists an abstract tree $\mathcal{T}$ whose nodes are indexed by facilities $j \in F$ and non-negative integers $r \geq 0$, and, for each node $\langle j, r \rangle$, an associated area $A(j, r) \subseteq V$ with the following properties*

1. *$\mathcal{T}$ has height $\mathrm{O}(\log(W/f_{\min}))$ and degree at most $2^{4\kappa}$, and for each $\langle j, r \rangle \in \mathcal{T}$ and its parent node $\langle j', r+1 \rangle$, $B(j, 7 \cdot 5^r) \subseteq B(j', 7 \cdot 5^{r+1})$.*
2. *$\mathcal{A}$ is a laminar family, its laminar structure is given by $\mathcal{T}$, and for each area $A(j, r)$, $A(j, r) \subseteq B(j, 7 \cdot 5^r)$*
3. *For any subset $\mathcal{C}$ of V, for any collection of balls $\mathcal{B}$ centered at facilities of F and covering $\mathcal{C}$, there exists a collection $\mathcal{S}$ of areas covering $\mathcal{C}$, such that $\sum_{\langle j, r \rangle \in \mathcal{S}} f_j + 7 \cdot 5^r = O(2^{2\kappa}) \sum_{B(j,R) \in \mathcal{B}} (f_j + R)$.*

3 Data structure

3.1 Solving the offline restricted problem

Given $\mathcal{C} \subseteq V$, we wish to compute the solution of minimum cost among all *solutions that are restricted to covering $\mathcal{C}$ using areas $A(j, r)$ for $\langle j, r \rangle \in \Pi$, where using area $A(j, r)$ has cost $f_j + c_2 \cdot 5^r$.* We call that the *restricted* problem. By Theorem 8 the optimal restricted cost is a $O(2^{2\kappa})$ approximation of the optimal (unrestricted) cost.

Computing the optimal solution to the restricted problem in an offline manner is straightforward, thanks to the laminar structure of the candidate areas. We first compute, for each node $\langle j, r \rangle$ of $\mathcal{T}$, the cost $c_{j,r} = f_j + c_2 \cdot 5^r$ of area $A(j, r)$, as well as the number $n_{j,r}$ of clients that are in area $A(j, r)$ but not in any of the areas of children nodes: since areas $A(j', r-1)$ are all disjoint by laminarity, we have $n_{j,r} = |\mathcal{C} \cap A(j, r)| - \cup_{\langle j', r-1 \rangle : \mathrm{parent}(j', r-1) = j} |\mathcal{C} \cap A(j', r-1)|$. We then compute the optimal cost $x_{j,r}$ of covering the clients of $\mathcal{C} \cap A(j, r)$ using only areas of the subtree of $\mathcal{T}$ rooted at $\langle j, r \rangle$, using the following bottom-up recurrence:

For $\langle j, r \rangle \in \Pi$ in bottom-up order in $\mathcal{T}$:

$$x_{j,r} = \begin{cases} c_{j,r} \text{ if } n_{j,r} > 0 \\ \min \left(c_{j,r}, \sum \{x_{j',r-1} : \langle j, r \rangle = \mathrm{parent}(j', r-1)\} \right) \text{ otherwise.} \end{cases}$$

Indeed, if $n_{j,r} \neq 0$ then the solution must use area $A(j, r)$; but then by laminarity area $A(j, r)$ covers all clients in that subtree, so no other area is needed in the solution, and

the cost is exactly the cost $c_{j,r}$ of $A(j,r)$. If on the other hand $n_{j,r} = 0$, then we have an alternative possibility: we could do without $A(j,r)$. Then, by disjointness of sibling areas the problem separates into independent subproblems, one for each child of $\langle j, r \rangle$, hence the recurrence simply sums their costs.

The cost of the optimal restricted solution is then $x_{j,\rho_{\max}}$ for the root $\langle j, \rho_{\max} \rangle$ of $\mathcal{T}$.

Given $c_{j,r}$ and $x_{j,r}$, computing the optimal restricted solution, a collection S of areas, is done recursively:

$$
S(j,r) = \begin{cases} \emptyset \text{ if } x_{j,r} = 0 \\ \{A(j,r)\} \text{ if } x_{j,r} = c_{j,r} \\ \cup \{S(j', r-1) : \text{parent}(j', r-1) = \langle j, r \rangle\} \text{ otherwise.} \end{cases}
$$

Thus the algorithm to compute the optimal set S of areas covering $\mathcal{C}$ in the restricted problem, given the values of $c_{j,r}, x_{j,r}$ explores a tree $\mathcal{T}'$ that, as it is a partial subtree of $\mathcal{T}$, also has height at most $O(\log(W/f_{\min}))$ and degree at most $2^{4\kappa}$; moreover its internal nodes are all ancestors of areas added to the solution S, so the running time to compute S itself is $O(2^{4\kappa} \log(W/f_{\min})|S|)$.

3.2   The dynamic data structure

The dynamic data structure supports insertions of clients, deletions of clients, queries for the cost of the optimal restricted solution, and queries for the set of open facilities and areas of the optimal restricted solution.

The algorithm will maintain two dynamic data structures:

1. a list of the currently existing clients $\mathcal{C} \subseteq V$, with, for each client p, the $\langle j, r \rangle \in \Pi$ such that $p \in A(j,r)$ and r is minimum; and
2. an *annotated dependency tree* $\mathcal{T}_A$, keeping for each node $v = \langle j, r \rangle$ the following additional information:

   a. its cost $c_v = f_j + 7 \cdot 5^r$,
   b. the number n_v of currently existing clients that belong to $A(j,r)$ but not to any descendant area,
   c. the value x_v, which is the minimum cost needed to cover all clients belonging to $A(j,r)$ using only areas $A(j', r')$ for $\langle j', r' \rangle \in \Pi$, and
   d. the value $y_v = \sum_{u \text{ child of } v} x_u$.

To initialize the data structures, from the preprocessing phase the algorithm is given the set Π of pairs $\langle j, r \rangle$, as well as the laminar family of areas $\mathcal{A}$ with its dependency tree $\mathcal{T}$ using the following representation, which can be easily computed in time linear in its size: (1) An array of size $\rho_{\max} - \rho_{\min} + 1$, keeping for each logradius $r \in [\rho_{\min}, \rho_{\max}]$ a list of all the facilities of J_r, and (2) An annotated tree data structure obtained from $\mathcal{T}$ by setting every n_v, x_v, y_v equal to 0, and $c_{j,r} = f_j + 7 \cdot 5^r$. The initial set of clients is $\mathcal{C} = \emptyset$.

Answering queries is done as in Section 3.1.

We next describe the client deletions. When a client p is deleted, we start from $\langle j, r \rangle$ in $\mathcal{T}_A$, such that $p \in A(j,r)$ and r is minimum; we decrement n_v and we traverse the path from $\langle j, r \rangle$ up to the root of $\mathcal{T}_A$, updating x_v and $y_{\text{parent}(v)}$ for every node visited along the way using the recurrence from Section 3.1. This takes time proportional to the height of the tree, $O(\log(W/f_{\min}))$.

Similarly, when a client p is inserted, we first find $\langle j, r \rangle$ in $\mathcal{T}_A$, such that $p \in A(j, r)$ and r is minimum, in a way to be described shortly; we increment n_v, and then we traverse the path from $\langle j, r \rangle$ up to the root of $\mathcal{T}_A$, similarly updating x_v and $y_{\text{parent}(v)}$.

Thus, it only remains to determine the pair $\langle j^*, r^* \rangle$ with smallest lograduis such that $p \in A(j^*, r^*)$. By Lemma 6, $p \in B(j^*, 7 \cdot 5^{r^*})$. Thus we will first find all pairs $\langle j, r \rangle$ such that $p \in B(j, 7 \cdot 5^r)$, based on them determine r^*, and then look for $\langle j^*, r^* \rangle$ in that set of balls. Thanks to Lemma 3, the first part can be done using a simple recursive algorithm starting from the root of $\mathcal{T}_A$ (see below). The second part simply uses the definition of areas, i.e., it finds the pair $\langle j^*, r^* \rangle$ where j^* has with minimum distance to p out of all pairs $\langle j, r^* \rangle$ with $p \in B(j, 7 \cdot 5^{r^*})$.

$$
\text{Pairs}(p, j, r) = \begin{cases} \emptyset \text{ if } p \notin B(j, 7 \cdot 5^r) \\ \{\langle j, r \rangle\} \cup \bigcup \{\text{Pairs}(p, j', r - 1) : \text{parent}(j', r - 1) = \langle j, r \rangle\} \text{ otherwise.} \end{cases}
$$

- let r^* be minimum such that there exists pairs $\langle j, r^* \rangle$ in the set $\text{Pairs}(p, j_{\text{root}}, \rho_{\text{max}})$.
- Among all such pairs, output the pair $\langle j^*, r^* \rangle$ minimizing $d(p, j^*)$.

The running time is dominated by the first part, which is $O(2^{4\kappa})$ times the number of pairs $\langle j, r \rangle$ such that $p \in B(j, 7 \cdot 5^r)$. There are $\log(W/f_{\text{min}})$ possible values of r. For each r, by Lemma 4 there are at most $2^{2\kappa}$ pairs $\langle j, r \rangle \in \Pi$ such that $p \in B(j, c_2 \cdot 5^r)$ and the algorithm has to test the $O(2^{4\kappa})$ children of each of them. Thus the running time to do an insertion is $O(2^{6\kappa} \log(W/f_{min}))$.

References

1 Helmut Alt, Esther M. Arkin, Hervé Brönnimann, Jeff Erickson, Sándor P. Fekete, Christian Knauer, Jonathan Lenchner, Joseph S. B. Mitchell, and Kim Whittlesey. Minimum-cost coverage of point sets by disks. In *Proceedings of the Twenty-second Annual Symposium on Computational Geometry*, SCG'06, pages 449–458, New York, NY, USA, 2006. ACM. `doi:10.1145/1137856.1137922`.

2 Sayan Bandyapadhyay and Kasturi R. Varadarajan. Approximate clustering via metric partitioning. *CoRR*, abs/1507.02222, 2015. URL: `http://arxiv.org/abs/1507.02222`.

3 Surender Baswana, Manoj Gupta, and Sandeep Sen. Fully dynamic maximal matching in o(log n) update time. *SIAM J. Comput.*, 44(1):88–113, 2015. `doi:10.1137/130914140`.

4 Babak Behsaz and Mohammad R. Salavatipour. On minimum sum of radii and diameters clustering. *Algorithmica*, 73(1):143–165, September 2015. `doi:10.1007/s00453-014-9907-3`.

5 Alina Beygelzimer, Sham Kakade, and John Langford. Cover trees for nearest neighbor. In William W. Cohen and Andrew Moore, editors, *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, volume 148 of *ACM International Conference Proceeding Series*, pages 97–104. ACM, 2006. `doi:10.1145/1143844.1143857`.

6 Sayan Bhattacharya, Deeparnab Chakrabarty, and Monika Henzinger. Deterministic fully dynamic approximate vertex cover and fractional matching in $o(1)$ amortized update time. *CoRR*, abs/1611.00198, 2016. URL: `http://arxiv.org/abs/1611.00198`.

7 Sayan Bhattacharya, Monika Henzinger, and Giuseppe F. Italiano. Design of dynamic algorithms via primal-dual method. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, pages 206–218, 2015. `doi:10.1007/978-3-662-47672-7_17`.

8 Sayan Bhattacharya, Monika Henzinger, and Giuseppe F. Italiano. Deterministic fully dynamic data structures for vertex cover and matching. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 785–804, 2015. `doi:10.1137/1.9781611973730.54`.

9 Sayan Bhattacharya, Monika Henzinger, and Danupon Nanongkai. New deterministic approximation algorithms for fully dynamic matching. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 398–411, 2016. `doi:10.1145/2897518.2897568`.

10 Sayan Bhattacharya, Monika Henzinger, and Danupon Nanongkai. Fully dynamic approximate maximum matching and minimum vertex cover in $o(\log^3 n)$ worst case update time. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 470–489, 2017. `doi:10.1137/1.9781611974782.30`.

11 Niv Buchbinder and Joseph (Seffi) Naor. The design of competitive online algorithms via a primal: Dual approach. *Found. Trends Theor. Comput. Sci.*, 3(2–3):93–263, February 2009. `doi:10.1561/0400000024`.

12 Fazli Can. Incremental clustering for dynamic information processing. *ACM Trans. Inf. Syst.*, 11(2):143–164, April 1993. `doi:10.1145/130226.134466`.

13 Moses Charikar and Rina Panigrahy. Clustering to minimize the sum of cluster diameters. *Journal of Computer and System Sciences*, 68(2):417–441, 2004. `doi:10.1016/j.jcss.2003.07.014`.

14 János Csirik, Leah Epstein, Csanád Imreh, and Asaf Levin. Online clustering with variable sized clusters. *Algorithmica*, 65(2):251–274, 2013. `doi:10.1007/s00453-011-9586-2`.

15 Srinivas R. Doddi, Madhav V. Marathe, Sekharipuram S. Ravi, David S. Taylor, and Peter Widmayer. Approximation Algorithms for Clustering to Minimize the Sum of Diameters. *Nordic journal of computing*, 7(3):185–203, 2000.

16 Dimitris Fotakis and Paraschos Koutris. Online sum-radii clustering. *CoRR*, abs/1109.5325, 2011. URL: `http://arxiv.org/abs/1109.5325`.

17 Matt Gibson, Gaurav Kanade, Erik Krohn, Imran A. Pirwani, and Kasturi Varadarajan. On clustering to minimize the sum of radii. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA'08, pages 819–825, Philadelphia, PA, USA, 2008. Society for Industrial and Applied Mathematics. URL: `http://dl.acm.org/citation.cfm?id=1347082.1347172`.

18 Matt Gibson, Gaurav Kanade, Erik Krohn, Imran A. Pirwani, and Kasturi Varadarajan. On metric clustering to minimize the sum of radii. In *Proceedings of the 11th Scandinavian Workshop on Algorithm Theory*, SWAT'08, pages 282–293, Berlin, Heidelberg, 2008. Springer-Verlag. `doi:10.1007/978-3-540-69903-3_26`.

19 Anupam Gupta, Ravishankar Krishnaswamy, Amit Kumar, and Debmalya Panigrahi. Online and dynamic algorithms for set cover. *CoRR*, abs/1611.05646, 2016. URL: `http://arxiv.org/abs/1611.05646`.

20 P. Hansen and B. Jaumard. Minimum sum of diameters clustering. *Journal of Classification*, 4(2):215–226, 1987.

21 Pierre Hansen and Brigitte Jaumard. Cluster analysis and mathematical programming. *Math. Program.*, 79(1-3):191–215, October 1997. `doi:10.1007/BF02614317`.

22 Robert Krauthgamer and James R. Lee. Navigating nets: Simple algorithms for proximity search. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA'04, pages 798–807, Philadelphia, PA, USA, 2004. Society for Industrial and Applied Mathematics. URL: `http://dl.acm.org/citation.cfm?id=982792.982913`.

23   Nissan Lev-Tov and David Peleg. Polynomial time approximation schemes for base station coverage with minimum total radii. *Comput. Netw. ISDN Syst.*, 47(4):489–501, March 2005. `doi:10.1016/j.comnet.2004.08.012`.

24   Guido Proietti and Peter Widmayer. Partitioning the nodes of a graph to minimize the sum of subgraph radii. In *Proceedings of the 17th International Conference on Algorithms and Computation*, ISAAC'06, pages 578–587, Berlin, Heidelberg, 2006. Springer-Verlag. `doi:10.1007/11940128_58`.

25   Satu Elisa Schaeffer. Survey: Graph clustering. *Comput. Sci. Rev.*, 1(1):27–64, August 2007. `doi:10.1016/j.cosrev.2007.05.001`.

26   Shay Solomon. Fully dynamic maximal matching in constant update time. *IEEE FOCS*, 2016. URL: `http://arxiv.org/abs/1604.08491`.