Dynamic Sum-Radii Clustering ⋆

Nicolas K. Blanchard[1] and Nicolas Schabanel[2]

[1] U. Paris Diderot (France), ENS Paris
http://www.irif.univ-paris-diderot.fr/users/nkblanchard
[2] CNRS, U. Paris Diderot (France), IXXI, U.Lyon (France),
http://www.irif.univ-paris-diderot.fr/users/nschaban

Abstract. Real networks have in common that they evolve over time and their dynamics have a huge impact on their structure. Clustering is an efficient tool to reduce the complexity to allow representation of the data. In 2014, Eisenstat *et al.* introduced a dynamic version of this classic problem where the distances evolve with time and where coherence over time is enforced by introducing a cost for clients to change their assigned facility. They designed a $\Theta(\ln n)$-approximation. An $O(1)$-approximation for the *metric* case was proposed later on by An *et al.* (2015). Both articles aimed at minimizing the sum of all client-facility distances; however, other metrics may be more relevant. In this article we aim to minimize the sum of the *radii of the clusters* instead. We obtain an asymptotically optimal $\Theta(\ln n)$-approximation algorithm where n is the number of clients and show that existing algorithms from An *et al.* (2015) do not achieve a constant approximation in the metric variant of this setting.

Keywords: Facility Location, Approximation Algorithms, Clustering, Dynamic Graphs

1 Introduction

Context. During the past decade, a massive amount of data has been collected on diverse networks such as the web (pages and links), social networks (e.g., Facebook, Twitter, and LinkedIn), social encounters in hospitals, schools, companies, conferences as well as in the wild [13, 15, 16]. These networks evolve over time, and their dynamics have a considerable impact on their structure and effectiveness [14]. Understanding the dynamics of evolving networks is a central question in many applied areas such as epidemiology, vaccination planning, antivirus design, management of human resources, and viral marketing. A relevant clustering of the data is often needed to design informative representations of massive data sets. Algorithmic approaches have already yielded useful insights on real networks such as the social interaction networks of zebras [17]. In most experiments, data is recorded first and analyzed next, see [15]. The complete evolution of the network is thus known from the beginning, as opposed to the online

⋆ This work was supported by Grants ANR-12-BS02-005 RDAM and IXXI-Molecal.

setting where one must continuously adapt a partial solution to new incoming data [7] .

Previous work. Given a set of facilities, a set of clients, and a measure of distances between them, the facility location problem consists in opening a subset of facilities and assigning the clients to open facilities so as to minimize a trade-off between the cost of opening the facilities and the cost corresponding to the distance between the clients and their assigned facilities. This problem and its many variants have been extensively studied since the 1960s, using tools such as LP-rounding [11], primal-dual methods [3] or greedy improvements [8]. The uncapacited version, where any number of clients can connect to a facility, is considered here as it is known to be a successful approach to clustering when the number of clusters is not known a priori.

In 2014, [5] introduced a dynamic version of this classic problem to handle situations where the distances evolve with time and where one looks for an assignment consistent with the evolution of the distances. To achieve a balance between the stability of the solution and its adaptability, they introduced a cost to be paid every time a client is assigned to a new facility. As shown in [5], in many natural scenarios the output solutions follow the observed dynamic better than independent optimizations of consecutive snapshots of the evolving distances. This has been further refined in [1], yielding an $O(1)$-approximation algorithm when the distances are metric (i.e., follow the triangular inequalities).

Our approach: Dynamic Sum-Radii Clustering. In both articles [5, 1], the distance cost in the objective consisted of the sum of all distances between every client and its assigned facility over all time steps. Whereas this distance cost makes perfect sense in the case where clients need to physically connect to a facility, other metrics are preferred in the context of clustering. The present article introduces a dynamic version of the problem studied in [3]. We aim at minimizing the *radii of the clusters*, i.e. the sum over all open facilities of their distances to their farthest assigned client at each time step. This objective focuses on the closeness of the clients to their assigned facility regardless of the number of clients assigned to each open facility. It is thus better suited to situations with clusters of very different sizes which are typically observed in nature where groups tend to follow power laws in size [13]. Optimal solutions for this objective cost have been explored in [6], where it was shown that even in the 1-dimensional euclidean space, optimal solutions can have surprisingly complex structures.

In the general setting, we introduce a primal LP-rounding algorithm that achieves a logarithmic approximation, which is shown to be asymptotically optimal unless $P = NP$. We then turn to metric distances and show that existing algorithms from [1] do not achieve a constant approximation in this setting, as the lack of cooperation between the clients is not being absorbed by the sum-of-radii objective anymore. The next section presents a formal definition of the problem and states our main results, proved in the following sections.

2 Definition and Main Results

2.1 Definitions

Dynamic Sum-Radii Clustering (DSRC). Given a set F of m facilities, a set C of n clients, their respective distances $(d_{ijt})_{i\in F, j\in C, t\in[T]}$ for each time step $t \in [T] = \{1, \dots, T\}$, an opening cost $f_{it} \geqslant 0$ for each facility i at time t, and a changing cost $g \geqslant 0$, the goal is to open at each time step t a subset $O_t \subseteq F$ of facilities and to assign each client $j \in C$ to an open facility from O_t so as to minimize the sum of:

Opening cost: $\sum_{t\in[T]} \sum_{i\in O_t} (f_{it} + r_{it})$, where for each facility $i \in O_t$, $r_{it} = \max\{d_{ijt} : j \in C \text{ s.t. } j \text{ is assigned to i at time t}\}$ denotes its open radius.

Changing cost: $\sum_{t\in[T-1]} \sum_{j\in C} g \cdot \mathbb{1}_{\{\varphi_{jt} \neq \varphi_{j(t+1)}\}}$.

Precisely, this problem is strictly equivalent to the linear program (1), inspired by [3, 5], when its variables $x_{ijt}, y_{irt}, z_{ijt}$ are restricted to integral values in $\{0, 1\}$. Their integral values are interpreted as follows: $x_{ijt} = 1$ iff Client j is assigned to Facility i at time t (Constraint (1.a)); $y_{irt} = 1$ iff Facility i is open with radius r at time t (Constraint (1.b)); $z_{ijt} = 1$ iff Client j is assigned to Facility i at time $t+1$ and was not assigned to i at t (Constraint (1.c)). Note that one can restrict the total number of y_{irt} variables to mnT as one shall only consider the radii r equal to some distance d_{ijt} for some $j \in C$, for each facility i and time t.

$$\left.\begin{aligned} \text{Minimize} \quad & \sum_{i\in F, r\geqslant 0, t\in[T]} y_{irt} \cdot (f_{it} + r) \quad + \quad g \cdot \sum_{i\in F, j\in C, t\in[T-1]} z_{ijt} \\ \text{such that } (1.a) \quad & \sum_{i\in F} x_{ijt} \geqslant 1 \qquad\qquad (\forall j \in C, t \in [T]) \\ (1.b) \quad & \sum_{r\,:\, r\geqslant d_{ijt}} y_{irt} \geqslant x_{ijt} \qquad\qquad (\forall i \in F, j \in C, t \in [T]) \\ (1.c) \quad & z_{ijt} \geqslant x_{ij(t+1)} - x_{ijt} \quad (\forall i \in F, j \in C, t \in [T-1]) \\ & \text{and } x_{ijt}, y_{ijrt}, z_{ijt} \geqslant 0. \end{aligned}\right\} \quad (1)$$

We denote by LP the optimum (fractional) value of (1), and for each time period $U \subseteq [T]$, by $\text{openCost}_U(x, y, z) = \sum_{i\in F, r\geqslant 0, t\in U} y_{irt} \cdot (f_{it} + r)$ the fractional opening cost of solution (x, y, z) during the time period U, and by $\text{changeCost}_U(x, y, z) = g \cdot \sum_{i\in F, j\in C, t\in U\setminus\{\max U\}} z_{ijt}$ the fractional changing cost of (x, y, z) during U. The index U is omitted when $U = [T]$.

2.2 Preprocessing

As in [5], our algorithm first preprocesses an optimal solution to this LP in order to obtain some useful properties. This preprocessing, detailed in the appendix, uses a rounding scheme for the z_{ijt} to determine at which discrete time the clients must change their assigned facility. This is achieved by the following Lemma.

Lemma 1 (Direct adaptation from [5]). *Given an optimal solution to LP* (1)*, one can compute a feasible solution* (x, y, z) *together with a collection of time intervals* $I_{1,1}, \ldots, I_{1,\ell_1}, \ldots, I_{n,1}, \ldots, I_{n,\ell_n}$ *such that:*

- *for all* $j \in C$*:* $I_{j,1}, \ldots, I_{j,\ell_j}$ *form a partition of* $[T]$*; and*
- *for all* $i \in F$*,* $j \in C$ *and* $k \in [\ell_j]$*:* x_{ijt} *is constant during each time interval* I_{jk}*; and*
- *for all* $j \in C$*:* $\ell_j - 1 \leq 2 \sum_{i \in F, t \in T} z_{ijt}$*; and*
- *the new solution costs at most twice as much as the original.*

Moreover, one can assume that for all i*,* j *and* t*:* $x_{ijt} \leqslant 1$ *and* $\sum_r y_{irt} \leqslant 1$*.*

2.3 Our main results

Let us first recall that thanks to a standard reduction from the Set Cover problem (folklore) to the (static) Facility Location problem, the Dynamic Sum-Radii Clustering problem has no $(1 - o(1)) \ln n$-approximation unless $P = NP$ (see Proposition 1 in appendix).

We then present three algorithms for the DSRC problem. Algorithms 1 and 2 (Sections 3.1 and 3.2) allow us to obtain a randomized approximation with optimal approximation ratio $\Theta(\ln n)$ for the general (non-metric) case:

Theorem 1 (Algorithm). *With probability at least* $1/4$*, Algorithm 2 (page 2) outputs in polynomial expected time a valid solution to the DSRC problem, with cost at most* $8 \ln(4n) \cdot \mathrm{OPT}$*.*

Note that the success probability and the approximation ratio can be improved by independent executions of the algorithm. The techniques in Section 3.2 also apply to the algorithm in [5] in the hourly non-metric setting, improving its approximation ratio from $\Theta(\log nT)$ to $\Theta(\log n)$. We then turn to the metric case and propose a candidate approximation algorithm based on the work [1], but show, by exhibiting a hard metric instance family, that its approximation ratio is no better than $\Omega(\ln \ln n)$ for the sum-of-radii objective.

Theorem 2 (Hard metric instance). *There is a metric instance family for which the Sum-of-radii ANS algorithm (algorithm 3, page 8) outputs solutions with cost* $\Omega(\log \log n)\,\mathrm{OPT}$ *w.h.p.*

3 Tight approximation algorithm for the general case

3.1 $O(\log(nT))$-approximation

As in [5], the first step consists in preprocessing an optimal solution to the LP in order to determine when clients should change the facility they're assigned to. Lemma 1 allows us to focus only on the opening cost within each time interval I_{jk} independently for each client j. Indeed, if one can assign a unique facility φ_{jk} to client j during each interval I_{jk}, then the changing cost for j is at most the

number of intervals minus one times g. As Lemma 1 ensures that for all $j \in C$: $\ell_j - 1 \leq 2 \sum_{i \in F, t \in T} z_{ijt}$, the resulting changing cost is at most twice the amount paid by the optimal solution in the original LP. It is worth noting, though, that the intervals are not the same for each client and are not synchronized. The dynamic dimension of the problem is hence simplified but not eliminated.

From now on, we can assume that the clients don't change facilities inside each of their intervals (which is verified by our algorithms). Hence, we shall focus on deciding which facilities to open, when, and with which radius, and how to assign each client to one of them during each of their time intervals. Algorithm 1 does that by combining $\log nT$ partial solutions, each of expected cost LP and obtained by opening a set of random facilities according the y_{irt}.

Algorithm 1: $O(\log nT)$-approximation

Preprocess an optimal solution to LP (1) to obtain a feasible solution (x, y, z) as in Lemma 1.
Let $Z = \ell_1 + \cdots + \ell_n$ be the total number of time intervals I_{jk} associated to (x, y, z) by Lemma 1.
Set $r_{it} := -\infty$ for all $i \in F$ and $t \in [T]$.
repeat $\ln(2Z)$ *times*
  for *each facility* i **do**
    Draw a random variable Y_i uniformly and independently in $[0, 1]$.
    for *every time* t **do**
      Let $\rho_{it} := \max\{\rho : \sum_{r \geqslant \rho} y_{irt} \geqslant Y_i\}$ ($\rho_{it} = -\infty$ if the set is empty)
      Set $r_{it} := \max(r_{it}, \rho_{it})$ and open Facility i with radius r_{it} at time t if $r_{it} \geqslant 0$.
  for *each client* j *and time interval* I_{jk} *during which* j *is not yet covered* **do**
    Connect j to any open facility i (if there is one) that covers j during the whole time interval I_{jk} (i.e., s.t. $d_{ijt} \leqslant r_{it}$ for all $t \in I_{jk}$).

We first analyse the cost of the algorithm, then prove that the solution is indeed correct.

Lemma 2. *The expected increase in total opening cost at each iteration of the repeat loop is at most* $\sum_{irt} y_{irt}(f_{it} + r)$.

Proof. The probability that Facility i is open with radius r at each iteration of the repeat loop is: $\Pr\{\rho_{it} = r\} = \Pr\{Y_i \leqslant \sum_{\rho \geqslant r} y_{i\rho t} \text{ and } Y_i > \sum_{\rho > r} y_{i\rho t}\} = \Pr\{Y_i \in (\gamma, \gamma + y_{irt}]\} = y_{irt}$ where $\gamma = \sum_{\rho > r} y_{i\rho t}$ and recalling that $\gamma + y_{irt} \leqslant 1$ by Lemma 1. It follows that the expected opening cost for Facility i at time t is precisely $\sum_r y_{irt}(f_{it} + r)$. As the radius of each facility i increases by at most ρ_{it} at each iteration of the repeat loop, the expected total added opening cost of each loop is thus at most: $\sum_{it} \sum_r y_{irt}(f_{it} + r)$.

Lemma 3. *For each client j and each time interval I_{jk}, at the end of each iteration of the repeat loop, the probability that j is not covered during I_{jk} is at most $1/e$.*

Proof. Fix a client j and a time t. Client j is covered if there is an open facility i with radius at least d_{ijt}, i.e. s.t. $Y_i \leqslant \sum_{r \geqslant d_{ijt}} y_{irt}$. As $x_{ijt} \leqslant \sum_{r \geqslant d_{ijt}} y_{irt}$ by constraint (1.b), j is thus covered by i as soon as $Y_i \leqslant x_{ijt}$ which happens with probability x_{ijt}. As the Y_is are independent, j is not covered by any facility at time t with probability at most $\prod_i (1 - x_{ijt}) \leqslant \left(1 - \sum_i x_{ijt}/m\right)^m \leqslant (1 - 1/m)^m \leqslant 1/e$ by concavity of the logarithm and constraint (1.a). Since the x_{ijt}s are constant for $t \in I_{jk}$, this also bounds from above the probability that j is not covered during the whole time interval I_{jk}.

Theorem 3. *With probability $1/4$, Algorithm 1 outputs a valid assignment of clients to open facilities with cost at most:*

$$8\ln(2Z) \cdot \mathrm{LP} \leqslant 8\ln(2Z) \cdot \mathrm{OPT} \leqslant 8\ln(2nT) \cdot \mathrm{OPT}.$$

Proof. As the iterations of the repeat loops are independent, each client j has a probability at most $1/e^{\ln(2Z)} = 1/2Z$ of not being covered during each interval I_{jk}. The union bound taken over all intervals I_{jk} ensures that the probability that some client is not covered at some time t by an open facility is at most $Z/2Z = 1/2$ at the end of the algorithm. Let A be the event that all clients are covered at all time steps by the assignment φ computed by Algorithm 1, and $\bar{A}$ its complementary event. Then, the $\mathbb{E}[\mathrm{cost}(\varphi)|A] = (\mathbb{E}[\mathrm{cost}(\varphi)] - \mathbb{E}[\mathrm{cost}(\varphi)|\bar{A}] \Pr \bar{A}) / \Pr A \leqslant \mathbb{E}[\mathrm{cost}(\varphi)] / \Pr A \leqslant 2 \cdot \ln(2Z) \cdot 2\,\mathrm{LP}$ by the previous lemmas. By Markov's inequality, we conclude that with probability at least $1/4$, Algorithm 1 produces a valid assignment of the clients to open facilities with total cost at most $2 \cdot 4\ln(2Z)\,\mathrm{LP} \leqslant 8\ln(2nT)\,\mathrm{OPT}$, since $Z \leqslant nT$ obviously.

3.2 $O(\log n)$-approximation

Concatenating two partial assignments around time t does not change the opening cost of each partial assignment and increases the changing cost by at most $g \cdot n$. We can greedily split the instance into several time periods, making sure that at least n and no more than $2n$ intervals I_{jk} end in each time period (except for the last). Doing so, the cost of stitching together two consecutive partial assignments is at most $n \times g$, hence no higher than the changing cost already paid within each part. By running Algorithm 1 on each partial solution corresponding to a time period and stitching the different solutions, we at most double the changing cost, increasing the bound to $4\,\mathrm{changeCost}(x, y, z)$. On each time period with T' intervals, the opening cost is at most $8\ln(2T')\,\mathrm{openCost}(x, y, z)$, with $T' \leq 2n$. This implies that the overall approximation ratio is $8\ln(4n)$ for Algorithm 2 on the facing page, proving Theorem 1.

Note that this technique also applies to the algorithm in [5], improving the approximation ratio in the non-metric hourly sum-of-distances setting from $O(\log nT)$ to $O(\log n)$.

Algorithm 2: Batch $O(\log n)$-approximation

Preprocess an optimal solution to LP (1) to obtain a feasible solution (x, y, z) as in Algorithm 1.
if $Z \leqslant 2n$ **then**
  Run Algorithm 1
else
  Partition time greedily into Q periods $U_q = [t_q, t_{q+1})$ where Q and $(t_q)_{q\in[Q+1]}$ are defined as follows: $t_1 = 1$, and t_q is defined inductively as the largest $t \leqslant T$ such that at most n intervals I_{jk} end between t_{q-1} and $t-1$. Set $t_{Q+1} = T+1$.
  for $q = 1..Q$ **do**
    Run several times Algorithm 1 with (x, y, z) on the instance restricted to time period U_q until it outputs a valid solution with opening cost at most $8\ln(4n)\,\mathrm{openCost}_{U_q}(x, y, z)$.
  Output the concatenation of the computed assignments in each time period U_q.

Proof (Proof of Theorem 1). Assume $Z > 2n$. As the instance restricted to interval U_q contains $Z_q \leqslant 2n$ overlapping intervals I_{jk}, Algorithm 1 outputs a solution for this restriction with opening cost at most $8\ln(4n)\,\mathrm{openCost}_{U_q}(x, y, z)$ with probability at least $1/4$. It follows that Algorithm 1 is run at most four times on expectation for each q, hence the polynomial expected time. The changing cost paid for the concatenating of the solutions is then at most :

$$g \cdot (Z_1 + \cdots + Z_Q + n(Q-1)) \leqslant g(Z + n\cdot\tfrac{Z}{n}) \leqslant 3g(Z-n) \leqslant 6\,\mathrm{changeCost}(x, y, z)$$

It follows that the solution output by Algorithm 2 costs at most :

$$\begin{aligned} 8\ln(4n)\,\mathrm{openCost}_{U_q}(x, y, z) + 6\,\mathrm{changeCost}(x, y, z) &\leqslant \max(6, 8\ln(4n))\,\mathrm{LP} \\ &\leqslant 8\ln(4n)\,\mathrm{OPT}\,. \end{aligned}$$

4 Lower bounds for the Metric Case

In this section, we focus on the *metric* case, i.e. where the distances d_{xyt} (with $x, y \in F \cup C$) verify the triangle inequalities at all times. Exploiting this additional property, [1] proposed an $O(1)$-approximation (referred to here as the *ANS algorithm*) for the Metric Dynamic Facility Location problem with the *sum-of-distances* objective. For the *sum-of-radii* objective studied here, it is unclear whether an $O(1)$-approximation exists when the distances are metric. Indeed, we were not able to obtain such an $O(1)$-approximation algorithm for metric DSRC. However, we show in this section that the natural adaptation of the ANS algorithm to the sum-of-radii setting cannot achieve any approximation ratio better than $\Omega(\ln\ln n)$ by exhibiting a hard metric instance family. This example demonstrates that the main issue is that clients have to collaborate to make the right choices in order to avoid rare errors that would be absorbed by the sum-of-distances objective but not by the sum-of-radii objective.

Adapting the ANS algorithm. The original ANS algorithm preprocesses the solution of the LP further so that every variable in the LP only takes one positive value besides 0. This is obtained by duplicating each facility at most nT times, so that only one client x_{ijt}-variable contributes to each of the copies of the y_{irt}-variables and for one radius r only.

Lemma 4 ([1]). *Given an optimal solution (x^*, y^*, z^*) to LP (1), one can compute an equivalent instance together with a feasible solution (x', y', z') to the corresponding LP s.t.:*

- *each facility i is replaced in the new instance by a set of (at most nT) virtual facilities located at the same position as i at all times and with opening cost f_{it}; and*
- *(x', y', z') verifies the properties in Lemma 1; and*
- *for each virtual facility i', there is a constant $c_{i'}$ and a client j such that for all time steps t, $x'_{i'jt} \in \{0, c_{i'}\}$, $y'_{i',d_{ijt},t} \in \{0, c_{i'}\}$ and $y'_{i'rt} = 0$ for all $r \neq d_{ijt}$; and*
- *the solution to the original LP is obtained for each facility by summing up the fractional solutions over its virtual copies.*

Algorithm 3: Sum-of-radii ANS algorithm (from [1])

Preprocess an optimal solution to LP (1) according to Lemma 4.
For each virtual facility $i' \in F'$: draw a random variable $Y_{i'}$ according to the exponential distribution of parameter $c_{i'}$ independently.
For each client j: draw a uniform random variable X_j from $[0, 1]$ independently.
for *each time step $t \in [T]$* **do**
  Starting from an empty bipartite Clients-Facilities graph G_t:
  • add an arc from each client j to the facility i' with minimal $Y_{i'}$ among those with $x_{i'jt} > 0$;
  • add an arc from each facility i' to the client j with smallest X_j among those with $x_{i'jt} > 0$.
  Open at time t every facility whose virtual copy belongs to a circuit in G_t with the corresponding radius, and assign each client j to the open facility at the end of the directed path originating from j enlarging its radius accordlingly.

Algorithm 3 presents the transcription of the ANS algorithm to the sum-of-radii objective. The only difference lies in using LP (1) instead of the linear program with the sum-of-distance objective in [1].

4.1 A hard instance family

The key to the performance of the ANS algorithm for the sum-of-distances objective in [1] is that the Y_is and X_js drop exponentially when one follows the

directed path originating from a client, which ensures that just enough facilities are open, and that all the clients are a constant factor away from their ideal facility on expectation. Deviations from the expectation are absorbed by the summation in the objective. In the following, we will exhibit a metric instance showing that the adaptation to the sum-of-radii objective (Algorithm 3) cannot obtain an approximation ratio better than $\Omega(\ln \ln n)$.

The static arborescent instance T_h. Consider for now the *static* (one time-step) instance T_h where the metric distance is defined by the L_∞ norm over $\mathbb{R}^h$, where the facilities are positioned at $(\pm 2^0, \pm 2^{-1}, \ldots, \pm 2^{-k+1}, 0, \ldots, 0)$ for $0 \leqslant k < h$ and where the clients are positioned at $(\pm 2^0, \pm 2^{-1}, \ldots, \pm 2^{-h+1})$. Facilities with coordinates in $(\pm 2^0, \pm 2^{-1}, \ldots, \pm 2^{-k+1}, 0, \ldots, 0)$ are said to be of *level* k; there are 2^k of them. We denote by λ_i the level of Facility i. We organize the instance as a tree by declaring that the client or facility located at $u = (u_1, \ldots, u_h)$ is a *descendant* of all the facilities located at $(u_1, \ldots, u_k, 0, \ldots, 0)$ for $0 \leqslant k < h$. The instance T_h consists thus of $n = 2^h$ clients and $m = 2^h - 1$ facilities. The distance between any two locations u and v in the tree is equal to 2^{-k+1} where k is the level of their lowest common ancestor. All facilities have zero opening cost. Figure 1 shows a flat representation of T_5.

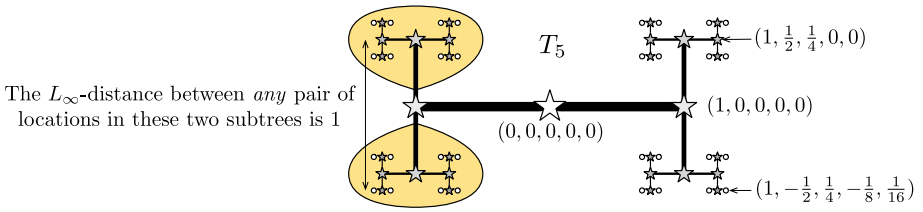

Fig. 1. A flat representation of the instance T_5 where each level of the tree lies in a different dimension. The clients and facilities are represented by circles and stars respectively. The levels of the facilities are represented by stars of decreasing size and edges of decreasing thickness.

Lemma 5 (Proof omitted, see appendix p. 14). *The optimal solutions to LP* (1) *for the static instance T_h have value* 1 *and the* uniform solution*, which opens a fraction $1/h$ of every facility i with radius $2^{-\lambda_i}$ and assigns each client to each of the h facilities covering it with fraction $1/h$, is optimal.*

We will first show that the adaptation of the ANS algorithm outputs a solution with cost $\Omega(\log \log n) \gg 1$ w.h.p. when presented with the uniform solution to LP (1) for T_h, and then show how to design a *dynamic* instance that forces LP (1) to output this uniform solution.

Running the adapted ANS algorithm on T_h with the uniform solution. The preprocessing leaves the uniform solution unchanged and the random variables Y_i

are i.i.d. according to an exponential law of parameter $1/h$. In order to improve readability, let us introduce $U_i = 1 - \exp(-Y_i/h)$ so that the U_is are uniformly distributed and ordered as the Y_is. The arcs in the graph G built by the algorithm at time 1 then consist of an arc for each client j, pointing to its ancestor facility i with the smallest U_i, and of an arc for each facility i, pointing to its descendant client j with the smallest X_j.

Lemma 6 (Proof omitted, see appendix p. 14). *The directed paths starting from a client in G have length at most 4 as illustrated by Fig. 2.*

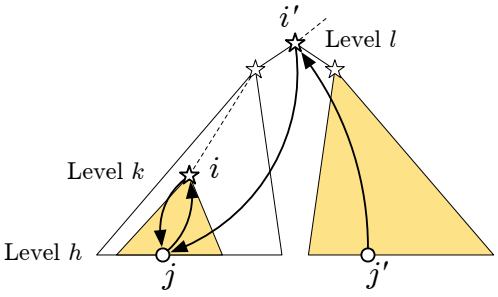

Fig. 2. Paths in the graph G built by ANS algorithm from the uniform solution for T_h.

To prove the $\Omega(\ln\ln n)$ lower bound (conditioned to the production of the uniform solution when solving LP (1)), we first need a combinatorial lemma, proved in the appendix. Let's consider a complete rooted binary tree A_q of height q where each node is labelled by a uniform random real chosen from $[0, 1]$ independently.

Lemma 7 (Proof omitted, see appendix p. 15). *The probability $p_q(x)$ that there is a branch in A_q where all the nodes have label $> x$ verifies:*

- *if $x < \frac{1}{2}$, then $2 - \frac{1}{1-x} < p_q(x) < 2 - \frac{1}{1-x} + \frac{(2x)^{q+2}}{4(1-x)}$ and $p_q(x) \searrow 2 - \frac{1}{1-x}$.*
- *if $x > \frac{1}{2}$, then $0 < p_q(x) < (1-x)(2(1-x))^q$ and $p_q(x) \searrow 0$.*

Using several technical lemmas in the appendix, we can prove the following:

Lemma 8 (Proof omitted, see appendix p. 15). *The expected opening cost of a facility i of level k is at least $2^{-k}(\ln k - \beta)/8h$ for a universal constant β.*

Which allows us to conclude that:

Lemma 9. *The expected opening cost of the solution output by the sum-of-radii ANS algorithm 3 from the uniform solution to LP (1) for T_h is $\Omega(\ln\ln n)$.*

Proof. By linearity of expectation and the lemmas above, $\mathbb{E}[\text{openCost}] \geqslant \sum_{k=10}^{h} 2^k \cdot 2^{-k}(\ln k - \beta)/8h = \Theta(\sum_{k=1}^{h}(\ln k)/h) = \Theta(\ln h) = \Theta(\ln\ln n)$ since $n = 2^h$.

4.2 Forcing the uniform fractional LP solution

Our lower bound for the T_h instance relies on running the algorithm on the uniform solution. Unfortunately, this solution is not a vertex of LP (1) and will not be output by any linear solver. We thus extend the instance T_h to a *dynamic* instance D_h whose optimal solution is unique and uniform, concluding the proof of Theorem 2. This D_h instance (detailed in the appendix) consists in several initial time steps with very low cost where the clients and facilities are mixed together (enforcing then the need for uniformity in the optimal solution), and a final time step equivalent to T_h.

Lemma 10 (Proof omitted, see appendix p. 18). *All optimal solutions to the instance D_h are uniform on the last time step.*

We can now conclude the proof of Theorem 2 through two corollaries.

Corollary 1 (Proof omitted, see appendix p. 18). *Algorithm 3 produces the same output for the last time step of D_h as for T_h.*

Let $D_h^{n^2}$ be the instance obtained by making n^2 independent copies of D_h located at distant locations in $\mathbb{R}^h$. The Hoeffding bound allows us to strengthen the result above by showing that the approximation ratio sum-of-radii ANS algorithm 3 on this new instance is at least $\Omega(\ln \ln n)$ with high probability, when run from the uniform solution to LP (1):

Corollary 2. *The opening cost of the solution output by the sum-of-radii ANS algorithm 3 from the uniform solution to LP (1) for $D_h^{n^2}$ is $\Omega(\ln \ln n)$ with probability $1 - 2^{-n}$.*

Proof. We directly apply the Hoeffding bound, observing that the cost of the solution output by sum-of-radii ANS algorithm 3 on D_h is at most twice the cost on T_h, hence at most $O(\log n)$.

5 Conclusion and Open Problems

We have obtained an asymptotically optimal $O(\log n)$-approximation algorithm for DSRC in the general case, with a technique that translates to the sum-of-distances case. We have also shown that the approximation ratio for the algorithm in [1] is no better than $\Omega(\ln \ln n)$ for metric instances. This leaves open the question of whether an $O(1)$-approximation algorithm exists in the metric case. Further experimental work has to be conducted to evaluate how these algorithms can help improve the representation of real dynamic graphs such as the ones in [15]. One final remark is that our algorithms all rely on the primal formulation of LP (1) while the algorithms in [3] for the static setting rely on the dual. Unfortunately, the dual variables seem to act evasively with respect to time in the dynamic setting. Understanding these dual variables is a promising direction towards an $O(1)$-approximation, if it exists.

References

1. Hyung-Chan An, Ashkan Norouzi-Fard, and Ola Svensson. Dynamic facility location via exponential clocks. In *SODA*, pages 708–721, 2015.
2. Babak Behsaz and Mohammad R. Salavatipour. On minimum sum of radii and diameters clustering. *Algorithmica*, 73(1):143–165, 2015.
3. Moses Charikar and Rina Panigrahy. Clustering to minimize the sum of cluster diameters. *J. Comput. Syst. Sci.*, 68(2):417–441, 2004.
4. Irit Dinur and David Steurer. Analytical approach to parallel repetition. In *STOC*, pages 624–633, 2014.
5. David Eisenstat, Claire Mathieu, and Nicolas Schabanel. Facility location in evolving metrics. In *ICALP*, pages 459–470, 2014.
6. Cristina G. Fernandes, Marcio T.I. Oshiro, and Nicolas Schabanel. Dynamic clustering of evolving networks: some results on the line. In *AlgoTel*, pages 1–4, May 2013.
7. Dimitris Fotakis and Paraschos Koutris. *Mathematical Foundations of Computer Science 2012: 37th International Symposium, MFCS 2012, Bratislava, Slovakia, August 27-31, 2012. Proceedings*, chapter Online Sum-Radii Clustering, pages 395–406. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
8. Sudipto Guha and Samir Khuller. Greedy strikes back: Improved facility location algorithms. *J. Algorithms*, 31(1):228–248, 1999.
9. Dorit S. Hochbaum. Heuristics for the fixed cost median problem. *Math. Program.*, 22(1):148–162, 1982.
10. Yin Tat Lee and Aaron Sidford. Path finding methods for linear programming: Solving linear programs in $\tilde{O}$(vrank) iterations and faster algorithms for maximum flow. In *Proceedings of the 2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, FOCS '14, pages 424–433, Washington, DC, USA, 2014. IEEE Computer Society.
11. Shi Li. A 1.488 approximation algorithm for the uncapacitated facility location problem. *Inf. Comput.*, 222:45–58, 2013.
12. Mohammad Mahdian, Yinyu Ye, and Jiawei Zhang. Approximation algorithms for metric facility location problems. *SIAM J. Comput.*, 36(2):411–432, 2006.
13. Mark E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003.
14. Romualdo Pastor-Satorras and Alessandro Vespignani. Epidemic spreading in scale-free networks. *Phys. Rev. Lett.*, 86:3200–3203, Apr 2001.
15. Juliette Stehlé, N. Voirin, Alain Barrat, Ciro Cattuto, Lorenzo Isella, Jean-François Pinton, Marco Quaggiotto, Wouter Van den Broeck, C. Régis, B. Lina, and P. Vanhems. High-resolution measurements of face-to-face contact patterns in a primary school. *PLoS ONE*, 6(8):e23176, 2011.
16. Siva R Sundaresan, Ilya R Fischhoff, Jonathan Dushoff, and Daniel I Rubenstein. Network metrics reveal differences in social organization between two fission-fusion species, grevy's zebra and onager. *Oecologia*, 151(1):140–149, 2007.
17. Chayant Tantipathananandh, Tanya Y. Berger-Wolf, and David Kempe. A framework for community identification in dynamic social networks. In *SIGKDD*, pages 717–726, 2007.

A Omitted proofs

A.1 Proof of Lemma 1 (Preprocessing)

Proof (Lemma 1). The proof is adapted to our LP and heavily inspired by the one in [5] that was used again in [1], with a slightly different LP.

Given a optimal solution (x^*, y^*, z^*) to LP (1), we want to compute a feasible solution (x, y, z) together with a collection of time intervals $I_{1,1}, \ldots, I_{1,\ell_1}, \ldots, I_{n,1}, \ldots, I_{n,\ell_n}$ such that:

- for all $i \in F$, $j \in C$, $t \in [T]$, $r \geqslant 0$: $x_{ijt} \leqslant 2x^*_{ijt}$ and $y_{irt} \leqslant 2y^*_{irt}$; and
- for all $j \in C$: $I_{j,1}, \ldots, I_{j,\ell_j}$ form a partition of $[T]$; and
- for all $i \in F$, $j \in C$ and $k \in [\ell_j]$: x_{ijt} is constant during each time interval I_{jk}; and
- for all $j \in C$: the total number of time interval changes for Client j, $\ell_j - 1$, verifies: $\ell_j - 1 \leqslant 2\sum_{it} z^*_{ijt}$.

To this end we set for each client j, $t^j_0 = 0$ and we are looking iteratively for t_{k+1}, equal to the biggest $t \in (t^j_k, T+1]$ such that $\sum_{i\in F}\left(\min_{t^j_k \leq u \leq t} x^*_{ijt}\right) \geq \frac{1}{2}$. If $t^j_{k+1} = T+1$ we stop here, if not we create a new interval corresponding to $[t^j_k, t^j_{k+1}]$ and look for t^j_{k+1}, until we end with $t_{\ell_j} = T+1$. We now have the correct intervals, but the third condition isn't satisfied yet. To do so, for each interval we set $x_{ijt} = 2 \times \min_{t^j_{k-1} \leq u \leq t_k} x^*_{iju}$. By construction, $2\sum_{i\in F} x_{iju} \geq 1$. By setting each $y_{irt} = 2 \times y^*_{irt}$ we also make sure that $x_{ijt} \leq \sum_{r \geq d_{ijt}} y_{irt}$, so the new solution is still feasible.

We can see that if between t^j_k and t^j_{k+1} we have $\sum_{i\in F}\left(\min_{t^j_{k-1} \leq u \leq t} x^u_{ij}\right) \leq \frac{1}{2}$, it means that $\sum_{t\in(t^j_k, t^j_{k+1}]} z^t_{ij} \geq \frac{1}{2}$. This in turn means that for each interval the initial solution paid at least $\frac{1}{2}g$, and here we pay at most g (to completely change all the x^t_{ij} between one interval and the next).

Hence both the changing cost and the facility opening cost are at most multiplied by 2 to achieve the property, and this preprocessing can be done in linear time. For the last assertion mentioned in the lemma, note that, if at any point $x_{ijt} > 1$, we can lower it to 1 without increasing any cost, and all clients will still be covered at all time steps. Similarly, if at some point for some i, $\sum_r y_{irt} > 1$, we can reduce the value of the positive y_{irt} with the smallest r, which improves the cost of the solution and preserves the constraints.

A.2 Hardness

Proposition 1 (Hardness, folklore). *The Dynamic Sum-of-Radii Clustering problem admits no* $(1 - o(1)) \ln n$*-approximation unless* $P = NP$*.*

Proof. We consider a Set Cover instance $A_1, \ldots, A_m \subseteq [n]$ and create an instance of DSRC with opening cost $f_i = 1$, changing cost $g = \infty$ and $T = 1$. It consists of one facility i per set A_i and a set of n clients, one per element. We set the distances d_{ij} to 0 if j belongs to A_i and to ∞ otherwise. Each solution with finite cost to the DSRC corresponds to a collection of sets covering all the elements of the Set Cover instance with the same cost, and reciprocally. An algorithm that would guarantee a $(1 - o(1)) \times \ln n$ approximation on DSRC would then guarantee the same for the Set Cover problem which would imply $P = NP$ according to [4].

Remark 1. In the theorem we set g to ∞ for ease of reading but it is enough to set it to m. Moreover, this result uses a single time step hence also holds for (non-metric) Static Clustering with Sum of Radii as well.

A.3 Proofs for the hard instances T_h and D_h

Proof of the two Lemmas about instance T_h

Proof (Lemma 5). We proceed by recurrence. There is only one optimal solution in T_0 (open the only facility with radius 1) and it costs 1. Assume that all optimal solutions cost 1 for T_{h-1}. Consider an optimal solution in T_h. Note that if a fraction α of a facility of level $k \geqslant 1$ is used to cover some client in the opposite subtree of the root, it has to be open with radius at least 2, and one can save α by opening instead a fraction α of the root facility. Now, if a fraction α of the facility of level 0 is open, as no facilities are used to cover each other's side, we are left with two instances of T_{h-1} downscaled by $(1 - \alpha)/2$, for which the optimal cost is $2(1 - \alpha)/2$ by recurrence. Hence, optimal solutions to LP (1) for instance T_h cost 1.

Finally, the uniform solution which opens a fraction $1/h$ of every facility i with radius $2^{-\lambda_i}$ covers every client (since they are at distance 2^{-k} of their level-k ancestor facility) and costs $\sum_{k=0}^{h-1} 2^k \cdot 2^{-k}/h = 1$; it is thus optimal.

Proof (Lemma 6). Start from a client j' pointing to the ancestor facility i' with the smallest $U_{i'}$. Let j be the client pointed by i', i.e. its descendant of minimum X_j. If $j = j'$, then the path has length 2. If $j \neq j'$, let i be the facility pointed by j. Note that i is necessarily a descendant of i' since j' pointing to i' implies that $U_{i'}$ is smaller than any of its ancestors. If $i = i'$, the path has length 3. Finally, if $i \neq i'$, as j is the client with the smallest X_j among the descendants of i' which include the descendants of i, X_j is also the smallest among the descendants of i and i necessarily points to j, thus the path has length 4.

Proof of Lemma 7 (Combinatorial lemma)

Proof (Lemma 7). $p_q(x)$ verifies for all $q \geqslant 0$:

$$\begin{cases} p_0(x) = 1 - x \\ p_{q+1}(x) = (1-x)(1-(1-p_q(x))^2) = (1-x)p_q(x)(2-p_q(x)) \end{cases}$$

If $1/2 < x < 1$, then $0 < p_q(x) < 2(1-x) \cdot p_{q-1}(x) < (1-x)(2(1-x))^q$ for all $q \geqslant 0$.

Now, assume that $0 < x < 1/2$. Let $f(p) = (1-x)p(2-p)$. f is increasing from $[0,1]$ onto $[0, 1-x]$, strictly convex, and verifies: $f(p) = p \Leftrightarrow (p = 0 \text{ or } p = 2 - \frac{1}{1-x})$, $f'(0) = 2(1-x) > 1$ and $f'(2 - \frac{1}{1-x}) = 2x > 0$ and $2 - \frac{1}{1-x} \leqslant 1 - x \leqslant 1$. It follows that the sequence $p_q(x) = f^q(1-x) \searrow 2 - \frac{1}{1-x}$. Furthermore, let $\epsilon_q = p_q(x) - 2 + \frac{1}{1-x}$. We have:

$$\begin{aligned} \epsilon_{q+1} &= f(2 - \frac{1}{1-x} + \epsilon_q) - 2 + \frac{1}{1-x} = \epsilon_q(2x - (1-x)\epsilon_q) \\ &< 2x \cdot \epsilon_q < (2x)^q \epsilon_0 < \frac{(2x)^{q+2}}{4(1-x)}, \end{aligned}$$

since $\epsilon_0 = -1 - x + \frac{1}{1-x} = \frac{x^2}{1-x}$.

Expected cost of a facility

Before proving the Lemma, we need a few preliminary results.

Lemma 11. *Each facility i is open with probability $1/h$.*

Proof. A facility i is open if its descendant client j with smallest X_j points to i, i.e. if U_i is smaller than $U_{i'}$ for each i' among the h ancestors of j. As the X_js and U_is are independent, this happens with probability exactly $1/h$.

Let us now consider an open facility i. A client j' which is not a descendant of i might be assigned to i if it points to a facility i' that points to a descendant j of i as illustrated in Fig. 2, in which case Facility i will be open with radius 2^l instead of 2^k, where k and l are respectively the levels of i and the closest common ancestor of j and j'.

Lemma 12. *Given an open facility i of level k, i will end up being open with radius at least 2^{-l} by the sum-of-radii ANS algorithm with probability at least $2^{l-k}/8l$ if $k \geqslant l \geqslant 10$.*

Proof. Given that Facility i is open and points to a client j that points to i, i will be open with radius at least 2^{-l} if the following events occur together (see Fig. 2):

(E1) The ancestor i' of level l of i verifies: $U_{i'} < U_{i''}$ for each i'' among the ancestors of i'.

(E2) $X_j < X_{j''}$ for each client j'' descendant of the facility i'.

(E3) The subtree of i' that does not contain i, contains a branch B leading to some leaf j' pointing to i', i.e. such that $U_{i''} > U_{i'}$ for all facility $i'' \in B$.

All probabilities in the following are conditioned to i being open and pointing to some client j. The probability of event (E1) is $1/l$ as the U-values are u.i.d. The probability of event (E2) is $2^{(l-h)-(h-k)} = 2^{l-k}$ since the X-values are u.i.d. and since j has already the minimum X-value among the clients descending from i. Furthermore, by Markov's inequality, as the expected value of the minimum of k u.i.d. reals in $[0,1]$ is $\frac{1}{k+1}$, we have that (E4) $X_i < \frac{2}{k+1}$ with probability at least $\frac{1}{2}$.

Given X_i and (E1), $X_{i'}$ is distributed as the minimum of l uniform random reals in $(X_i, 1]$ and its expected value is thus $X_i + \frac{1-X_i}{l+1}$. Consequently, given the events (E1) and (E4), Markov's inequality gives that with probability at least $\frac{1}{2}$: $X_{i'} < \frac{2}{k+1} + 2\frac{1-\frac{2}{k+1}}{l+1} < \frac{1}{3}$ for all $k \geqslant l \geqslant 10$. According to Lemma 7, if $X_{i'} < \frac{1}{3}$, then event (E3) occurs with probability at least $2 - \frac{1}{1-\frac{1}{3}} = \frac{1}{2}$. We conclude by independence of the X- and U-values that (E1), (E2) and (E3) occur together with probability at least $2^{l-k}/(l \times 2 \times 2 \times 2)$ as soon as $k \geqslant l \geqslant 10$.

We can now finish the proof :

Proof (Lemma 8). Let r_i be the radius at which i is open ($r_i = 0$ if i is closed). Then $\Pr\{r_i \geqslant 2^{-l}\} = \Pr\{i \text{ is open}\} \cdot \Pr\{r_i \geqslant 2^{-l} \mid i \text{ is open}\} \geqslant 2^{l-k}/8lh$. Thus, $\mathbb{E}[r_i] = \int_0^\infty \Pr\{r_i \geqslant r\}dr \geqslant \sum_{l=10}^{k} 2^{-l}2^{l-k}/8lh = 2^{-k}(\ln k - \beta)/8h$ for some universal constant β.

The hard dynamic instance D_h

Definition 1 (The simplex instance). *The instance S_h consists in $h+1$ clients, all at distance 2 from each other, together with $h+1$ facilities, such that the ith facility is at distance 1 from all clients but the ith, from which it is at distance 2. All clients and facilities are at distance 1 from the origin. Recall that any distances with values 1 or 2 are metric. This structure is also realized in $(\mathbb{R}^h, L_\infty)$ by placing the clients at the vertices of an $(h+1)$-simplex of side 2 centered at the origin, and the facilities at the center of the facets.*

This simplex instance will be used to make sure that the $h+1$ clients are all nearly uniformly attached to the facilities, and by using many structures of that kind (one per time step), we can force the clients to have a small preference to have an uniform attachment to the facilities.

Lemma 13. *The LP* (1) *for* S_h *admits a unique optimal solution: open a fraction* $\frac{1}{h}$ *of each facility with radius* 1*.*

Proof. All facilities are open with radius 0, 1 or 2 as their distances to the clients are either 1 or 2. The cost of the uniform solution proposed is $\frac{h+1}{h}$. Consider an optimal solution. Suppose first that a fraction α of some facility is open with radius 2. Opening a fraction $\frac{\alpha}{h+1}$ of all facilities covers the same fraction of the clients and costs only $\alpha\frac{h+1}{h}$ instead of 2α contradicting its optimality. Hence all open facilities are open with radius 1. If some facility i were open with fraction less than $\frac{1}{h}$, then at least one other facility i' would need to be open with fraction more than $\frac{1}{h}$ to cover the clients on i's facet. As the single client at distance 2 from Facility i' cannot be covered by i' and covering it costs at least 1, and the total cost would then be more than $1 + \frac{1}{h}$, hence not optimal.

Definition 2 (The hard instance). *We consider the following dynamic instance* D_h *with* $T = 2^h + 1$ *time steps,* $(h+1)2^h$ *clients and* $2^h - 1$ *facilities plus a special facility. It has no opening costs (*$f_{it} = 0$*) and the changing cost is* $g = 2^{-4h}$*. This instance goes through two phases:*

- *The last time step consists of the structure* T_h *where each client* j *is replaced by* $h + 1$ *copies of itself at the same location, plus the special facility located far far away. We denote by* σ_j *the set of the* $h + 1$ *copies of* j*.*
- *The first* 2^h *steps consist of a series of scaled-down simplices. At time* j*, for* $j = 1..2^h$*, the* $h + 1$ *clients in* σ_j *and the* h *ancestor facilities of* j *plus the special facilities adopt the simplex structure* S_h *scaled down by* $s = 2^{-4h}$*; all the other clients and facilities are located at the origin.*

We define the *uniform solution* D_h as the solution where every client in σ_j assigns a fraction $\frac{1}{h}$ to the h facilities at distance s in their simplex during the first 2^h time steps and moves (for h of them) the fraction $\frac{1}{h}$ it has assigned on the special facility to the ancestor of j on which it had no assignment yet. In this solution, all facilities are open at all time steps: with radius 0 (when at the origin) or s (when at a vertex) in the simplex time steps, and with radius $2^{-\lambda}$ (where λ is their level in T_h) at the last step.

Lemma 14. *The total cost of* D_h *is at least* $1 + 2^h \cdot \frac{h+1}{h} \cdot s$ *and the uniform solution has cost* $1 + 2^h \cdot \frac{h+1}{h} \cdot s + 2^h \cdot g$ *.*

Proof. Every solution costs at least as much as the static solution for each time step. As we have 2^h steps with cost at least $\frac{h+1}{h} \cdot s$ each, and one last step with cost at least 1, we get the first part. This, however, ignores changing costs. The uniform solution consists of the best static solutions at each time step. The changing cost is paid at the last time step and involves moving a fraction $h \cdot \frac{1}{h} \cdot 2^h$ from the special facility (now far far away) to other facilities. This costs $2^h \cdot g$ in total.

Lemma 15. *Any solution that does not correspond to a static optimal on each time step can be improved.*

Proof. Suppose that a fraction $\frac{1}{h} + \epsilon$ (for some $\epsilon > 0$) of a client is assigned to a facility during its simplex phase. It costs at least an additional ϵs to cover the client at this step. Adopting the static optimal solution, we save ϵs on the opening cost for the facility in this step, and increase the changing costs by at most $(h+1)g = (h+1)2^{-4h} \ll \epsilon s$. The same goes for the last step, where increasing the fraction by ϵ costs at least $\epsilon \cdot 2^{-h} \gg \epsilon \cdot g$.

This allows us to prove Lemma 10:

Proof (Lemma 10). In every solution to the first 2^h time steps, we can defer every changing cost to the last time step as all clients covered during their simplex time step are covered during all simplices steps at no additional costs (since they are located at the origin, covered by all facilities). This means that one can assume that before the last step, the solution is uniform. During the last time step, every client must move the $\frac{1}{h}$ fraction it had put on the virtual facility to another facility. There is always one open facility covering the client on which it can put that fraction, resulting in no additional costs besides the changing cost. If some client chose another facility instead, this facility would have to increase either its fraction or radius, yielding a strictly costlier and thus non-optimal solution.

And finally, to prove Corollary 1:

Proof (Corollary 1). The special facility is not selected in the last time step so it doesn't affect the algorithm. As all the sets σ_j have the same size, as all the random variables Xs and Ys are independent, and as the execution of the algorithm only depends on their relative order, all the clients behave as if there were only one client in each set σ_j. The cost of the algorithm at this last step is thus identical to the one for T_h.