A Polynomial-Time Approximation Scheme for Facility Location on Planar Graphs

Vincent Cohen-Addad
Sorbonne Université, CNRS,
Laboratoire d'informatique de Paris 6, LIP6,
F-75252 Paris, France
vincent.cohen.addad@ens-lyon.org

Marcin Pilipczuk
Michał Pilipczuk
Institute of Informatics, University of Warsaw,
Warsaw, Poland
{michal,marcin}.pilipczuk@mimuw.edu.pl

Abstract—**We consider the classic FACILITY LOCATION problem on planar graphs (non-uniform, uncapacitated). Given an edge-weighted planar graph G, a set of clients $C \subseteq V(G)$, a set of facilities $F \subseteq V(G)$, and opening costs** open$: F \to \mathbb{R}_{\geqslant 0}$**, the goal is to find a subset D of F that minimizes** $\sum_{c \in C} \min_{f \in D} \operatorname{dist}(c, f) + \sum_{f \in D} \operatorname{open}(f)$**.**

The FACILITY LOCATION problem remains one of the most classic and fundamental optimization problem for which it is not known whether it admits a *polynomial-time approximation scheme* (PTAS) on planar graphs despite significant effort for obtaining one. We solve this open problem by giving an algorithm that for any $\varepsilon > 0$, computes a solution of cost at most $(1 + \varepsilon)$ times the optimum in time $n^{2^{\mathcal{O}(\varepsilon^{-2} \log(1/\varepsilon))}}$.

Keywords-facility location, polynomial-time approximation scheme, planar metric

I. INTRODUCTION

We study the classic FACILITY LOCATION objective in planar metrics. Given an edge-weighted planar graph G, together with a set C of vertices called *clients*, a set F of vertices called *candidate facilities*, and *opening costs* open$: F \to \mathbb{R}_{\geqslant 0}$, the FACILITY LOCATION problem asks for a subset D of F that minimizes $\sum_{c \in C} \min_{f \in D} \operatorname{dist}(c, f) + \sum_{f \in D} \operatorname{open}(f)$.

The FACILITY LOCATION objective is a model of choice when trying to identify the best location for public infrastructures such as hospitals, water tanks or fire stations, or when looking for the best location for warehouses or delivery stores. More recent applications also include prepositionning transportation resources such as bikes, scooters, or cabs. This has made FACILITY LOCATION a fundamental problem that attracted a lot of attention over the years, both in theoretical computer science and in operations research communities. Since the problem is NP-hard, but one is often satisfied with a near-optimum solution, a large volume of work was devoted to the design of approximation algorithms [1], [2], [3], [4], culminating with the 1.488-approximation algorithm

by Li [5]. Unfortunately, there is no chance of going much beyond this result, as the problem is known to be NP-hard to approximate within factor better than 1.46-approximation [6].

Therefore, a natural route is to consider restricted metrics arising in applications. For example, when the underlying metric of the application is a road networks, the shortest path metric induced by edge-weighted planar graphs is model of choice. Thus, it has been a long standing open question whether FACILITY LOCATION admits a polynomial-time approximation scheme on planar graphs. For the uniform case, this was resolved only recently in the affirmative by Cohen-Addad et al. [7] using a simple local search algorithm: given a solution D, determine whether there exists a solution D' of better cost that differs from D by at most $\mathcal{O}(1/\varepsilon^2)$ centers. If so, take D' as the new solution and repeat, otherwise output D. However, no such approach is known to work in the nonuniform case and, in fact, it is easy to show that the same local search heuristic would provide a solution of cost at least twice the optimum in the worst-case for planar inputs. This has been a major roadblock since local search is the only technique we know so far for obtaining approximation schemes to min-sum clustering objectives such as the classic k-median, k-means or for uniform facility location, despite a significant effort from the community. In fact, and perhaps surprisingly, such a situation is not unique. For the problem of computing a maximum independent set of pseudo-disks, local search yields a PTAS in the unweighted case and it remains an important open problem as whether a PTAS exists for the weighted case [8]. Thus, obtaining a PTAS for the "weighted" version of some problems seems a much harder task than for the unweighted case.

Our main result is a polynomial-time approximation scheme for the (nonuniform, uncapacitated) FACILITY LOCATION problem in planar graphs. From a complexity perspective, our result refutes APX-hardness of FACILITY LOCATION on planar graphs (unless NP = P). From a techniques perspective, we believe that our approach provides a new set of interesting tools, such as for example a "metric-Baker" layering tailored to min-sum objectives (and so of a different nature than the "metric-Baker" used for k-center in recent works [9], [10]). More formally, we show that

following theorem.

Theorem 1. *Given a* FACILITY LOCATION *instance* (G, C, F, open), *where* G *is a planar graph, and an accuracy parameter* $\varepsilon > 0$, *one can in* $n^{2^{\mathcal{O}(\varepsilon^{-2} \log(1/\varepsilon))}}$ *time compute a solution of cost at most* $(1 + \varepsilon)$ *times the optimum cost.*

We now describe the structure of the proof and our algorithm. To do so conveniently, let us first introduce some terminology: we define for a set $D \subseteq F$, the *connection cost* of D is as $\mathsf{conn}(D, C) = \sum_{c \in C} \mathrm{dist}(c, D)$ and the *opening cost* of D as $\sum_{f \in D} \mathsf{open}(f)$.

The first step of our algorithm is to compute an $\mathcal{O}(1)$-approximate solution to a modified input instance where every opening cost is scaled down by a factor of ε. This solution $\widetilde{D}$ is computed through a greedy procedure and it is still an $\mathcal{O}(\varepsilon^{-1})$-approximation to the original instance. Interestingly, this solution reveals a lot of structure of the input graph metric, which will be crucial for the proof of Theorem 1. Indeed, the proof of the theorem and our algorithm can be broken into two pieces. The first one consists in a partitioning of the instance into separate, more structured, and almost independent sub-instances (based on the output of the greedy procedure). The second one is a heavily technical dynamic programming algorithm for solving these sub-instances.

To understand how the two pieces articulate, we need to introduce a couple of definitions. Let $f \in \widetilde{D}$ be an opened facility and let $\mathsf{cluster}(f)$ be the set of clients connected to f in the solution $\widetilde{D}$ (i.e., $\mathsf{cluster}(f)$ consists of these clients $c \in C$ for which f is the closest facility from $\widetilde{D}$). The *average cost* of $\mathsf{cluster}(f)$ is defined as:

$$\mathsf{avgcost}(f) = \frac{\mathsf{open}(f) + \sum_{c \in \mathsf{cluster}(f)} \mathrm{dist}(c, f)}{|\mathsf{cluster}(f)|}.$$

At a high-level, the sub-instances will be defined by dividing the metric space according to the clustering induced by $\widetilde{D}$: putting in the same instance the clusters of $\widetilde{D}$ that have roughly the same $\mathsf{avgcost}$ values. More concretely, a deep analysis of the structure of the approximate solution $\widetilde{D}$ and an intricate Baker-type layering step based on average costs of the facilities of $\widetilde{D}$ yields an instance such that (i) all values of $\mathsf{avgcost}(f)$ for $f \in \widetilde{D}$ are within constant ratio from each other, and (ii) for every $f \in \widetilde{D}$ and $c \in \mathsf{cluster}(f)$ the distance $\mathrm{dist}(c, f)$ is within constant ratio of $\mathsf{avgcost}(f)$. This is described in Section II.

The second part of the algorithm described in Section III, consists mainly of our technical dynamic programming algorithm for solving the instances produced in the first part.

II. REDUCING TO THE CONSTANT SCOPE OF THE AVERAGE COSTS

Setup: We shall work with an instance $I = (G, C, F, \mathsf{open})$ where G is a planar edge-weighted graph, $C \subseteq V(G)$ is a set of clients, $F \subseteq V(G)$ is a set of

facilities, and $\mathsf{open} \colon F \to \mathbb{R}_{\geqslant 0}$ defines the opening cost of facilities. We shall assume that G is embedded in a sphere and that distances between pairs of vertices of G are finite and pairwise distinct.

For a set of clients $S \subseteq C$ and solution $R \subseteq F$, by $\mathsf{conn}(S, R)$ we denote the contribution of clients from S to the connection cost of R and by $\mathsf{open}(R)$ the opening cost of R. That is,

$$\mathsf{conn}(S, R) = \sum_{c \in S} \min_{f \in R} \mathrm{dist}(c, f)$$

and

$$\mathsf{open}(R) = \sum_{f \in R} \mathsf{open}(f).$$

We write $\mathsf{conn}(R)$ for $\mathsf{conn}(C, R)$. Thus, the cost of R is defined as $\mathsf{cost}(R) = \mathsf{conn}(R) + \mathsf{open}(R)$. For the remainder of this section, let us fix some optimum solution D in I, and we denote $\mathsf{OPT} = \mathsf{cost}(D)$.

We consider the accuracy parameter $\varepsilon > 0$; w.l.o.g. we assume that $\varepsilon < 1/10$. Our goal is to compute a $(1 + c\varepsilon)$-approximate solution for some constant c, so that ε can be scaled appropriately at the end.

Recall that the considered problem admits a constant-factor approximation for the problem: as shown by Li [5], given an instance of non-uniform facility location one can in polynomial time find a solution of cost at most α times the optimum, where $\alpha = 1.488$. We apply this algorithm to the input instance, obtaining a solution $D' \subseteq F$, and we rescale the distances and the opening costs by the same factor so that

$$\mathsf{cost}(D') = \varepsilon^{-1} \cdot (|F| + |C| \cdot |E(G)|).$$

Note that this means that the total contribution of edges of length less than 1 and facilities of opening cost less than 1 to any solution is bounded by $|F| + |C| \cdot |E(G)| \leqslant \varepsilon \cdot \mathsf{cost}(D') \leqslant \alpha\varepsilon \cdot \mathsf{OPT}$. Thus, at the cost of paying at most $\varepsilon \cdot \mathsf{cost}(D') \leqslant \alpha\varepsilon \cdot \mathsf{OPT}$ we may assume that all edges of length less than 1 can be traversed for free, hence we may simply contract them. Similarly, we zero the opening costs of all facilities whose opening cost is less than 1. Therefore, we assume that all edges in G have weight at least 1 and all opening costs are either 0 or at least 1, while

$$\mathsf{OPT} = \Theta(\varepsilon^{-1} \cdot (|F| + |C| \cdot |E(G)|)). \tag{1}$$

Robust approximate solution: Let us consider the modified instance

$$\widetilde{I} = (G, C, F, \varepsilon \cdot \mathsf{open});$$

that is, the instance is the same as I but all the opening costs are scaled down by a multiplicative factor of ε. For a solution $R \subseteq F$, we denote the cost of R in the instance $\widetilde{I}$ by $\mathsf{cost}(R; \widetilde{I})$; note that $\mathsf{cost}(R; \widetilde{I}) = \mathsf{conn}(R) + \varepsilon \cdot \mathsf{open}(R)$. Note that for any $R \subseteq F$, we have $\varepsilon \cdot \mathsf{cost}(R) \leqslant \mathsf{cost}(R; \widetilde{I}) \leqslant \mathsf{cost}(R)$.

We apply the aforementioned α-approximation algorithm of Li [5] to the instance $\widetilde{I}$. Furthermore, we will need the following property from the returned approximate solution $\widetilde{D}$:

$$\mathsf{cost}(\widetilde{D} \cup \{f\}; \widetilde{I}) \geqslant \mathsf{cost}(\widetilde{D}; \widetilde{I}) \quad \text{for every } f \in F; \quad (2)$$

This is trivially true for any $f \in \widetilde{D}$ and to ensure that this holds for every f, we make use of the following greedy process. As long as there exists a facility $f \in F \setminus \widetilde{D}$ that violates the condition above, we add it to $\widetilde{D}$.

Finally, at the end of this greedy process we remove from $\widetilde{D}$ all facilities that do not serve any client, that is, we remove all facilities $f \in \widetilde{D}$ such that for every $c \in C$ we have $\mathrm{dist}(c, \widetilde{D}) < \mathrm{dist}(c, f)$. Note that this step does not increase the cost of $\widetilde{D}$ and does not break property (2). We now start analysing the structure of $\widetilde{D}$.

We start by verifying that $\widetilde{D}$ is actually an $\mathcal{O}(\varepsilon^{-1})$-approximate solution in the original instance.

Lemma 2. *We have* $\mathsf{cost}(\widetilde{D}) \leqslant \alpha \varepsilon^{-1} \cdot \mathsf{OPT}$.

Proof: Recalling that D is an optimum solution in I, we have that

$$\mathsf{cost}(D; \widetilde{I}) \leqslant \mathsf{cost}(D) = \mathsf{OPT}.$$

On the other hand, $\widetilde{D}$ is an α-approximate solution in $\widetilde{I}$, hence

$$\mathsf{cost}(\widetilde{D}; \widetilde{I}) \leqslant \alpha \cdot \mathsf{cost}(D; \widetilde{I})$$

Finally, as observed before we have

$$\varepsilon \cdot \mathsf{cost}(\widetilde{D}) \leqslant \mathsf{cost}(\widetilde{D}; \widetilde{I}).$$

Combining the above three inequalities yields the claim. ∎

Let $R \subseteq F$ be a nonempty set of facilities. For a facility $f \in R$, the *R-cluster* of f, denoted $\mathsf{cluster}(f, R)$, is the set of all clients that are served by f in the solution R; that is:

$$\mathsf{cluster}(f, R) = \{c \in C \colon \mathrm{dist}(c, f) = \min_{g \in R} \mathrm{dist}(c, g)\}.$$

Note that since distances between pairs of vertices in G are pairwise different, the R-clusters are pairwise disjoint. In the sequel we will most often work with $\widetilde{D}$-clusters, hence we use shorthands: a *cluster* means a $\widetilde{D}$-cluster and for $f \in \widetilde{D}$ we denote $\mathsf{cluster}(f) = \mathsf{cluster}(f, \widetilde{D})$.

The next lemma intuitively says the following: for any subset of clients, its connection cost in $\widetilde{D}$ is not much larger than its connection cost D.

Lemma 3. *For any subset of clients $S \subseteq C$ we have*

$$\mathsf{conn}(S, \widetilde{D}) \leqslant \mathsf{conn}(S, D) + \varepsilon \cdot \mathsf{open}(D).$$

Proof: For any $f \in D$, let

$$\sigma(f) = \mathsf{conn}(\mathsf{cluster}(f, D) \cap S, D) + \varepsilon \cdot \mathsf{open}(f).$$

Observe that the right hand side of the inequality is equal to $\sum_{f \in D} \sigma(f)$.

Consider modifying the solution $\widetilde{D}$ by opening facility f, for any $f \in D$, and applying (2). If in solution $\widetilde{D} \cup \{f\}$ we consider directing all clients of $\mathsf{cluster}(f, D) \cap S$ to f and all other clients as in $\widetilde{D}$, then

$$
\begin{aligned}
0 &\leqslant \mathsf{cost}(\widetilde{D} \cup \{f\}; \widetilde{I}) - \mathsf{cost}(\widetilde{D}; \widetilde{I}) \\
&\leqslant \mathsf{conn}(\mathsf{cluster}(f, D) \cap S, D) \\
&\quad -\mathsf{conn}(\mathsf{cluster}(f, D) \cap S, \widetilde{D}) + \varepsilon \cdot \mathsf{open}(f) \\
&= \sigma(f) - \mathsf{conn}(\mathsf{cluster}(f, D) \cap S, \widetilde{D}).
\end{aligned}
$$

By summing the above inequality through all $f \in D$, we infer that

$$
\begin{aligned}
0 &\leqslant \sum_{f \in D} \sigma(f) - \sum_{f \in D} \mathsf{conn}(\mathsf{cluster}(f, D) \cap S, \widetilde{D}) \\
&= (\mathsf{conn}(S, D) + \varepsilon \cdot \mathsf{open}(D)) - \mathsf{conn}(S, \widetilde{D}).
\end{aligned}
$$

This establishes the claim. ∎

For any $f \in \widetilde{D}$, we define the *average cost* of f as

$$\mathsf{avgcost}(f) = \frac{\mathsf{open}(f) + \sum_{c \in \mathsf{cluster}(f)} \mathrm{dist}(c, f)}{|\mathsf{cluster}(f)|}.$$

Note that in this definition we use the original opening costs of facilities, not the scaled-down ones. Recall here that $\mathsf{cluster}(f)$ is nonempty for each $f \in \widetilde{D}$ as we removed from $\widetilde{D}$ all facilites that do not serve any clients. Moreover, we have

$$\mathsf{cost}(\widetilde{D}) = \sum_{f \in \widetilde{D}} |\mathsf{cluster}(f)| \cdot \mathsf{avgcost}(f). \quad (3)$$

Next, we prove that for every cluster $\mathsf{cluster}(f)$ for any $f \in \widetilde{D}$, there is always a facility of the optimum solution D that is not far from f, measured in terms of $\mathsf{avgcost}(f)$. We first state the lemma in a very abstract form so that we can apply it later in various settings.

Lemma 4. *Let $I = (G, C, F, \mathsf{open})$ be a* NON-UNIFORM FACILITY LOCATION *instance, $R \subseteq F$ a nonempty set of facilities, $K \subseteq C$ a nonempty set of clients, and let $f \notin R$ be a facility. Assume that*

$$\mathrm{dist}(f, R) > \frac{2}{|K|} \cdot \left(\mathsf{open}(f) + \sum_{c \in K} \mathrm{dist}(c, f) \right).$$

Then $\mathsf{cost}(R; I) > \mathsf{cost}(R \cup \{f\}; I)$.

Proof: Let

$$a := \frac{\mathsf{open}(f) + \sum_{c \in K} \mathrm{dist}(c, f)}{|K|}.$$

Observe that every client $c \in \mathsf{cluster}(f)$ has to be served in solution R by a facility that is at distance more than $2a$ from f, implying by triangle inequality that

$$\min_{g \in R} \mathrm{dist}(c, g) > 2a - \mathrm{dist}(c, f).$$

Take solution $R \cup \{f\}$. By considering directing all the clients of K to f, and all other clients as in R, we observe that

$$\operatorname{cost}(R \cup \{f\}) - \operatorname{cost}(R)$$
$$\leqslant \sum_{c \in K} \operatorname{dist}(c, f) - \sum_{c \in K} \min_{g \in R} \operatorname{dist}(c, g) + \operatorname{open}(f)$$
$$< \left(2 \sum_{c \in K} \operatorname{dist}(c, f) + \operatorname{open}(f)\right) - 2|K| \cdot a$$
$$\leqslant 2|K| \cdot a - 2|K| \cdot a = 0.$$

This implies that $\operatorname{cost}(R \cup \{f\}) < \operatorname{cost}(R)$ as desired. ∎

Corollary 5. *For every $f \in \widetilde{D}$ there exists $g \in D$ such that $\operatorname{dist}(f, g) \leqslant 2 \cdot \operatorname{avgcost}(f)$.*

Proof: The claim is obvious for $f \in D$. Otherwise, we apply Lemma 4 to the instance I, optimum solution D, the facility f, and $K = \operatorname{cluster}(f)$. The optimality of D implies then that $\operatorname{dist}(f, D) \leqslant 2 \cdot \operatorname{avgcost}(f)$. ∎

Concentrating the clusters: We now analyze every cluster $\operatorname{cluster}(f)$ for $f \in \widetilde{D}$ and show that, at the cost of changing the value of OPT only slightly, we may assume that all clients of $\operatorname{cluster}(f)$ have connection cost w.r.t. $\widetilde{D}$ not differing much from $\operatorname{avgcost}(f)$. More precisely, we would like to get rid of clients that are far and close according to the following definition: for $f \in \widetilde{D}$, let

$$\mathsf{Far}(f) = \{c \in \operatorname{cluster}(f) \colon \operatorname{dist}(c, f) > \varepsilon^{-2} \cdot \operatorname{avgcost}(f)\},$$
$$\mathsf{Close}(f) = \{c \in \operatorname{cluster}(f) \colon \operatorname{dist}(c, f) < \varepsilon^{2} \cdot \operatorname{avgcost}(f)\}.$$

Moreover, we define

$$\mathsf{Far} = \bigcup_{f \in \widetilde{D}} \mathsf{Far}(f) \quad \text{and} \quad \mathsf{Close} = \bigcup_{f \in \widetilde{D}} \mathsf{Close}(f).$$

Let
$$\Psi = \operatorname{conn}(\mathsf{Far}, \widetilde{D}).$$

For each $f \in \widetilde{D}$ let us pick any vertex $x(f)$ of G that is at distance exactly $\varepsilon^{2} \cdot \operatorname{avgcost}(f)$ from f (subdividing some edge, if a priori there is none). Construct C' from C by performing the following operation for each $f \in \widetilde{D}$: move all clients of $\mathsf{Far}(f) \cup \mathsf{Close}(f)$ to $x(f)$, thus placing $|\mathsf{Far}(f)| + |\mathsf{Close}(f)|$ clients at $x(f)$. Similarly, for $f \in \widetilde{D}$ we define $\operatorname{cluster}'(f)$ to be the image of $\operatorname{cluster}(f)$ under this operation, i.e. with clients from $\mathsf{Far}(f) \cup \mathsf{Close}(f)$ replaced as above.

Let
$$I' = (G, C', F, \operatorname{open});$$

that is, I' is constructed from I by replacing the client set with C'. Let OPT' be the minimum cost of a solution in the instance I'. We now verify that in order to find near-optimum solution to I, it suffices to find a near-optimum solution to I'.

Lemma 6. *We have*
$$\mathsf{OPT}' \leqslant (1 + 6\alpha\varepsilon)\mathsf{OPT} - \Psi$$

Moreover, for every $R \subseteq F$, we have
$$\operatorname{cost}(R; I) \leqslant \operatorname{cost}(R; I') + \Psi + 3\alpha\varepsilon \cdot \mathsf{OPT}.$$

Proof: For the first inequality, note that we have

$$\operatorname{conn}(C', D)$$
$$= \operatorname{conn}(C, D)$$
$$+ \sum_{f \in \widetilde{D}} \sum_{c \in \mathsf{Far}(f)} (\operatorname{dist}(x(f), D) - \operatorname{dist}(c, D))$$
$$+ \sum_{f \in \widetilde{D}} \sum_{c \in \mathsf{Close}(f)} (\operatorname{dist}(x(f), D) - \operatorname{dist}(c, D))$$
$$\leqslant \operatorname{conn}(C, D)$$
$$+ \sum_{f \in \widetilde{D}} \sum_{c \in \mathsf{Far}(f)} (\operatorname{dist}(x(f), D) - \operatorname{dist}(c, D))$$
$$+ \sum_{f \in \widetilde{D}} \sum_{c \in \mathsf{Close}(f)} \operatorname{dist}(c, x(f)). \qquad (4)$$

Let us analyze the last summand first. Observe that for each $f \in \widetilde{D}$ and $c \in \mathsf{Close}(f)$, we have

$$\operatorname{dist}(c, x(f)) \leqslant \operatorname{dist}(c, f) + \operatorname{dist}(f, x(f)) \leqslant 2\varepsilon^{2} \cdot \operatorname{avgcost}(f).$$

Thus, using (3) we have

$$\sum_{f \in \widetilde{D}} \sum_{c \in \mathsf{Close}(f)} \operatorname{dist}(c, x(f))$$
$$\leqslant \sum_{f \in \widetilde{D}} |\mathsf{Close}(f)| \cdot 2\varepsilon^{2} \cdot \operatorname{avgcost}(f)$$
$$\leqslant 2\varepsilon^{2} \cdot \sum_{f \in \widetilde{D}} |\operatorname{cluster}(f)| \cdot \operatorname{avgcost}(f)$$
$$= 2\varepsilon^{2} \cdot \operatorname{cost}(\widetilde{D}) \leqslant 2\alpha\varepsilon \cdot \mathsf{OPT}. \qquad (5)$$

We are left with analyzing the middle summand of the right hand side of (4). Observe that we have

$$\sum_{f \in \widetilde{D}} \sum_{c \in \mathsf{Far}(f)} \operatorname{dist}(c, D) = \operatorname{conn}(\mathsf{Far}, D).$$

By Lemma 3 applied to $S = \mathsf{Far}$, we infer that

$$\Psi = \operatorname{conn}(\mathsf{Far}, \widetilde{D}) \leqslant \operatorname{conn}(\mathsf{Far}, D) + \varepsilon \cdot \mathsf{OPT},$$

and thus we have

$$\sum_{f \in \widetilde{D}} \sum_{c \in \mathsf{Far}(f)} \operatorname{dist}(c, D) \geqslant \Psi - \varepsilon \cdot \mathsf{OPT}. \qquad (6)$$

For every $f \in \widetilde{D}$, let $g(f)$ be the facility of D that is closest to f. By Corollary 5 we have that

$$\operatorname{dist}(f, g(f)) \leqslant 2 \cdot \operatorname{avgcost}(f).$$

Now, for every $c \in \mathsf{Far}(f)$ we have

$$
\begin{aligned}
3 \cdot \mathrm{dist}(c, f) &\geqslant 3\varepsilon^{-2} \cdot \mathsf{avgcost}(f) \\
&\geqslant \varepsilon^{-2} \cdot \mathrm{dist}(f, g(f)) + \varepsilon^{-4} \cdot \mathrm{dist}(f, x(f)) \\
&\geqslant \varepsilon^{-2} \cdot \mathrm{dist}(x(f), g(f)) \\
&\geqslant \varepsilon^{-2} \cdot \mathrm{dist}(x(f), D),
\end{aligned}
$$

where in the second step we used $\mathrm{dist}(f, x(f)) = \varepsilon^2 \cdot \mathsf{avgcost}(f)$. Summing this inequality through all $f \in \widetilde{D}$ and $c \in \mathsf{Far}(f)$ we obtain that

$$
\begin{aligned}
\mathsf{conn}(\mathsf{Far}, \widetilde{D}) &= \sum_{f \in \widetilde{D}} \sum_{c \in \mathsf{Far}(f)} \mathrm{dist}(c, f) \\
&\geqslant \frac{\varepsilon^{-2}}{3} \sum_{f \in \widetilde{D}} \sum_{c \in \mathsf{Far}(f)} \mathrm{dist}(x(f), D),
\end{aligned}
$$

which means that

$$
\begin{aligned}
&\sum_{f \in \widetilde{D}} \sum_{c \in \mathsf{Far}(f)} \mathrm{dist}(x(f), D) \\
&\leqslant 3\varepsilon^2 \cdot \mathsf{conn}(\mathsf{Far}, \widetilde{D}) \\
&\leqslant 3\varepsilon^2 \cdot \mathsf{cost}(\widetilde{D}) \leqslant 3\alpha\varepsilon \cdot \mathsf{OPT}. \quad (7)
\end{aligned}
$$

By combining (4), (5), (6), and (7) we infer that

$$
\begin{aligned}
\mathsf{OPT}' &\leqslant \mathsf{cost}(D; I') \\
&= \mathsf{open}(D) + \mathsf{conn}(C', D) \\
&\leqslant \mathsf{open}(D) + \mathsf{conn}(C, D) \\
&\quad +2\alpha\varepsilon \cdot \mathsf{OPT} - \Psi + \varepsilon \cdot \mathsf{OPT} + 3\alpha\varepsilon \cdot \mathsf{OPT} \\
&\leqslant \mathsf{cost}(D; I) + 6\alpha\varepsilon \cdot \mathsf{OPT} - \Psi \\
&= (1 + 6\alpha\varepsilon)\mathsf{OPT} - \Psi.
\end{aligned}
$$

This establishes the first inequality.

For the second inequality, again by triangle inequality we have

$$
\begin{aligned}
\mathsf{conn}(C, R) &\leqslant \mathsf{conn}(C', R) + \sum_{f \in \widetilde{D}} \sum_{c \in \mathsf{Far}(f)} \mathrm{dist}(c, x(f)) \\
&\quad + \sum_{f \in \widetilde{D}} \sum_{c \in \mathsf{Close}(f)} \mathrm{dist}(c, x(f)).
\end{aligned}
$$

The last summand has already been estimated in (5), so we are left with analyzing the middle summand. Observe that for each $f \in \widetilde{D}$ and $c \in \mathsf{Far}(f)$, we have

$$
\begin{aligned}
\mathrm{dist}(c, x(f)) &\leqslant \mathrm{dist}(c, f) + \mathrm{dist}(f, x(f)) \\
&\leqslant (1 + \varepsilon^4) \cdot \mathrm{dist}(c, f),
\end{aligned}
$$

where the last inequality follows from $\mathrm{dist}(c, f) \geqslant \varepsilon^{-2} \cdot \mathsf{avgcost}(f)$ and $\mathrm{dist}(f, x(f)) = \varepsilon^2 \cdot \mathsf{avgcost}(f)$. Thus, we

have

$$
\begin{aligned}
&\sum_{f \in \widetilde{D}} \sum_{c \in \mathsf{Far}(f)} \mathrm{dist}(c, x(f)) \\
&\leqslant (1 + \varepsilon^4) \cdot \sum_{f \in \widetilde{D}} \sum_{c \in \mathsf{Far}(f)} \mathrm{dist}(c, f) \\
&= (1 + \varepsilon^4) \cdot \mathsf{conn}(\mathsf{Far}, \widetilde{D}) \\
&= \Psi + \varepsilon^4 \cdot \mathsf{conn}(\mathsf{Far}, \widetilde{D}) \\
&\leqslant \Psi + \varepsilon^4 \cdot \mathsf{cost}(\widetilde{D}) \\
&\leqslant \Psi + \alpha\varepsilon^3 \cdot \mathsf{OPT}. \quad (8)
\end{aligned}
$$

By combining (8), (5), and (8) we obtain that

$$
\begin{aligned}
\mathsf{cost}(R; I) &= \mathsf{open}(R) + \mathsf{conn}(C, R) \\
&\leqslant \mathsf{open}(R) + \mathsf{conn}(C', R) \\
&\quad +\Psi + 3\alpha\varepsilon \cdot \mathsf{OPT} \\
&= \mathsf{cost}(R; I') + \Psi + 3\alpha\varepsilon \cdot \mathsf{OPT}.
\end{aligned}
$$

This concludes the proof. ∎

Corollary 7. *For any $R \subseteq F$, if*

$$
\mathsf{cost}(R; I') \leqslant (1 + \gamma)\mathsf{OPT}' + \delta,
$$

for some $\gamma, \delta \geqslant 0$, then

$$
\mathsf{cost}(R; I) \leqslant (1 + 2\gamma + 8\alpha\varepsilon)\mathsf{OPT} + \delta.
$$

Proof: First, note that

$$
\mathsf{OPT}' \leqslant (1 + 5\alpha\varepsilon)\mathsf{OPT} - \Psi \leqslant 2 \cdot \mathsf{OPT}.
$$

Then we have

$$
\begin{aligned}
\mathsf{cost}(R; I) &\leqslant \mathsf{cost}(R; I') + \Psi + 3\alpha\varepsilon \cdot \mathsf{OPT} \\
&\leqslant (1 + \gamma)\mathsf{OPT}' + \delta + \Psi + 3\alpha\varepsilon \cdot \mathsf{OPT} \\
&\leqslant \mathsf{OPT}' + 2\gamma\mathsf{OPT} + \delta + \Psi + 3\alpha\varepsilon \cdot \mathsf{OPT} \\
&\leqslant (1 + 5\alpha\varepsilon)\mathsf{OPT} - \Psi + 2\gamma\mathsf{OPT} + \delta \\
&\quad +\Psi + 3\alpha\varepsilon \cdot \mathsf{OPT} \\
&= (1 + 2\gamma + 8\alpha\varepsilon)\mathsf{OPT} + \delta,
\end{aligned}
$$

as claimed. ∎

Thus, by Corollary 7 we may focus on finding a near-optimum solution to instance I' instead of I. The instance I', however, has the following concentration property that will be useful later on: for every $f \in \widetilde{D}$ and $c \in \mathsf{cluster}'(f)$, we have

$$
\varepsilon^2 \cdot \mathsf{avgcost}(f) \leqslant \mathrm{dist}(c, f) \leqslant \varepsilon^{-2} \cdot \mathsf{avgcost}(f).
$$

Finally, we check that solution $\widetilde{D}$ is still not too expensive in the instance I'.

Lemma 8. *For every $f \in \widetilde{D}$ it holds that*

$$
\begin{aligned}
&\mathsf{open}(f) + \sum_{c \in \mathsf{cluster}'(f)} \mathrm{dist}(c, f) \\
&\leqslant (1 + \varepsilon^2) \cdot |\mathsf{cluster}(f)| \cdot \mathsf{avgcost}(f). \quad (9)
\end{aligned}
$$

In total, we have

$$\text{open}(\widetilde{D}) + \sum_{f \in \widetilde{D}} \sum_{c \in \text{cluster}'(f)} \text{dist}(c, f) \leqslant 2\alpha\varepsilon^{-1} \cdot \text{OPT}. \quad (10)$$

Proof: Recall that

$$|\text{cluster}(f)| \cdot \text{avgcost}(f) = \text{open}(f) + \sum_{c \in \text{cluster}(f)} \text{dist}(c, f).$$

Thus, to show (9), it suffices to prove that

$$\sum_{c \in \text{Far}(f) \cup \text{Close}(f)} (\text{dist}(x(f), f) - \text{dist}(c, f)))$$
$$\leqslant \quad \varepsilon^2 |\text{cluster}(f)| \cdot \text{avgcost}(f).$$

For each $c \in \text{Far}(f)$, we have $\text{dist}(x(f), f) = \varepsilon^2 \cdot \text{avgcost}(f)$ and $\text{dist}(c, f) \geqslant \varepsilon^{-2} \cdot \text{avgcost}(f)$, hence $\text{dist}(x(f), f) - \text{dist}(c, f) \leqslant 0$. On the other hand $\text{dist}(x(f), f) = \varepsilon^2 \cdot \text{avgcost}(f)$, hence for each $c \in \text{Close}(f)$ we have $\text{dist}(x(f), f) - \text{dist}(c, f) \leqslant \varepsilon^2 \cdot \text{avgcost}(f)$. This proves (9).

By summing (9) over all $f \in \widetilde{D}$ we obtain that

$$\text{open}(\widetilde{D}) + \sum_{f \in \widetilde{D}} \sum_{c \in \text{cluster}'(f)} \text{dist}(c, f)$$
$$\leqslant \quad (1 + \varepsilon^2) \cdot \text{cost}(\widetilde{D})$$
$$\leqslant \quad 2\alpha\varepsilon^{-1} \cdot \text{OPT},$$

as claimed. ∎

Note that in Lemma 8, the left hand side of (10) is lower bounded by $\text{cost}(\widetilde{D}, I')$, but is not necessarily equal to it, as the clients of each cluster $\text{cluster}'(f)$ are assigned to f, which may cease to be the closest facility after moving a client.

Layering on magnitudes of the average cost: We now work with the instance I'. The goal is to use the obtained properties of clusters to break the instance into several independent ones at the cost of additionally paying εOPT, so that each of the instances concerns only clients from clusters with average cost of the same magnitude. This is because such instances can be solved efficiently using the following crucial lemma, whose proof will be given later.

Lemma 9. *Suppose we are given an instance $J = (G, C, F, \text{open})$ of NON-UNIFORM FACILITY LOCATION where G is planar. Moreover, we are provided a real $r > 1$ and a set of facilities $D^\circ \subseteq F$ such that the clients of C can be partitioned into nonempty clusters $(\text{cluster}(f))_{f \in D^\circ}$ so that the following properties hold for each $f \in D^\circ$:*

- *$1 \leqslant \text{dist}(c, f) \leqslant r$ for each $c \in \text{cluster}(f)$; and*
- *$\text{open}(f) + \sum_{c \in \text{cluster}(f)} \text{dist}(c, f) \leqslant |\text{cluster}(f)| \cdot r$.*

Then, given $\varepsilon > 0$, one can in time $n^{\mathcal{O}(\varepsilon^{-2}r)}$ compute a solution to J with cost at most $(1 + \varepsilon)\text{OPT}(J) + \varepsilon \cdot M$, where $M = \text{open}(D^\circ) + \sum_{f \in D^\circ} \sum_{c \in \text{cluster}(f)} \text{dist}(c, f)$.

Breaking into separate instances that can be treated using Lemma 9 will be done using layering on the levels of magnitude of average costs of facilities from $\widetilde{D}$. While the layering itself will be quite standard, the proof of the separation property between the instances will be quite non-trivial and will require the fine understanding of properties of $\widetilde{D}$ that we have developed.

Let us partition the facilities of $\widetilde{D}$ into layers $(L_i)_{i \in \mathbb{Z}}$, where L_i comprises facilities $f \in \widetilde{D}$ satisfying

$$\varepsilon^{4i} \geqslant \text{avgcost}(f) > \varepsilon^{4i+4}.$$

For $i \in \mathbb{Z}$, let

$$\ell_i = \sum_{f \in L_i} \left(\text{open}(f) + \sum_{c \in \text{cluster}'(f)} \text{dist}(c, f) \right).$$

By Lemma 8, we have

$$\sum_{i \in \mathbb{Z}} \ell_i \leqslant 2\alpha\varepsilon^{-1} \cdot \text{OPT}. \quad (11)$$

Let $q = \lceil \varepsilon^{-2} \rceil$. Pick $a \in \{0, 1, \dots, q-1\}$ such that $\sum_{i:\, i \equiv a \bmod q} \ell_i$ is minimum. Then by (8) and the fact that $q \geqslant \varepsilon^{-2}$ we infer that

$$\sum_{i:\, i \equiv a \bmod q} \ell_i \leqslant \varepsilon^2 \cdot \text{cost}(\widetilde{D}; I') \leqslant 2\alpha\varepsilon \cdot \text{OPT}. \quad (12)$$

Now, define

$$S = \bigcup_{i:\, i \equiv a \bmod q} L_i$$

and

$$W_j = \bigcup_{jq+a < i < (j+1)q+a} L_i \quad \text{for } j \in \mathbb{Z}.$$

Set W_j will be called the *j-ring*. It follows that S and $(W_j)_{j \in \mathbb{Z}}$ form a partition of $\widetilde{D}$.

Intuitively, the idea is to construct a near optimum solution by buying all the facilities of S and using them to serve all clients served by them in $\widetilde{D}$ (the cost of this is bounded by $2\alpha\varepsilon \cdot \text{OPT}$ by (12)), and constructing an instance for each nonempty ring W_j that is subsequently approximated using Lemma 9. However, we need to prepare those instances carefully so that they can be solved separately.

To this end, we heavily rely on Lemma 4 that more or less says that one needs to open a facility within $2 \cdot \text{avgcost}(f)$ of f for every $f \in \widetilde{D}$. This, together with the exponential scale of average costs, implies that while focusing on the ring W_j we do not need to understand how the solution to rings $W_{j'}$ for $j' > j$ looks like (namely, what are the precise locations of the facilities); instead, we just put one zero-cost facility at every $f \in W_{j'}$ that mimics the closest opened facility, this will be satisfying up to losing a factor $(1 + \varepsilon)$.

Let us now proceed with formal details. Denote

$$C_S = \bigcup_{f \in S} \text{cluster}'(f).$$

For every $j \in \mathbb{Z}$ we create the following instance $J_j = (G, C_j, F_j, \text{open}_j)$:

- The graph G is the graph from the original instance;
- $C_j = \bigcup_{f \in W_j} \mathsf{cluster}'(f)$, that is, all clients in clusters of facilities from the ring W_j;
- $F_j = F$ are all facilities from the input;
- $\mathsf{open}_j(f) = 0$ for every $f \in W_{j'}$ with $j' > j$ and every $f \in S$, and $\mathsf{open}_j(f) = \mathsf{open}(f)$ otherwise.

Note that the sets $(C_j)_{j \in \mathbb{Z}}$ are pairwise disjoint and together with C_S form a partition C. For every $j \in \mathbb{Z}$ let $D_j^{\mathrm{free}} = S \cup \bigcup_{j' > j} W_{j'}$ be the set of facilities f with $\mathsf{open}_j(f)$ redefined to 0 in the definition of J_j.

Observe also that if $W_j = \emptyset$, then $C_j = \emptyset$: the instance is trivial and it admits the empty set as the optimum solution. The algorithm does not really need to construct these instances (and thus in fact constructs at most n instances J_j), but we prefer to define them for the sake of clarity of notation. We henceforth call the instances J_j *trivial* if $W_j = \emptyset$ and *nontrivial* otherwise.

We now verify that it suffices to solve each instance J_j separately. This is done through two lemmas. In the first one, we show how to combine solutions to the instances J_j into a solution to the instance I'.

Lemma 10. *Assume we are given sets $D_j \subseteq F_j$ for every nontrival instance J_j. Then one can construct in polynomial time a set $R \subseteq F$ such that*

$$\mathsf{cost}(R; I') \leqslant \sum_j \mathsf{cost}(D_j; J_j) + 10\alpha\varepsilon \cdot \mathsf{OPT}. \quad (13)$$

Proof: For every nontrivial instance J_j and for every $f \in F_j \setminus D_j$ we check whether opening f would not increase the cost of D_j in J_j; if this is the case, we add f to D_j. We also add D_j^{free} to D_j as it does not increase the cost of D_j. Henceforth we assume that for every nontrivial instance J_j and every $f \in F_j \setminus D_j$ it holds that

$$\mathsf{cost}(D_j \cup \{f\}; J_j) \geqslant \mathsf{cost}(D_j; J_j). \quad (14)$$

We define $D_j = D_j^{\mathrm{free}}$ for every trivial instance J_j. Note that property (14) also holds for the trivial instances. Let $D_j' = D_j \setminus D_j^{\mathrm{free}}$ for every $j \in \mathbb{Z}$; note that $D_j' = \emptyset$ for trivial J_j. Let

$$R := S \cup \bigcup_{j \in \mathbb{Z}} D_j'.$$

We claim that R satisfies the requirements of the lemma; it is clearly computable in polynomial time as $D_j' = \emptyset$ for trivial J_j. Note that $D_j \setminus D_j' = D_j^{\mathrm{free}}$ for every $j \in \mathbb{Z}$.

For a facility $f \in D_j$, let $\mathsf{cluster}(f, D_j; J_j) \subseteq C_j$ be the set of clients served by f in the solution D_j to J_j; that is, $\mathsf{cluster}(f, D_j; J_j)$ is the set of these $c \in C_j$ for which f is the closest facility from D_j. Consider redirecting, in the solution R to the instance I', all clients from $\mathsf{cluster}'(f)$ to

f, for every $f \in S \subseteq R$. Then we have:

$$\mathsf{cost}(R; I') \leqslant \left(\mathsf{open}(S) + \sum_{f \in S} \sum_{c \in \mathsf{cluster}'(f)} \mathrm{dist}(c, f) \right)$$
$$+ \left(\mathsf{open}\left(\bigcup_{j \in \mathbb{Z}} D_j' \right) \right.$$
$$+ \sum_{j \in \mathbb{Z}} \sum_{f \in D_j'} \sum_{c \in \mathsf{cluster}(f, D_j; J_j)} \mathrm{dist}(c, R) \right)$$
$$+ \left(\sum_{j \in \mathbb{Z}} \sum_{f \in D_j^{\mathrm{free}}} \sum_{c \in \mathsf{cluster}(f, D_j; J_j)} \mathrm{dist}(c, R) \right).$$

We bound the three summands in the inequality above separatedly. By (12), the first summand is bounded by $2\alpha\varepsilon\mathsf{OPT}$. Since $D_j' \subseteq R \cap D_j$ for every $j \in \mathbb{Z}$, we have for the second summand:

$$\mathsf{open}\left(\bigcup_{j \in \mathbb{Z}} D_j' \right) + \sum_{j \in \mathbb{Z}} \sum_{f \in D_j'} \sum_{c \in \mathsf{cluster}(f, D_j; J_j)} \mathrm{dist}(c, R)$$
$$\leqslant \sum_{j \in \mathbb{Z}} \left(\mathsf{open}(D_j) + \sum_{f \in D_j'} \sum_{c \in \mathsf{cluster}(f, D_j; J_j)} \mathrm{dist}(c, f) \right)$$
$$= \sum_{j \in \mathbb{Z}} \left(\mathsf{open}(D_j) \right.$$
$$+ \left. \mathsf{conn}\left(\bigcup_{f \in D_j'} \mathsf{cluster}(f, D_j; J_j), D_j; J_j \right) \right).$$

We now estimate the third summand. Consider a nontrivial instance J_j and a facility $f \in W_j$. Recall that $\mathsf{cluster}'(f) \subseteq C_j$. By applying Lemma 4 to the instance J_j, solution D_j, facility f, and set $K = \mathsf{cluster}'(f)$ we infer that (14) ensures that there exists $g \in D_j$ with

$$\mathrm{dist}(f, g) \leqslant 2 \cdot \frac{\mathsf{open}(f) + \sum_{c \in \mathsf{cluster}'(f)} \mathrm{dist}(c, f)}{|\mathsf{cluster}'(f)|}.$$

Plugging now the bound of Lemma 8, we obtain

$$\begin{aligned} \mathrm{dist}(f, D_j) &\leqslant 2(1 + \varepsilon^2) \cdot \mathsf{avgcost}(f) \\ &\leqslant 4 \cdot \mathsf{avgcost}(f) \\ &\leqslant 4\varepsilon^{4(jq+a+1)}. \end{aligned} \quad (15)$$

We now observe the following.

Claim 1. *For every facility $f \in D_j$, we have*

$$\mathrm{dist}(f, R) \leqslant 4 \sum_{j'=j+1}^{\infty} \varepsilon^{4(j'q+a+1)}.$$

PROOF. Since all but a finite number of D_j-s are empty, we can proceed by induction on j, assuming the claim holds for

all $j' > j$. Take any $f \in D_j$. If $f \in R$ then $\mathrm{dist}(f, R) = 0$ and we are done. Otherwise, $f \in D_j \setminus R \subseteq \bigcup_{j' > j} D_{j'}$, so $f \in D_{j'}$ for some $j' > j$. By (15), there exists $g \in D_{j'}$ such that

$$\mathrm{dist}(f, g) \leqslant 4\varepsilon^{4(j'q+a+1)}.$$

By induction assumption for g, we have

$$\mathrm{dist}(g, R) \leqslant 4 \sum_{j''=j'+1}^{\infty} \varepsilon^{4(j''q+a+1)}.$$

Hence, we have

$$
\begin{aligned}
\mathrm{dist}(f, R) &\leqslant \mathrm{dist}(f, g) + \mathrm{dist}(g, R) \\
&\leqslant 4\varepsilon^{4(j'q+a+1)} + 4 \sum_{j''=j'+1}^{\infty} \varepsilon^{4(j''q+a+1)} \\
&\leqslant 4 \sum_{j'=j+1}^{\infty} \varepsilon^{4(j'q+a+1)},
\end{aligned}
$$

as required. ⌟

By Claim 1, for every $f \in D_j^{\mathrm{free}}$ and $c \in \mathrm{cluster}(f, D_j; J_j)$ with $c \in \mathrm{cluster}'(f_c)$ for some $f_c \in W_j$ we have the following:

$$
\begin{aligned}
\mathrm{dist}(c, R) &\leqslant \mathrm{dist}(c, f) + \mathrm{dist}(f, R) \\
&\leqslant \mathrm{dist}(c, f) + \sum_{j'=j+1}^{\infty} 4\varepsilon^{4(j'q+a+1)} \\
&\leqslant \mathrm{dist}(c, f) + \varepsilon^{4((j+1)q+a+1)} \cdot \frac{4}{1 - \varepsilon^{4q}} \\
&\leqslant \mathrm{dist}(c, f) + 8\varepsilon^4 \cdot \mathrm{avgcost}(f_c).
\end{aligned}
$$

By summing the above bound through all $j \in \mathbb{Z}$ and $f \in D_j^{\mathrm{free}}$ we obtain

$$
\begin{aligned}
&\sum_{j \in \mathbb{Z}} \sum_{f \in D_j^{\mathrm{free}}} \sum_{c \in \mathrm{cluster}(f, D_j; J_j)} \mathrm{dist}(c, R) \\
&\leqslant \sum_{j \in \mathbb{Z}} \mathrm{conn}\left(\bigcup_{f \in D_j^{\mathrm{free}}} \mathrm{cluster}(f, D_j; J_j), D_j; J_j \right) \\
&\quad + 8\varepsilon^4 \cdot \mathrm{cost}(\widetilde{D}).
\end{aligned}
$$

Since $\mathrm{cost}(\widetilde{D}) \leqslant \alpha\varepsilon^{-1} \cdot \mathsf{OPT}$, we can combine the obtained bounds as follows:

$$
\begin{aligned}
&\mathrm{cost}(R; I') \\
&\leqslant 2\alpha\varepsilon\mathsf{OPT} \\
&\quad + \sum_{j \in \mathbb{Z}} \Bigg(\mathrm{open}(D_j) \\
&\qquad + \mathrm{conn}\left(\bigcup_{f \in D_j'} \mathrm{cluster}(f, D_j; J_j), D_j; J_j \right) \Bigg) \\
&\quad + \sum_{j \in \mathbb{Z}} \mathrm{conn}\left(\bigcup_{f \in D_j^{\mathrm{free}}} \mathrm{cluster}(f, D_j; J_j), D_j; J_j \right) \\
&\quad + 8\alpha\varepsilon^3\mathsf{OPT} \\
&= \sum_{j} \mathrm{cost}(D_j; J_j) + 2\alpha\varepsilon\mathsf{OPT} + 8\alpha\varepsilon^3\mathsf{OPT} \\
&\leqslant \sum_{j} \mathrm{cost}(D_j; J_j) + 10\alpha\varepsilon\mathsf{OPT}.
\end{aligned}
$$

This concludes the proof. ∎

The second lemma shows that optima in instances J_j almost partition the optimum in I.

Lemma 11. *For $j \in \mathbb{Z}$, let OPT_j be the cost of the optimum solution of J_j. Then*

$$\sum_{j \in \mathbb{Z}} \mathsf{OPT}_j \leqslant (1 + 9\alpha\varepsilon) \cdot \mathsf{OPT}.$$

Proof: Let D' be an optimum solution to I'. For every $f \in D'$ let $j(f)$ be the maximum value of j such that there exists $g \in W_j$ with $\mathrm{dist}(f, g) \leqslant 3\varepsilon^{-2} \cdot \mathrm{avgcost}(g)$. If no such j exists, we set $j(f)$ to be the minimum value of j for which J_j is nontrivial. For every $j \in \mathbb{Z}$ we define

$$D_j' = \{f \in D' \mid j(f) = j\} \qquad \text{and} \qquad D_j = D_j' \cup D_j^{\mathrm{free}};$$

note that $D_j' = \emptyset$ for trivial J_j. Our goal is to estimate $\sum_{j \in \mathbb{Z}} \mathrm{cost}(D_j; J_j)$ by $\mathrm{cost}(D', I')$ plus some terms of the order of $\varepsilon \cdot \mathsf{OPT}$. First, it is immediate from the definition that $\mathrm{open}(D') = \sum_{j \in \mathbb{Z}} \mathrm{open}_j(D_j)$. Clearly, for trivial J_j we have $D_j = D_j^{\mathrm{free}}$ and $\mathrm{cost}(D_j; J_j) = 0$. Let J_j be nontrivial. Consider a client $c \in C_j$; by the definition of J_j, there exists $f_0 \in W_j$ with $c \in \mathrm{cluster}'(f_0)$.

Let $f \in D'$ be the facility that serves c in the solution D', that is, $\mathrm{dist}(c, f) = \mathrm{dist}(c, D')$. We consider cases depending on the relation of $j(f)$ and j.

Case 1: $j(f) > j$. By the definition of $j(f)$, there exists $g \in W_{f(j)} \subseteq D_j^{\mathrm{free}}$ with $\mathrm{dist}(f, g) \leqslant 3\varepsilon^{-2} \cdot \mathrm{avgcost}(g) \leqslant 3\varepsilon^2 \cdot \mathrm{avgcost}(f_0)$. Therefore

$$
\begin{aligned}
\mathrm{dist}(c, D_j) &\leqslant \mathrm{dist}(c, g) \\
&\leqslant \mathrm{dist}(c, f) + 3\varepsilon^2 \cdot \mathrm{avgcost}(f_0) \\
&= \mathrm{dist}(c, D') + 3\varepsilon^2 \cdot \mathrm{avgcost}(f_0).
\end{aligned}
$$

Case 2: $j(f) = j$. Here $f \in D_j$ and thus

$$\text{dist}(c, D_j) \leqslant \text{dist}(c, f) = \text{dist}(c, D').$$

Case 3: $j(f) < j$. Supposing that $f_0 \notin D'$, Lemma 4 applied to the (optimal) solution D' in I' with facility f_0 and $K = \text{cluster}'(f_0)$ yields that there exists $g_0 \in D'$ with

$$
\begin{aligned}
\text{dist}(f_0, g_0) &\leqslant 2 \cdot \frac{\text{open}(f_0) + \sum_{c \in \text{cluster}'(f_0)} \text{dist}(c, f_0)}{|\text{cluster}'(f_0)|} \\
&\leqslant 2(1 + \varepsilon^2) \cdot \text{avgcost}(f_0) \leqslant 4 \cdot \text{avgcost}(f_0).
\end{aligned}
$$

Here, the penultimate inequality follows from Lemma 8. If $f_0 \in D'$, then we can take $g_0 = f_0$ and the above inequality also holds.

By the definition of $j(f)$ we have that $\text{dist}(f, f_0) > 3\varepsilon^{-2} \cdot \text{avgcost}(f_0)$. On the other hand, $\text{dist}(c, f_0) \leqslant \varepsilon^{-2} \cdot \text{avgcost}(f_0)$ while $\text{dist}(f_0, g_0) \leqslant 4 \cdot \text{avgcost}(f_0) \leqslant \varepsilon^{-2} \cdot \text{avgcost}(f_0)$. Since $g_0 \in D'$, we infer that f is not the closest to c facility of D', a contradiction. We infer that this case is impossible.

We conclude that in any case, we have

$$\text{dist}(c, D_j) \leqslant \text{dist}(c, D') + 3\varepsilon^2 \cdot \text{avgcost}(f_0).$$

By summing this bound through all the clients and adding opening costs to both sides, we obtain

$$
\begin{aligned}
\sum_{j \in \mathbb{Z}} \text{cost}(D_j; J_j) &\leqslant \text{cost}(D'; I') + 3\varepsilon^2 \cdot \text{cost}(\widetilde{D}) \\
&\leqslant \text{OPT}' + 3\alpha\varepsilon \cdot \text{OPT} \\
&\leqslant (1 + 9\alpha\varepsilon)\text{OPT},
\end{aligned}
$$

where in the last inequality we use Lemma 6. This finishes the proof of the lemma. ∎

We conclude this section with the observation that it remains to prove Lemma 9 in order to show a polynomial-time approximation scheme for NON-UNIFORM FACILITY LOCATION in planar graphs. After initial preprocessing of the input instance I, Corollary 7 asserts that it suffices to find a $(1 + \mathcal{O}(\varepsilon))$-approximate solution to I'.

To this end, we break I' into instances $(J_j)_{j \in \mathbb{Z}}$. For every nontrivial J_j, we scale all the edge lengths and opening costs of J_j by a factor of $\varepsilon^{-(4(jq+q+a)+2)}$ and define $D^\circ = W_j$ and $\text{cluster}(f) := \text{cluster}'(f)$ for every $f \in D^\circ$. Note that $(\text{cluster}(f))_{f \in D^\circ}$ partitions C_j. Let

$$r = 2\varepsilon^{-4q} \leqslant 2\varepsilon^{-4\varepsilon^{-2}}.$$

Then, since for every $f \in W_j$ we have

$$\varepsilon^{4(jq+a+1)} \geqslant \text{avgcost}(f) > \varepsilon^{4(jq+q+a)} \qquad (16)$$

and for every $c \in \text{cluster}'(f)$ it holds that

$$\varepsilon^2 \cdot \text{avgcost}(f) \leqslant \text{dist}(c, f) \leqslant \varepsilon^{-2} \cdot \text{avgcost}(f),$$

we infer that after scaling the distances, $1 \leqslant \text{dist}(c, f) \leqslant r/2$ for every $f \in W_j$ and $c \in \text{cluster}'(f)$. Furthermore, (16)

together with Lemma 8 imply the second condition of Lemma 9.

Consequently, the algorithm of Lemma 9 applied to J_j prepared as above with accuracy parameter ε^2 (instead of ε) runs in time $n^{2^{\mathcal{O}(\varepsilon^{-2} \log(1/\varepsilon))}}$ and returns a solution D_j of cost (after scaling back again all the edge weights and opening costs) satisfying

$$\text{cost}(D_j; J_j) \leqslant (1 + \varepsilon^2)\text{OPT}_j + \varepsilon^2 \cdot M_j,$$

where

$$M_j = \text{open}(W_j) + \sum_{f \in W_j} \sum_{c \in \text{cluster}'(f)} \text{dist}(c, f).$$

Observe that

$$\sum_{j \in \mathbb{Z}} M_j \leqslant \text{cost}(\widetilde{D}) \leqslant 2\alpha\varepsilon^{-1}\text{OPT}.$$

Thus Lemma 10 allows us to combine the solutions D_j into a solution R to I' of cost satisfying:

$$\text{cost}(R; I') \leqslant (1 + \varepsilon^2)\sum_{j \in \mathbb{Z}} \text{OPT}_j + 12\alpha\varepsilon \cdot \text{OPT}.$$

By Lemma 11, this value is at most

$$(1 + \varepsilon^2)\text{OPT}' + 18\alpha\varepsilon \cdot \text{OPT}.$$

Finally, we may apply Corollary 7 to conclude that

$$
\begin{aligned}
\text{cost}(R; I) &\leqslant (1 + 2\varepsilon^2 + 8\alpha\varepsilon)\text{OPT} + 18\alpha\varepsilon \cdot \text{OPT} \\
&\leqslant (1 + 28\alpha\varepsilon) \cdot \text{OPT},
\end{aligned}
$$

as required. Consequently, it remains to prove Lemma 9.

III. DYNAMIC PROGRAMMING ALGORITHM

A. Overview

Before we proceed to the formal proof of Lemma 9, we give a short overview. The approach is based on a rather standard layering argument plus portal-based Divide&Conquer. While the formal reasoning is quite lengthy due to a number of technical details that require attention, we hope that presenting an intuitive description of consecutive steps will help the reader with guiding through the proof.

Suppose D is an optimum solution to instance J. The first observation is that D enjoys a similar proximity property as expressed in Lemma 4. Namely, every client $c \in C$ is at distance at most $3r$ from some facility of D. The argument is essentially the same: if all clients from some cluster $\text{cluster}(f)$ for $f \in D^\circ$ had connection costs larger than r in the solution D, one could improve D by opening facility f and redirecting all clients from $\text{cluster}(f)$ to f. Otherwise, some client from $\text{cluster}(f)$ is within distance at most r from D, which implies that all of them are at distance at most $3r$.

This proximity property allows us to apply standard layering. We fix a vertex s and classify facilities from D°

of the graph into layers $(D_i^\circ)_{i\in\mathbb{N}}$ of width $8r$ according to distances from s: layer D_i° comprises facilities $f \in D^\circ$ satisfying $i \cdot 8r \leqslant \mathrm{dist}(s,f) < (i+1) \cdot 8r$. With every facility $f \in D^\circ$ we can associate its contribution to M, equal to $\mathsf{open}(f) + \sum_{c\in\mathsf{cluster}(f)} \mathrm{dist}(c,f)$. Now, denoting $q = \lceil \varepsilon^{-1} \rceil$, there exists $a \in \{0, 1, \ldots, q-1\}$ such that the total contribution of facilities from layers D_i° with $i \equiv a \bmod q$ is at most εM. Hence, by paying cost εM we may open these facilities and direct all clients from their clusters to them. Now it is easy to see that we have a separation property: instance J can be decomposed into instances $(J_j)_{j\in\mathbb{N}}$, where J_j concerns connecting all clients from clusters of facilities of $\bigcup_{jq+a<i<(j+1)q+a} D_i^\circ$ to facilities within distance between $(jq+a)\cdot 8r - 4r$ and $((j+1)q+a)\cdot 8r - 4r$ from s, which can be (approximately) solved separately. This is because in the optimum solution, no client-facility path used for connection crosses any of the entirely bought layers due to having length at most $3r$.

Let us focus on one instance J_j. We may contract all vertices at distance less than $(jq+a)\cdot 8r - 8r$ onto s and remove all vertices at distance more than $((j+1)q+a)\cdot 8r$, as these vertices anyway will not participate in any shortest path used by an optimum solution. Thus, we essentially achieve a small radius property in J_j: one may assume that all vertices are at distance at most $8qr = \mathcal{O}(\varepsilon^{-1}r)$ from s.

The idea is to compute a near-optimum solution to J_j using Divide&Conquer on balanced separators, presented as dynamic programming. Using standard separation properties of planar graphs one can show that the graph (or rather its plane embedding) admits a hierarchical decomposition into regions so that the decomposition has depth at most $\log n$ and every region is boundaried by a union of at most 6 shortest paths, all with one endpoint in s. Thus, each of these paths has length $\mathcal{O}(\varepsilon^{-1}r)$. We apply dynamic programming over this decomposition, where we put portals on the boudaries of regions to limit the number of states. That is, along each path we put portals spaced at δ, for some parameter $\delta > 0$, and we allow paths connecting clients with facilities to cross region boundaries only through portals. Since the decomposition has depth $\log n$, each connection path in the optimum solution can be "snapped to portals" to conform with this requirement by using at most $2\log n$ snappings, incurring a total additional cost of $2\delta \cdot \log n$. Therefore, we put $\delta = \varepsilon/\log n$ so that this error is bounded by $\mathcal{O}(\varepsilon)$, which summed through all clients yields an $\mathcal{O}(\varepsilon M)$ error term in total. Thus, the total number of portals on the boundary of each region is $\mathcal{O}(\delta^{-1}\varepsilon^{-1}r) = \mathcal{O}(\varepsilon^{-2}r\log n)$.

In the dynamic programming state associated with a region R, we are concerned about opening facilities within R to serve all clients in R. However, on the boundary of R we have $\mathcal{O}(\varepsilon^{-2}r\log n)$ portals that carry information about the assumed interaction between the parts of the overall solution within R and outside of R. For every portal π, this information consists of two pieces:

- *request* $\mathsf{req}(\pi)$ that gives a hard request on the sought solution within R: there has to be a facility opened at distance at most $\mathsf{req}(\pi)$ from π;
- *prediction* $\mathsf{pred}(\pi)$ that gives a possibility of connecting clients to portals: every client c can be connected to π at connection cost $\mathrm{dist}(c,\pi) + \mathsf{pred}(\pi)$.

Intuitively, predictions represent "virtual" opened facilities residing outside of R, which can be accessed at an additional cost given by $\mathsf{pred}(\pi)$, while by satisfying requests we make sure that predictions in other regions can be fulfilled. Since all client-facility paths in the optimum solution are of length at most $3r$, we may assume that all requests and predictions in all considered states are bounded by $3r$. At the cost of an additional error term $\mathcal{O}(\varepsilon M)$ we can also assume that requests and predictions are rounded to integer multiples of δ. Thus, for every portal π we can limit ourselves to $\mathcal{O}(\delta^{-1}r) = \mathcal{O}(\varepsilon^{-2}r\log n)$ possibilities for $\mathsf{req}(\pi)$ and same for $\mathsf{pred}(\pi)$.

Let us estimate the number of states constructed so far. For each of $\mathcal{O}(\varepsilon^{-2}\log n)$ portals on the boundary of R we have $\mathcal{O}(\varepsilon^{-2}r\log n)$ possibilities for $\mathsf{req}(\pi)$ and for $\mathsf{pred}(\pi)$, yielding a total number of states being $(\varepsilon^{-2}r\log n)^{\mathcal{O}(\varepsilon^{-2}r\log n)} = n^{\mathrm{poly}(1/\varepsilon)\cdot r\cdot\log\log n}$, which is quasi-polynomial. As transitions in this dynamic programming can be implemented efficiently, this already yields a QPTAS, and we are left with reducing the number of states to polynomial.

The final trick is to take a closer look at what we store in the states. Since $\mathsf{req}(\cdot)$ stores the requested distance to the closest facility opened within R, it is safe to assume that $\mathsf{req}(\cdot)$ (before rounding to integer multiples of δ) will be 1-Lipschitz in the following sense: for any two portals π, ρ, we have

$$|\mathsf{req}(\pi) - \mathsf{req}(\rho)| \leqslant \mathrm{dist}(\pi,\rho).$$

An analogous reasoning can be applied to predictions, so we can assume that $\mathsf{pred}(\cdot)$ is 1-Lipschitz as well. Now consider any of the 6 shortest paths comprising the boundary of R, say P. On this path we put portals spaced at δ, say $\pi_1, \ldots, \pi_\ell$ for $\ell \leqslant \mathcal{O}(\varepsilon^{-2}r\log n)$ in the order on P. As argued, after rounding we have $\mathcal{O}(\varepsilon^{-2}\log n)$ possibilities for $\mathsf{req}(\pi_1)$, but observe that once (rounded) $\mathsf{req}(\pi_{i-1})$ is chosen, there are only at most 5 possibilites for $\mathsf{req}(\pi_i)$: it must be an integer multiple of δ that differs from $\mathsf{req}(\pi_{i-1})$ by at most 2δ, due to $\mathrm{dist}(\pi_{i-1}, \pi_i) = \delta$. Hence, the total number of choices for the values of requests along P is bounded by $\mathcal{O}(\varepsilon^{-2}\log n) \cdot 5^{\mathcal{O}(\varepsilon^{-2}r\log n)} = n^{\mathcal{O}(\varepsilon^{-2}r)}$. Same argument applies to predictions, and as the boundary of R consists of at most 6 such paths, the total number of states we need to consider is $n^{\mathcal{O}(\varepsilon^{-2}r)}$.

B. Proof of Lemma 9

We now proceed with the formal proof of Lemma 9. For the remainder of this section, let us fix the setting and the notation from the statement of Lemma 9.

Fix an optimum solution $D \subseteq F$ in the instance J. We first prove that in fact, every client is not too far from its closest facility in D.

Lemma 12. *For each $c \in C$ there exists $g \in D$ such that* $\mathrm{dist}(c, g) \leqslant 3r$.

Proof: Let $f \in D^\circ$ be such that $c \in \mathsf{cluster}(f)$; then $\mathrm{dist}(c, f) \leqslant r$. We shall prove that there exists some client $d \in \mathsf{cluster}(f)$ and facility $g \in D$ such that $\mathrm{dist}(d, g) \leqslant r$. Indeed, if this is true, then we have $\mathrm{dist}(c, g) \leqslant \mathrm{dist}(c, f) + \mathrm{dist}(f, d) + \mathrm{dist}(d, g) \leqslant r + r + r = 3r$, as required.

Suppose otherwise: for each $d \in \mathsf{cluster}(f)$, the distance from d to the closest facility from D is larger than r. As $\mathsf{cluster}(f)$ is nonempty, the total connection cost incurred by clients from $\mathsf{cluster}(f)$ in solution D can be lower bounded as follows:

$$\sum_{c \in \mathsf{cluster}(f)} \mathrm{dist}(c, D)$$
$$> |\mathsf{cluster}(f)| \cdot r \geqslant \mathsf{open}(f) + \sum_{c \in \mathsf{cluster}(f)} \mathrm{dist}(c, f).$$

This means that the solution $D \cup \{f\}$ has a strictly smaller cost than D, which contradicts the optimality of D. ∎

Let G' be the subgraph of G induced by all vertices whose distance from D° is at most $4r$. Observe that all clients of C are placed at vertices of G'. Lemma 12 now immediately implies the following.

Lemma 13. *It holds that $D \subseteq V(G')$ and for every $c \in C$ we have $\mathrm{dist}_{G'}(c, D) = \mathrm{dist}_G(c, D)$.*

Proof: For the first assertion, by the optimality of D, for every $g \in D$ there exists some client $c \in C$ such that g is the facility of D closest to c. By Lemma 12 we have $\mathrm{dist}_G(c, g) \leqslant 3r$. If now $f \in D^\circ$ is such that $c \in \mathsf{cluster}(f)$, then $\mathrm{dist}_G(c, f) \leqslant r$. Hence $\mathrm{dist}_G(f, g) \leqslant r + 3r = 4r$, so $g \in V(G')$.

For the second assertion, observe that by Lemma 12, for every client $c \in C$, the shortest path from c to a facility of D traverses only vertices that are at distance at most $4r$ from the facility $f \in D^\circ$ satisfying $c \in \mathsf{cluster}'(f)$. It follows that the distance from c to D is the same in G as in G'. ∎

Let F' consist of all the facilities that are placed at vertices of G', and let $J' = (G', C, F', \mathsf{open})$. We observe that Lemma 12 implies that we can work with instance J' instead of J.

Corollary 14. *For every $R \subseteq F'$, we have $\mathsf{cost}(R; J') \geqslant \mathsf{cost}(R; J)$. Moreover, we have $\mathsf{cost}(D; J') = \mathsf{cost}(D; J)$ and consequently $\mathsf{OPT}(J') = \mathsf{OPT}(J)$.*

Proof: The first assertion is straightforward, because G' is an induced subgraph of G, hence distances between vertices of G' are not smaller in G' than in G. For the second assertion, observe that by Lemma 12 we have $D \subseteq F'$ and

$\mathrm{dist}_{G'}(c, D) = \mathrm{dist}_G(c, D)$ for every client $c \in C$, hence the connection cost of D in J and in J' are the same. As the opening costs are also obviously the same, we conclude that indeed $\mathsf{cost}(D; J') = \mathsf{cost}(D, J)$. This, together with the first assertion, immediately entails $\mathsf{OPT}(J') = \mathsf{OPT}(J)$. ∎

From now on we will assume that the graph G' is connected. This can be achieved either by connecting the connected components using edges of very large (but finite) weight, or applying the forthcoming reasoning to every connected component of G' separately and taking the union of obtained solutions.

Fix any vertex s and partition the vertices of G' into layers $(\mathsf{layer}_i)_{i \in \mathbb{N}}$ as follows: for $i \in \mathbb{N}$ we set:

$$\mathsf{layer}_i = \{v \in V(G') : i \cdot 8r \leqslant \mathrm{dist}(u, s) < (i + 1)8r\}.$$

Let $D_i^\circ = D^\circ \cap \mathsf{layer}_i$. Denote $q = \lceil \varepsilon^{-1} \rceil$. Since $(D_i^\circ)_{i \in \mathbb{N}}$ is a partition of D°, it follows that there exists $a \in \{0, 1, \ldots, q-1\}$ such that denoting $S = \bigcup_{i : \ i \equiv a \bmod q} D_i^\circ$, we have

$$\sum_{f \in S} \left(\mathsf{open}(f) + \sum_{c \in \mathsf{cluster}(f)} \mathrm{dist}(c, f) \right)$$
$$\leqslant \varepsilon \cdot \sum_{f \in D^\circ} \left(\mathsf{open}(f) + \sum_{c \in \mathsf{cluster}(f)} \mathrm{dist}(c, f) \right)$$
$$= \varepsilon \cdot M. \tag{17}$$

Moreover, obviously such a can be found in polynomial time. For $j \in \mathbb{N}$, define the *j-th ring* as

$$W_j = \bigcup_{jq+a < i < (j+1)q+a} \mathsf{layer}_i.$$

For future reference, we note that rings are separated from each other.

Lemma 15. *For any different $j, j' \in \mathbb{N}$ and $u \in W_j$ and $u' \in W_{j'}$, we have $\mathrm{dist}_{G'}(u, u') > 8r$.*

Proof: By the definition of W_j and $W_{j'}$ and since $j \neq j'$, we have $|\mathrm{dist}_{G'}(u, s) - \mathrm{dist}_{G'}(u', s)| > 8r$. Then the statement follows by triangle inequality. ∎

The idea now is to buy the facilities of S and connect the clients from $C_S = \bigcup_{f \in S} \mathsf{cluster}(f)$ to the centers of their clusters — which incurs cost at most $\varepsilon \cdot M$ by (17) — and to construct a separate instance for each ring W_j so that these instances can be solved independently. We now carefully define those instances.

Fix $j \in \mathbb{N}$ and construct graph H_j obtained from G' in the following manner:

1) Remove all vertices w of G' satisfying $w \in \bigcup_{\iota > jq+a} L_\iota$.
2) Contract all vertices w of G' satisfying $w \in \bigcup_{\iota < (j-1)q+a} L_\iota$ onto s; we shall use the name s also for the vertex obtained as the result of this contraction.

3) For every vertex w that, after the contraction explained above, becomes a neighbor of s, we assign the edge sw weight $\text{dist}_{G'}(s, w)$.

Note that in the second, the set of vertices w contracted onto s induces a connected subgraph of G', and thus the contraction is well-defined and preserves the planarity. We shall identify vertices of H_j with their origins in G' in the obvious way.

In essence, graph H_j retains all the relevant information about distances between vertices of W_j. This is formalized in the following lemma.

Lemma 16. *The following assertions hold for each $j \in \mathbb{N}$:*

(P1) For every pair of vertices $u, v \in V(H_j)$, we have $\text{dist}_{H_j}(u, v) \geqslant \text{dist}_{G'}(u, v)$.

(P2) For every vertex $u \in V(H_j)$, we have $\text{dist}_{H_j}(u, s) = \text{dist}_{G'}(u, s)$.

(P3) For every pair of vertices $u, v \in V(H_j)$ satisfying $u \in W_j$ and $\text{dist}_{G'}(u, v) \leqslant 3r$, we have $\text{dist}_{H_j}(u, v) = \text{dist}_{G'}(u, v)$.

Proof: For assertion (P1), it suffices to observe that every path in H_j with endpoints u and v can be lifted to a path in G' of the same length by substituting any edge incident to s, say sw, by the shortest path between s and w in G'. For assertion (P2), we already know that $\text{dist}_{H_j}(u, s) \geqslant \text{dist}_{G'}(u, s)$, and to see that $\text{dist}_{H_j}(u, s) \leqslant \text{dist}_{G'}(u, s)$ we may observe that on the shortest path in G' from s to u, vertices contracted onto s form a prefix; this prefix can be then replaced by a single edge of the same weight. For assertion (P3), the assumption that $u \in W_j$ implies that in G', the vertex u is at distance more than $3r$ from any vertex that is removed or contracted onto s in the construction of H_j. Hence, the shortest path from u to v in G' survives the construction of H_j intact. ∎

Fix

$$L = 8r(q+1) \leqslant 16\varepsilon^{-1}r.$$

For future reference, we also note the following observation.

Lemma 17. *Let Q be a shortest path in H from s to some vertex u. Then the length of $Q - s$ (i.e. Q with the first vertex removed) is smaller than L.*

Proof: Let v be the successor of s on the path Q. By the construction of H we have that $u, v \in \bigcup_{(j-1)q+a \leqslant \iota \leqslant jq+a} L_\iota$ which in particular means that

$$8r((j-1)q+a) \leqslant \text{dist}(s, v), \text{dist}(s, u) < 8r(jq+a+1).$$

Since v lies on the shortest path from s to u, it follows that the length of the suffix of Q from v to u (which is $Q - s$) is equal to the $\text{dist}(v, u)$, which in turn is smaller than $8r(jq+a+1) - 8r((j-1)q+a) = 8r(q+1) = L$. ∎

Having defined the graph H_j, we define the facility set F_j and client set C_j as follows:

$$F_j = F' \cap \bigcup_{(j-1)q+a \leqslant \iota \leqslant jq+a} L_\iota$$

and

$$C_j = \bigcup_{f \in D^\circ \cap W_j} \text{cluster}(f).$$

Note that $F_j \subseteq V(H_j)$ and $C_j \subseteq V(H_j)$. Finally, we put

$$J_j = (H_j, C_j, F_j, \text{open});$$

that is, the opening costs are inherited from the original instance J. We now prove that by paying a small cost, we may solve instances J_j separately.

Lemma 18. *We have*

$$\text{OPT}(J') \geqslant \sum_{j \in \mathbb{N}} \text{OPT}(J_j).$$

Moreover, for any sequence of solutions $(R_j)_{j \in \mathbb{N}}$ to instances $(J_j)_{j \in \mathbb{N}}$, respectively, we have

$$\text{cost}\left(S \cup \bigcup_{j \in \mathbb{N}} R_j; J'\right) \leqslant \varepsilon \cdot M + \sum_{j \in \mathbb{N}} \text{cost}(R_j; J_j).$$

Proof: For each $j \in \mathbb{N}$, let D_j be the set consisting of all facilities $f \in D$ with the following property: there exists a client $c \in C_j$ for which f is the closest facility from D. By Lemmas 12 and 13, we have $\text{dist}_{G'}(c, D_j) \leqslant 3r$ for all $c \in C_j$, while from the definition of D_j it further follows that $\text{dist}_{G'}(f, C_j) \leqslant 3r$ for all $f \in D_j$. Also, every client $c \in C_j$ is at distance at most r from the center of its cluster, which is a facility of D° that resides in W_j. Hence, every facility $f \in D_j$ is at distance at most $4r$ from W_j. By Lemma 15 and triangle inequality we now infer that sets $(D_j)_{j \in \mathbb{N}}$ are pairwise disjoint. Moreover, we have $D_j \subseteq F_j$ and thus D_j can be treated as a solution to the instance J_j Therefore, by Lemma 16, assertions (P1) and (P3), we have

$$
\begin{aligned}
\text{OPT}(J') &= \text{cost}(D; J') \\
&= \text{open}(D) + \sum_{c \in C} \text{dist}_{G'}(c, D) \\
&= \sum_{j \in \mathbb{N}} \left(\text{open}(D_j) + \sum_{c \in C_j} \text{dist}_{G'}(c, D_j) \right) \\
&= \sum_{j \in \mathbb{N}} \left(\text{open}(D_j) + \sum_{c \in C_j} \text{dist}_{H_j}(c, D_j) \right) \\
&= \sum_{j \in \mathbb{N}} \text{cost}(D_j; J_j) \geqslant \sum_{j \in \mathbb{N}} \text{OPT}(J_j),
\end{aligned}
$$

completing the proof of the first assertion.

For the second assertion, since C_S and $(C_j)_{j \in \mathbb{N}}$ form a partition of C, we have

$$\text{cost}\left(S \cup \bigcup_{j \in \mathbb{N}} R_j; J'\right)$$

$$\leqslant \quad \text{open}(S) + \sum_{c \in C_S} \text{dist}(c, S)$$

$$+ \sum_{j \in \mathbb{N}} \left(\text{open}(R_j) + \sum_{c \in C_j} \text{dist}_{G'}(c, R_j)\right)$$

$$\leqslant \quad \text{open}(S) + \sum_{c \in C_S} \text{dist}(c, S)$$

$$+ \sum_{j \in \mathbb{N}} \left(\text{open}(R_j) + \sum_{c \in C_j} \text{dist}_{H_j}(c, R_j)\right)$$

$$\leqslant \quad \varepsilon \cdot M + \sum_{j \in \mathbb{N}} \text{cost}(R_j; J_j).$$

where in the second inequality we use Lemma 16, assertion (P1), while in the last inequality we use (17). ∎

Hence, from now on we focus on finding a near-optimum solutions to instances J_j, for each $j \in \mathbb{N}$ for which $C_j \neq \emptyset$, as such solutions can be combined into a near-optimum solution to J' using Lemma 18, which is then a near-optimum solution to J by Corollary 14. This will be done by dynamic programming. Fix $j \in J$ for which C_j is non-empty. For brevity, in the following we write H for H_j. Before we proceed, let us observe that J_j enjoys the same proximity property as J, expressed in Lemma 12.

Lemma 19. *Suppose D_j is an optimum solution in the instance J_j. Then for each $c \in C_j$ there exists $g \in D_j$ such that $\text{dist}_H(c, g) \leqslant 3r$.*

Proof: Apply the same reasoning as in the proof of Lemma 12, noting that all relevant vertices and paths are completely contained H due to being at distance at most $3r$ from W_j. ∎

Getting a suitable decomposition: Our dynamic programming will work over a suitable decomposition of the graph H. To define this decomposition, we will need some structural understanding of H and its embedding.

Recall that we assume that H is embedded in a sphere Σ. We shall assume that H is triangulated, as we can always triangulate it using edges of weight $+\infty$. Let L be the set of faces[1] of H. For future reference, we let $\xi \colon V(H) \to L$ be a function that assigns to every vertex u of H an arbitrary face $\xi(u)$ incident to u.

Let S be the spanning tree of shortest paths from s. That is, if for each $v \in V(H)$ by P_v we denote the shortest path from v to s in H, then S is the union of paths $\{P_v \colon v \in V(H)\}$.

---

[1]We use L here instead of usual F in order to avoid using the same letter as for facility sets.

Let $S^\star$ be the spanning subgraph of the dual $H^\star$ of H consisting of edges of $H^\star$ that are dual to the edges *not* belonging to S. It is well-known that $S^\star$ is then a spanning tree of $H^\star$.

Let

$$A = \{(f, g), (g, f) \colon fg \in E(S^\star)\};$$

that is, for each edge fg of $S^\star$ we add to A two (oriented) arcs: (f, g) and (g, f). For an arc $a \in A$, let $L(a) \subseteq L$ denote the set of those faces of H that are contained in this connected component of $S^\star$ with (unoriented) a removed that contains the head of a. For nonempty $B \subseteq A$, we denote

$$L(B) = \bigcap_{a \in B} L(a),$$

and we put $L(\emptyset) = L$ by convention. We may now state and prove the decomposition lemma that we shall need; in the following, all logarithms are base 2.

Lemma 20. *In polynomial time one can compute a rooted tree T together with a labelling β of nodes of T with subsets of A such that the following holds:*

(T1) T has depth at most $\log n$;

(T2) for each node t of T, we have $|\beta(t)| \leqslant 3$;

(T3) if t_0 is the root of T, then $L(\beta(t_0)) = L$;

(T4) for each leaf t of T, we have $|L(\beta(t))| = 1$;

(T5) each non-leaf node t of T has at most 7 children, and if $\text{chld}(t)$ denotes the set of children of t, then

$$L(\beta(t)) = \biguplus_{t' \in \text{chld}(t)} L(\beta(t))$$

and

$$\beta(t) \subseteq \bigcup_{t' \in \text{chld}(t)} \beta(t').$$

Proof: A subset X of nodes of $S^\star$ is *connected* if it induces a connected subtree of X. For a subset of nodes X, by ∂X we denote the set of edges of $S^\star$ with one endpoint in X and second outside of X. Let a *block* be any nonempty, connected subset of nodes X such that $|\partial X| \leqslant 3$. Note that since H is triangulated, $S^\star$ is a tree with maximum degree at most 3, so every node of T constitutes a single-node block. We observe the following.

Claim 2. *Every block X with $|X| \geqslant 2$ admits a partition into at most 7 blocks, each of size at most $|X|/2$.*

PROOF. Let $Z \subseteq X$ be the set of all the nodes of X that have a neighbor (in $S^\star$) outside of X. Then $|Z| \leqslant 3$ and, consequently, there exists a node $x \in X$ such that every connected component of $S^\star[X] - x$ contains at most one node of Z. Further, it is well known that in $S^\star[X]$ there exists a balanced node: a node y such that every connected component of $S^\star[X] - y$ has at most $|X|/2$ nodes. Then $S^\star[X] - \{x, y\}$ has at most 5 connected components, and it is straightforward to see that each of them is a block and

---

contains at most $|X|/2$ nodes. Hence, as $|X| \geqslant 2$, for the promised partition of X into blocks we can take the node sets of the connected components of $S^\star[X] - \{x, y\}$, plus blocks $\{x\}$ and $\{y\}$ (or just $\{x\}$, in case $x = y$). ⌟

We now construct the tree T together with labeling $\beta(\cdot)$ by recursively applying Claim 2 as follows. We start with the block L and, as long as the currently decomposed block X has size larger than 1, we apply Claim 2 to X and recursively decompose all the blocks comprising the obtained partition. Then T is the tree of this recursion and the nodes of T can be naturally labelled with blocks decomposed in corresponding calls; thus, the root of T is labelled by L, while the leaves of T are labelled by single-node blocks. Finally, for every node t of T, say associated with a block X_t, we set $\beta(t)$ to consist of edges of ∂X_t oriented towards endpoints belonging to X_t. It is straightforward to verify that the obtained pair (T, β) satisfies all of the required properties. Also, the above reasoning can be trivially translated into a polynomial-time algorithm computing (T, β). ∎

Thus, Lemma 20 essentially provides a hierarchical decomposition of the face set of H using separators consisting of six-tuples of shortest paths originating in s: two per each arc in $\beta(t)$. The idea is to put portals on those separators and run a bottom-up dynamic programming on the tree T that assembles a near-optimum solution while snapping paths to the portals along the way. First, however, we need to understand how to put portals on paths in H.

Portalization: Let X be a set of vertices of H and let $f : X \to \mathbb{R} \cup \{+\infty\}$ be a function. For positive reals d, σ and reals $\alpha \leqslant \beta$, we shall say that f is

- *d-discrete* if all its values are integer multiples of d;
- *$[\alpha, \beta]$-bounded* if every its value is either $+\infty$ or belongs to the interval $[\alpha, \beta]$; and
- *Lipschitz with slack σ* if

$$|f(u) - f(v)| \leqslant \text{dist}(u, v) + \sigma$$

for all $u, v \in X$ with $f(u), f(v) < +\infty$.

A function that is d-discrete, $[\alpha, \beta]$-bounded, and Lipschitz with slack σ will be called *$(d, \alpha, \beta, \sigma)$-normal*.

For portalization of shortest paths we shall use the following lemma.

Lemma 21. *Let P be a shortest path in H with one endpoint in s and let $d \in \mathbb{R}_{\geqslant 0}$. Then one can find a set Π of at most $(L/d)+2$ vertices on P with the following property: for every vertex u on P, there exists $\pi \in \Pi$ such that $\text{dist}(u, \pi) \leqslant d$. Moreover, for any reals $\alpha \leqslant \beta$, the number of functions on Π that are (d, α, β, d)-normal is at most $((\beta - \alpha)/d)^2 \cdot 2^{\mathcal{O}(L/d)}$, and such functions can be enumerated in time $((\beta - \alpha)/d)^2 \cdot 2^{\mathcal{O}(L/d)}$.*

Proof: Let $m = \beta - \alpha$. Let $P' = P - s$, i.e., P' is P with the first vertex removed. Then, by Lemma 17, the length of P' is smaller than L.

Let u and v be the endpoints of P'; then P' is the shortest path connecting u and v. Partition the vertices of P' into intervals $I_0, I_1, I_2, \ldots, I_p$, where $p = \lfloor L/d \rfloor$ such that I_i comprises vertices w of P' satisfying $id \leqslant \text{dist}(u, w) < (i+1)d$; since the length of P' is smaller than L, each of the vertices of P' is placed in one of these intervals. Observe that vertices within every interval I_i are pairwise at distance smaller than d. Therefore, we may construct a suitable set Π' for the path P' by taking one vertex π_i from each interval I_i that is non-empty; thus, Π' has size at most $p \leqslant (L/d) + 1$. Finally, we set $\Pi = \Pi' \cup \{s\}$.

We now bound the number (d, α, β, d)-normal functions f on Π. Note that there are at most $m/d + 2$ possibilities for the value $f(s)$, as this value is either an integer multiple of d between α and β, or $+\infty$. Therefore, it suffices to bound the number of (d, α, β, d)-normal functions on Π' by $(m/d) \cdot 2^{\mathcal{O}(L/d)}$. Recall that $|\Pi'| \leqslant (L/d) + 1$, hence there are at most $2^{(L/d)+1}$ choices on which portals will be assigned value $+\infty$. Supposing that this choice has been made, we bound the number of choices of (finite) values on remaining portals. Let $1 \leqslant i_1 < i_2 < \ldots < i_q \leqslant p$ be the indices such that portals chosen to be assigned finite values are in intervals $I_{i_1}, \ldots, I_{i_q}$. As above, there are at most $m/d + 1$ possibilities for the value $f(\pi_{i_1})$. However, for $j > 1$, the value $f(\pi_{i_j})$ must satisfy inequality

$$\begin{aligned}
|f(\pi_{i_j}) - f(\pi_{i_{j-1}})| &\leqslant \text{dist}(\pi_{i_j}, \pi_{i_{j-1}}) + d \\
&< (i_j - i_{j-1} + 1)d + d \\
&= (i_j - i_{j-1})d + 2d.
\end{aligned}$$

As $f(\pi_{i_j})$ has to be an integer multiple of d, once $f(\pi_{i_{j-1}})$ has been chosen, there are at most $2(i_j - i_{j-1}) + 4$ choices for the value of $f(\pi_{i_j})$. Hence, having chosen $f(\pi_{i_1})$, the number of choices for the remaining values $f(\pi_{i_2}), \ldots, f(\pi_{i_q})$ is bounded by

$$\begin{aligned}
\prod_{j=2}^{q} (2(i_j - i_{j-1}) + 4) &\leqslant 6^q \cdot \prod_{j=2}^{q} (i_j - i_{j-1}) \\
&\leqslant 6^q \cdot \prod_{j=2}^{q} 2^{i_j - i_{j-1}} \\
&= 6^q \cdot 2^{i_q - i_1} \\
&\leqslant 6^q \cdot 2^p \leqslant 12^p.
\end{aligned}$$

Since $p \leqslant (L/d) + 1$, we conclude that the total number of (d, α, β, d)-normal functions on Π' is bounded by $(m/d) \cdot 2^{\mathcal{O}(L/d)}$, as required.

The above reasoning can be trivially used to construct the promised enumeration algorithm. ∎

Defining subproblems: As expected, in dynamic programming we will need to solve more general subproblems, where portals on boundaries of these subproblems are taken into account. Formally, in an instance of the generalized problem we are working with:

- The original set of available facilities F_j, which we denote $F^\diamond$ for consistency; this set is always the same in all instance of the generalized problem, and is equipped with the original opening cost function $\mathsf{open}(\cdot)$.
- A subset of relevant clients $C^\diamond \subseteq C_j$; this set varies in instances of the generalized problem.
- A set of portals Π, which are vertices of H.
- A *prediction function* $\mathsf{pred}\colon \Pi \to \mathbb{R} \cup \{+\infty\}$.
- A *request function* $\mathsf{req}\colon \Pi \to \mathbb{R} \cup \{+\infty\}$.

Whenever considering an instance of the generalized problem, all distances are measured in H. Note that we allow negative requests and predictions.

Consider an instance $K = (C^\diamond, \Pi, \mathsf{req}, \mathsf{pred})$ of the generalized problem. For a solution $R \subseteq F^\diamond$, the connection cost of a client $c \in C^\diamond$ is defined as

$$\mathsf{conn}_K(c, R) = \min(\min_{f \in R} \mathrm{dist}(c, f), \min_{\pi \in \Pi}(\mathrm{dist}(c, \pi) + \mathsf{pred}(\pi))).$$

That is, every client can be connected either to a facility of f at the cost of the distance to this facility, or to a portal at the cost of the distance to this portal plus its prediction. Note that portals are always all open, so the factor $\min_{\pi \in \Pi}(\mathrm{dist}(c, \pi) + \mathsf{pred}(\pi))$ is independent of the solution R. We will say that c is *served* by the facility f or portal π for which the minimum above is attained.

A solution $R \subseteq F^\diamond$ is *feasible* if for every portal $\rho \in \Pi$ with $\mathsf{req}(\rho) \neq +\infty$, its request is satisfied in the following sense:

$$\min_{f \in R} \mathrm{dist}(\rho, f) \leqslant \mathsf{req}(x).$$

Note that the request of a portal has to be satisfied by a facility included in the solution; it cannot be satisfied by another portal. Again ρ is *served* by the facility f for which the minimum above is attained.

To analyze the approximation error, we will need to gradually relax the feasibility constraint. For this, for a nonnegative real λ we shall say that a solution $R \subseteq F^\diamond$ is λ-*near feasible* if for every portal $\rho \in \Pi$ with $\mathsf{req}(\rho) \neq +\infty$ there exists a facility $f \in R$ with $\mathrm{dist}(\rho, f) \leqslant \mathsf{req}(\rho) + \lambda$. That is, we relax all requests by an additive factor of λ.

Finally, for $\gamma \in \mathbb{R}_{\geqslant 0}$, a solution $R \subseteq F^\diamond$ is γ-*close* in K if

$$\mathsf{conn}_K(c, R) \leqslant \gamma \qquad \text{for every } c \in C^\diamond; \text{and}$$
$$\mathrm{dist}(\pi, R) \leqslant \gamma \qquad \text{for every } \pi \in \Pi \text{ with } \mathsf{req}(\pi) \neq +\infty.$$

The cost of a solution R is defined as

$$\mathsf{cost}(R; K) = \mathsf{open}(R) + \sum_{c \in C^\diamond} \mathsf{conn}_K(c, R).$$

Note that the connection costs of portals *do not* contribute to the cost of the solution. They are only used to define (near) feasibility of a solution. Thus, every portal essentially puts a hard constraint that there needs to be a facility opened within some distance from it. By $\mathsf{OPT}(K)$ we denote the minimum cost of a feasible solution to K.

The intuitive meaning of predictions and requests in the dynamic programming are as follows. In the following, think of dynamic programming over the decomposition provided by Lemma 20 as a recursive algorithm that breaks the given instance into simpler ones (whose number is at most 7), solves them using subcalls, and assembles the obtained solutions into a solution to the input instance. Whenever we break the instance using some separator, which constists of a constant number of shortest paths, we put portals along them using Lemma 21 in all the obtained subinstances. For every portal π we guess in which subinstance lies the closest facility f that is open in the (unknown) optimum solution, and we approximately guess the distance d from π to this facility (up to additive accuracy δ, to be defined later). This allows us to define the requests and predictions in subinstances: in the subinstance that is guessed to contain f we put a request d on π to make sure that some facility at this distance is indeed open there, while in other subinstances we put a prediction d on π, so that solutions in these subinstances may use a virtual, "promised" facility at distance d from π.

Since recursion has depth $\mathcal{O}(\log n)$ by Lemma 20, condition (T1), the rounding error will accumulate through $\mathcal{O}(\log n)$ levels. Therefore, we needed to put $\delta = \mathcal{O}(\varepsilon / \log n)$ and make rounding errors of magnitude $\mathcal{O}(\delta) \cdot \mathsf{OPT}$ at each level, so that the total error is kept at $\mathcal{O}(\varepsilon) \cdot \mathsf{OPT}$. Precisely, we fix

$$\delta = \frac{\varepsilon}{\log n}.$$

Dynamic programming states: Once we have defined the generalized problem with portals, we may formally define the instances solved in the dynamic programming. For every vertex v of H, we may apply Lemma 21 to P_v and $d = \delta$, thus obtaining a suitable set of vertices $\Pi_v \subseteq V(P_v)$ of size at most $\delta^{-1}L + 2 = \mathcal{O}(\varepsilon^{-2}r \log n)$.

For each node t of T, we define

$$C_t^\diamond = \xi^{-1}(L(\beta(t))) \cap C^\diamond \qquad \text{and} \qquad \Pi_t = \bigcup_{uv \in B_t} \Pi_u \cup \Pi_v,$$

where B_t is the set of edges of H dual to the arcs of $\beta(t)$. Note that by condition (T5) of Lemma 20, we have

$$\Pi_t \subseteq \bigcup_{t' \in \mathsf{chld}(t)} \Pi_{t'} \qquad \text{for each non-leaf node } t \text{ of } T.$$

Observe also that if t_0 is the root of T, then $C_{t_0}^\diamond = C_j$ and $\Pi_{t_0} = \emptyset$. Finally, the following lemma expresses the crucial separation property provided by the decomposition (T, β).

Lemma 22. *Let s and t be nodes of T that are not in the ancestor-descendant relation, and let $u \in \xi^{-1}(L(\beta(s)))$ and $v \in \xi^{-1}(L(\beta(t)))$. Then there exists a portal $\rho \in \Pi_t$ such that*

$$\mathrm{dist}(u, v) \geqslant \mathrm{dist}(u, \rho) + \mathrm{dist}(\rho, v) - 2\delta.$$

Furthermore, the same holds when s is an ancestor of t and $u \in \Pi_s$.

Proof: Let B be the set of edges of H that are dual to the arcs of $\beta(t)$, and let Z be the set of endpoints of these edges. Consider removing all paths P_z for $z \in Z$ and all edges of B from the plane. Then the plane breaks into several connected components, out of which one consists of exactly the faces of $L(\beta(t))$. It follows that every path connecting a vertex from $\xi^{-1}(L(\beta(t)))$ with a vertex that does not belong to $\xi^{-1}(L(\beta(t)))$ has to intersect one of the paths P_z for some $z \in Z$. Observe that $v \in \xi^{-1}(L(\beta(t)))$. Moreover, if s and t are not in the ancestor-descendant relation in T, then $L(\beta(s))$ and $L(\beta(t))$ are disjoint, implying $u \notin \xi^{-1}(L(\beta(t)))$. Also, if $u \in \Pi_s$ and s is an ancestor of t, then either u lies on one of the paths P_z for $z \in Z$, or $u \notin \xi^{-1}(L(\beta(t)))$.

In both cases we conclude that the shortest path connecting u and v, call it Q, has to intersect the path P_z for some $z \in Z$. Let w be any vertex in the intersection of these two paths. Then, by Lemma 21, there exists $\rho \in \Pi_z \subseteq \Pi_t$ such that $\mathrm{dist}(w, \rho) \leqslant \delta$. We conclude that

$$
\begin{aligned}
\mathrm{dist}(u,v) &= \mathrm{dist}(u,w) + \mathrm{dist}(w,v) \\
&\geqslant \mathrm{dist}(u,w) + \mathrm{dist}(w,\rho) \\
&\quad + \mathrm{dist}(\rho,w) + \mathrm{dist}(w,v) - 2\delta \\
&\geqslant \mathrm{dist}(u,\rho) + \mathrm{dist}(\rho,v) - 2\delta,
\end{aligned}
$$

as required. ∎

For every node t of T, we define $\widetilde{\mathcal{N}}_t$ to be the set of all functions from Π_t to $\mathbb{R} \cup \{+\infty\}$. Further, let $\mathcal{N}_t \subseteq \widetilde{\mathcal{N}}_t$ be the subset of all those functions from $\widetilde{\mathcal{N}}_t$ that are $(\delta, -5\varepsilon, 3r + 5\varepsilon, \delta)$-normal; in the sequel, when saying just *normal* we mean being $(\delta, -5\varepsilon, 3r + 5\varepsilon, \delta)$-normal. While $\widetilde{\mathcal{N}}_t$ is infinite, $\mathcal{N}_t$ is finite and actually of polynomial size.

Lemma 23. *For each node t of T we have that $|\mathcal{N}_t| \leqslant n^{\mathcal{O}(\varepsilon^{-2}r)}$ and $\mathcal{N}_t$ can be enumerated in time $n^{\mathcal{O}(\varepsilon^{-2}r)}$.*

Proof: By Lemma 17, for each vertex u of H the number of normal functions on Π_u is at most $(\delta^{-1}r)^2 \cdot 2^{\mathcal{O}(\delta^{-1}L)} = n^{\mathcal{O}(\varepsilon^{-2}r)}$. Observe that Π_t is the union of at most 6 sets of the form Π_u, for vertices u that are endpoints of edges dual to the arcs $\beta(t)$. Hence every normal function on Π_t can be described by a 6-tuple of such functions on sets of the form Π_u for u as above. Thus, we have $|\mathcal{N}_t| \leqslant n^{\mathcal{O}(\varepsilon^{-2}r)}$ as well. Moreover, since normal functions on Π_u can be enumerated in time $n^{\mathcal{O}(\varepsilon^{-2}r)}$ for each vertex u, to enumerate $\mathcal{N}_t$ it suffices to enumerate all 6-tuples of functions as above, and filter out those 6-tuples whose union is either ill-defined or is not Lipschitz with slack δ. This takes time $n^{\mathcal{O}(\varepsilon^{-2}r)}$. ∎

Now, for every $t \in V(T)$ and pair $\eta = (\mathsf{pred}, \mathsf{req}) \in \widetilde{\mathcal{N}}_t \times \widetilde{\mathcal{N}}_t$, we define the instance $K_t(\eta)$ of the generalized problem as follows:

$$
K_t(\eta) = (C_t^\diamond, \Pi_t, \mathsf{pred}, \mathsf{req}).
$$

Before the explaining how these instances are going to be solved using dynamic programming, let us verify that the subproblem at the root of T corresponds to the instance J_j that we are trying to (approximately) solve.

Lemma 24. *Suppose t_0 is the root of T and, noting that $\Pi_{t_0} = \emptyset$, we let $K = K_{t_0}((\emptyset, \emptyset))$. Then, for any $\lambda \geqslant 0$, every λ-near feasible solution R to K satisfies*

$$
\mathsf{cost}(R; J_j) = \mathsf{cost}(R; K).
$$

In particular, we have

$$
\mathsf{OPT}(J_j) = \mathsf{OPT}(K).
$$

Proof: The first assertion follows immediately by observing that the formulas for $\mathsf{cost}(R; J_j)$ and $\mathsf{cost}(R; K)$ are the same, because there are no portals in K. The second assertion follows immediately from the first by observing that every solution R to K is λ-near feasible for any $\lambda \geqslant 0$, because in K there are no portals. ∎

Computing transitions: We first show that the subproblems in the leaves of T can be solved in polynomial time. For this, we use the following lemma.

Lemma 25. *There is an algorithm that given an instance $K = (C^\diamond, \Pi, \mathsf{pred}, \mathsf{req})$ of the generalized problem and $\lambda \geqslant 0$, finds the least expensive λ-near feasible solution to K in time $3^{|\Pi|+k} \cdot n^{\mathcal{O}(1)}$, where k is the total number of distinct vertices on which the clients of $C^\diamond$ are placed.*

Proof: Let W be the set of distinct vertices on which $C^\diamond$ are placed, and for $u \in W$ let $\gamma(u)$ be the number of clients placed at vertex u. We perform standard dynamic programming over subsets of Π and of W, where we keep track of the cost of connecting any subset of portals and any subset of vertices of W, while introducing candidate facilities one by one. Precisely, let $f_1, \ldots, f_p$ be the facilities of $F^\diamond$, enumerated in any order. Then for every $i \in \{0, 1, \ldots, p\}$, $A \subseteq \Pi$, and $B \subseteq W$, define value $\mathsf{dp}[i, A, B]$ to be the smallest cost of a λ-near feasible solution contained in $\{f_1, f_2, \ldots, f_i\}$, where in the near-feasibility check we consider only requests of portals from A, and in the connection cost computation we consider only clients placed at vertices from B. Then it is easy to see that the function $\mathsf{dp}[\cdot, \cdot, \cdot]$ satisfies the following recursive formula.

$$
\mathsf{dp}[0, A, B] = \begin{cases} 0 & \text{if } A = B = \emptyset, \\ +\infty & \text{otherwise;} \end{cases}
$$

$$
\begin{aligned}
\mathsf{dp}[i, A, B] = \min(\ &\mathsf{dp}[i-1, A, B], \\
&\mathsf{open}(f_i) \\
&+ \min_{\substack{A' \subseteq A, \, B' \subseteq B : \, \forall \pi \in A \setminus A' \\ \mathrm{dist}(\pi, f_i) \leqslant \mathsf{req}(\pi) + \lambda}} \mathsf{dp}[i-1, A', B'] \\
&+ \sum_{u \in B \setminus B'} \gamma(u) \cdot \mathrm{dist}(u, f_i) \quad).
\end{aligned}
$$

Using the above formula, we can in time $3^{|\Pi|+k} \cdot n^{\mathcal{O}(1)}$ compute all the $2^{|\Pi|+k} \cdot (p+1)$ values of the function $\mathsf{dp}[\cdot, \cdot, \cdot]$, and return $\mathsf{dp}[p, \Pi, W]$ as the sought minimum cost. A λ-near feasible solution attaining this cost can be retrieved from dynamic programming tables by standard means within the same running time. ∎

Corollary 26. *Suppose t is a leaf of T and $\lambda \geqslant 0$ is a given real. Then, in total time $n^{\mathcal{O}(\varepsilon^{-2}r)}$ one can compute, for each $\eta \in \mathcal{N}_t \times \mathcal{N}_t$, the least expensive λ-near feasible solution $R_{t,\eta} \subseteq F^{\diamond}$ to $K_t(\eta)$.*

Proof: To compute each solution $R_{t,\eta}$, we apply the algorithm of Lemma 25 to instance $K_t(\eta)$ for $\eta \in \mathcal{N}_t \times \mathcal{N}_t$ and λ. Since t is a leaf of T, all clients in $K_t(\eta)$ lie on the unique face of $L(\beta(t))$ (Lemma 20, condition (T4)), hence they are all place on distinct three vertices. Therefore, the running time used by each application of the algorithm of Lemma 25 is $3^{|\Pi_t|+3} \cdot n^{\mathcal{O}(1)} = n^{\mathcal{O}(\varepsilon^{-2}r)}$. Since the number of pairs $\eta \in \mathcal{N}_t \times \mathcal{N}_t$ is $|\mathcal{N}_t|^2 \leqslant n^{\mathcal{O}(\varepsilon^{-2}r)}$, the total running time follows. ∎

We now proceed to the main point: how to compute values for a node of T based on values for its children. We first introduce even more helpful notation. For a non-leaf node t of T, let $\Omega_t = \bigcup_{t' \in \mathsf{chld}(t)} \Pi_t$; then $\Pi_t \subseteq \Omega_t$.

For a non-leaf node t of T, define

$$\widetilde{\mathcal{M}}_t = \prod_{t' \in \mathsf{chld}(t)} \widetilde{\mathcal{N}}_t.$$

For each $t' \in \mathsf{chld}(t)$ we have a natural restriction operator $\mathsf{restrict}_{t,t'} : \widetilde{\mathcal{M}}_t \to \widetilde{\mathcal{N}}_{t'}$ that maps every tuple from $\widetilde{\mathcal{M}}_t$ to its t'-component. Next, define

$$\widetilde{\mathcal{U}}_t = \widetilde{\mathcal{N}}_t \times \widetilde{\mathcal{N}}_t \qquad \text{and} \qquad \widetilde{\mathcal{W}}_t = \widetilde{\mathcal{M}}_t \times \widetilde{\mathcal{M}}_t.$$

Operator $\mathsf{restrict}_{t,t'}(\cdot)$ can be then regarded as an operator from $\widetilde{\mathcal{W}}_t$ to $\widetilde{\mathcal{U}}_{t'}$ by considering acting coordinate-wise.

Having defined sets $\widetilde{\mathcal{M}}_t$, $\widetilde{\mathcal{U}}_t$, and $\widetilde{\mathcal{W}}_t$, we define sets $\mathcal{M}_t$, $\mathcal{U}_t$, and $\mathcal{W}_t$ by replacing $\widetilde{\mathcal{N}}_t$ with $\mathcal{N}_t$ in the definitions. Since every node of T has at most 7 children (Lemma 20, condition (T5)), by Lemma 23 we have that $|\mathcal{M}_t| \leqslant n^{\mathcal{O}(\varepsilon^{-2}r)}$ and all sets $\mathcal{M}_t$ can be computed in time $n^{\mathcal{O}(\varepsilon^{-2}r)}$. Then we also have that

$$|\mathcal{U}_t|, |\mathcal{W}_t| \leqslant n^{\mathcal{O}(\varepsilon^{-2}r)} \qquad \text{for each node } t \text{ of } T,$$

and all the sets $\mathcal{U}_t, \mathcal{W}_t$ can be computed in time $n^{\mathcal{O}(\varepsilon^{-2}r)}$.

We now describe tuples from $\widetilde{\mathcal{W}}_t$ that may be used in the dynamic programming to combine solutions from smaller subproblems into a solution to a larger subproblem. The intuition here is that when breaking a subproblem into smaller ones, we have to ensure that requests and predictions appropriately match so that solutions to smaller subproblems can be combined to a solution to the original subproblem.

Definition 27. Consider a non-leaf node t of T. We shall say that a pair $\eta = (\mathsf{req}, \mathsf{pred}) \in \widetilde{\mathcal{U}}_t$ and a pair $\varphi =$

$((\mathsf{req}_{t'})_{t' \in \mathsf{chld}(t)}, (\mathsf{pred}_{t'})_{t' \in \mathsf{chld}(t)}) \in \widetilde{\mathcal{W}}_t$ are *compatible* (denoted $\eta \sim \varphi$) if the following two conditions hold:

(C1) For every $\pi \in \Pi_t$ with $\mathsf{req}(\pi) \neq +\infty$ there exists $t' \in \mathsf{chld}(t)$ and $\rho \in \Pi_{t'}$ such that $\mathsf{req}_{t'}(\rho) + \mathrm{dist}(\pi, \rho) \leqslant \mathsf{req}(\pi)$.

(C2) For every $t' \in \mathsf{chld}(t)$ and $\rho \in \Pi_{t'}$ with $\mathsf{pred}_{t'}(\rho) \neq +\infty$, there either exists $\pi \in \Pi_t$ with $\mathsf{pred}(\pi) + \mathrm{dist}(\pi, \rho) \leqslant \mathsf{pred}_{t'}(\rho)$, or there exists $t'' \in \mathsf{chld}(t)$ and $\rho' \in \Pi_{t''}$ with $\mathsf{req}_{t''}(\rho') + \mathrm{dist}(\rho', \rho) \leqslant \mathsf{pred}_{t'}(\rho)$.

Observe that given $\eta \in \widetilde{\mathcal{U}}_t$ and $\varphi \in \widetilde{\mathcal{W}}_t$, it can be verified in polynomial time whether $\eta \sim \varphi$.

Finally, we formulate and prove two lemmas that will imply the correctness of our dynamic programming. The first one concerns combining solutions to smaller subproblems into solutions to larger subproblems. The second one concerns projecting solutions to larger subproblems to solutions to smaller subproblems.

Lemma 28. *Suppose t is a non-leaf node of T and let $\eta \in \widetilde{\mathcal{U}}_t$ and $\varphi \in \widetilde{\mathcal{W}}_t$ be compatible. Suppose further that, for all $t' \in \mathsf{chld}(t)$, $R_{t',\eta_{t'}}$ is a feasible solution to the instance $K_{t'}(\eta_{t'})$, where $\eta_{t'} = \mathsf{restrict}_{t,t'}(\varphi)$. Then*

$$R = \bigcup_{t' \in \mathsf{chld}(t)} R_{t',\eta_{t'}}$$

is a feasible solution to the instance $K_t(\eta)$ and, moreover,

$$\mathsf{cost}(R; K_t(\eta)) \leqslant \sum_{t' \in \mathsf{chld}(t)} \mathsf{cost}(R_{t',\eta_{t'}}; K_{t'}(\eta_{t'})).$$

Proof: For brevity, we shall denote $R_{t'} = R_{t',\eta_{t'}}$ and $K_{t'} = K_{t'}(\eta_{t'})$. Also, let $\eta = (\mathsf{pred}, \mathsf{req})$ and $K_t = K_t(\eta)$.

We first verify that R is a feasible solution to K_t. Take any portal $\pi \in \Pi_t$ with $\mathsf{req}(\pi) \neq +\infty$. Since $\eta \sim \varphi$, by (C1) there exists $t' \in \mathsf{chld}(t)$ and $\rho \in \Pi_{t'}$ such that $\mathsf{req}_{t'}(\rho) + \mathrm{dist}(\pi, \rho) \leqslant \mathsf{req}(\pi)$. As $R_{t'}$ is a feasible solution to $K_{t'}$, there exists $f \in R_{t'}$ such that $\mathrm{dist}(\rho, f) \leqslant \mathsf{req}_{t'}(\rho)$. Then $f \in R$ as well and

$$\mathrm{dist}(\pi, f) \leqslant \mathrm{dist}(\pi, \rho) + \mathrm{dist}(\rho, f)$$
$$\leqslant \mathrm{dist}(\pi, \rho) + \mathsf{req}_{t'}(\rho) \leqslant \mathsf{req}(\pi),$$

which certifies that the request of π is satisfied by R. Hence, R is indeed a feasible solution to K_t.

We are left with proving the postulated upper bound on $\mathsf{cost}(R; K_t)$. Take any client $c \in C_t^{\diamond}$. As $(C_{t'}^{\diamond})_{t' \in \mathsf{chld}(t)}$ form a partition of $C_t^{\diamond}$, there exists a unique node $t' \in \mathsf{chld}(t)$ satisfying $c \in C_{t'}^{\diamond}$. Then there either exists a facility $f \in R_{t'}$ satisfying

$$\mathrm{dist}(c, f) = \mathsf{conn}_{K_{t'}}(c; R_{t'})$$

or there exists a portal $\rho \in \Pi_{t'}$ satisfying

$$\mathrm{dist}(c, \rho) + \mathsf{pred}_{t'}(\rho) = \mathsf{conn}_{K_{t'}}(c; R_{t'}).$$

In the former case, since $R_{t'} \subseteq R$ we can conclude that

$$\mathsf{conn}_K(c; R) \leqslant \mathsf{conn}_{K_{t'}}(c; R_{t'}). \tag{18}$$

576

In the latter case, by (C2) either exists $\pi \in \Pi_t$ with $\mathsf{pred}(\pi) + \mathrm{dist}(\pi, \rho) \leqslant \mathsf{pred}_{t'}(\rho)$, or there exists $t'' \in \mathsf{chld}(t)$ and $\rho' \in \Pi_{t''}$ with $\mathsf{req}_{t''}(\rho') + \mathrm{dist}(\rho', \rho) \leqslant \mathsf{pred}_{t'}(\rho)$. In the first subcase we conclude that

$$
\begin{aligned}
\mathsf{conn}_K(c; R) &\leqslant \mathrm{dist}(c, \pi) + \mathsf{pred}(\pi) \\
&\leqslant \mathrm{dist}(c, \rho) + \mathrm{dist}(\pi, \rho) + \mathsf{pred}(\pi) \\
&\leqslant \mathrm{dist}(c, \rho) + \mathsf{pred}_{t'}(\rho) = \mathsf{conn}_{K_{t'}}(c; R_{t'}),
\end{aligned}
$$

which again establish inequality (18) in this subcase. On the other hand, in the second subcase there exists a facility $f \in R_{t''}$ with $\mathrm{dist}(\rho', f) \leqslant \mathsf{req}_{t''}(\rho')$. As $f \in R$ as well, we infer that

$$
\begin{aligned}
\mathsf{conn}_K(c; R) &\leqslant \mathrm{dist}(c, f) \\
&\leqslant \mathrm{dist}(c, \rho) + \mathrm{dist}(\rho, \rho') + \mathrm{dist}(\rho', f) \\
&\leqslant \mathrm{dist}(c, \rho) + \mathrm{dist}(\rho, \rho') + \mathsf{req}_{t''}(\rho') \\
&\leqslant \mathrm{dist}(c, \rho) + \mathsf{pred}_{t'}(\rho) = \mathsf{conn}_{K_{t'}}(c; R_{t'}).
\end{aligned}
$$

Hence, again inequality (18) is satisfied.

We conclude that in every case, inequality (18) holds. Summing this inequality through all clients $c \in C_t^\diamond$ and adding $\mathsf{open}(R)$ to both sides yields yields that $\mathsf{cost}(R; K_t) \leqslant \sum_{t' \in \mathsf{chld}(t)} \mathsf{cost}(R_{t'}; K_{t'})$, as required. ∎

Lemma 29. *Suppose t is a non-leaf node of T. Suppose further that $\eta \in \widetilde{\mathcal{U}}_t$ is such that all predictions involved in η are nonnegative, and R is a λ-near feasible γ-close solution to $K_t(\eta)$, for some reals $\lambda, \gamma > 0$. Then there exist $\varphi \in \widetilde{\mathcal{W}}_t$ that is compatible with η and $(\lambda + 5\delta)$-near feasible $(\gamma + 5\delta)$-close solutions $R_{t', \eta_{t'}} \subseteq R$ to instances $K_{t'}(\eta_{t'})$ for $t' \in \mathsf{chld}(t)$, where $\eta_{t'} = \mathsf{restrict}_{t,t'}(\varphi)$, such that*

$$
\mathsf{cost}(R; K_t(\eta)) \geqslant \sum_{t' \in \mathsf{chld}(t)} \mathsf{cost}(R_{t', \eta_{t'}}; K_{t'}(\eta_{t'})) - 5\delta |C_t^\diamond|.
$$

Moreover, all request and prediction functions involved in φ are $(\delta, -\lambda - 5\delta, \gamma + 4\delta, \delta)$-normal, and all predictions involved in φ are nonnegative.

Proof: Denote $K_t = K_t(\eta)$ and $\eta = (\mathsf{pred}, \mathsf{req})$. For each $t' \in \mathsf{chld}(t)$, let

$$
R_{t'} = \xi^{-1}(L(\beta(t'))) \cap R.
$$

Then $(R_{t'})_{t' \in \mathsf{chld}(t)}$ form a partition of R.

For any $t' \in \mathsf{chld}(t)$ and $\rho \in \Pi_{t'}$, we shall say that ρ is *facility-important* if

- there exists a facility $f \in R_{t'}$ and a client $c \in C^\diamond$ served by f in R such that $\mathrm{dist}(c, \rho) + \mathrm{dist}(\rho, f) \leqslant \mathrm{dist}(c, f) + 4\delta$; or
- there exists a facility $f \in R_{t'}$ and portal $\pi \in \Pi_t$ with $\mathsf{req}(\pi) \neq +\infty$ served by f in R such that $\mathrm{dist}(\pi, \rho) + \mathrm{dist}(\rho, f) \leqslant \mathrm{dist}(\pi, f) + 2\delta$.

Further, ρ is *client-important* if

- there exists a client $c \in C_{t'}^\diamond$ and a facility $f \in R$ that serves c in R such that $\mathrm{dist}(c, \rho) + \mathrm{dist}(\rho, f) \leqslant \mathrm{dist}(c, f) + 2\delta$; or
- there exists a client $c \in C_{t'}^\diamond$ and a portal $\pi \in \Pi_t$ that serves c in R such that $\mathrm{dist}(c, \rho) + \mathrm{dist}(\rho, \pi) \leqslant \mathrm{dist}(c, \pi) + 2\delta$.

We observe the following.

Claim 3. *Let $\rho \in \Pi_{t'}$ for some $t' \in \mathsf{chld}(t)$. If ρ is facility-important, then*

$$
\min_{f \in R_{t'}} \mathrm{dist}(\rho, f) \leqslant \gamma + 4\delta.
$$

If ρ is client-important, then

$$
\min(\min_{f \in R} \mathrm{dist}(\rho, f), \min_{\pi \in \Pi_t} \mathrm{dist}(\rho, \pi) + \mathsf{pred}(\pi)) \leqslant \gamma + 2\delta
$$

PROOF. Recall that R is γ-close in K_t. When ρ is facility-important due to the first alternative in the definition, we have

$$
\mathrm{dist}(\rho, f) \leqslant \mathrm{dist}(c, f) + 4\delta \leqslant \gamma + 4\delta;
$$

here and in the following, we assume notation from the definition. Also, when ρ is facility-important due to the second alternative, we have

$$
\mathrm{dist}(\rho, f) \leqslant \mathrm{dist}(\pi, f) + 2\delta \leqslant \gamma + 2\delta.
$$

Now, if ρ is client-important due to the first alternative in the definition, then we have

$$
\mathrm{dist}(\rho, f) \leqslant \mathrm{dist}(c, f) + 2\delta \leqslant \gamma + 2\delta.
$$

Also, when ρ is facility-important due to the second alternative, we have

$$
\mathrm{dist}(\rho, \pi) + \mathsf{pred}(\pi) \leqslant \mathrm{dist}(c, \pi) + \mathsf{pred}(\pi) + 2\delta \leqslant \gamma + 2\delta.
$$

This concludes the proof. ⌟

For a real x, let $\mathsf{round}^\downarrow(x)$ be the largest integer multiple of δ that is not larger than x, and $\mathsf{round}^\uparrow(x)$ be the smallest integer multiple of δ that not smaller than x. That is,

$$
\mathsf{round}^\downarrow(x) = \delta \cdot \lfloor x/\delta \rfloor \qquad \text{and} \qquad \mathsf{round}^\uparrow(x) = \delta \cdot \lceil x/\delta \rceil.
$$

We now define $\varphi = (\mathsf{pred}_{t'}, \mathsf{req}_{t'})_{t' \in \mathsf{chld}(t)}$. Consider any $t' \in \mathsf{chld}(t)$ and $\rho \in \Pi_{t'}$. We put

- $\mathsf{req}_{t'}(\rho) = +\infty$ if ρ is not facility-important, and otherwise

$$
\mathsf{req}_{t'}(\rho) = -4\delta + \mathsf{round}^\downarrow \left(\min_{f \in R_{t'}} \mathrm{dist}(\rho, f) - \lambda \right);
$$

- $\mathsf{pred}_{t'}(\rho) = +\infty$ if ρ is not client-important, and otherwise

$$
\mathsf{pred}_{t'}(\rho) = 2\delta + \mathsf{round}^\uparrow \left(\min \left(\min_{f \in R} \mathrm{dist}(\rho, f), \min_{\pi \in \Pi_t} \mathrm{dist}(\rho, \pi) + \mathsf{pred}(\pi) \right) \right).
$$

Clearly, functions $\mathsf{req}_{t'}(\cdot)$ and $\mathsf{pred}_{t'}(\cdot)$ are δ-discrete and, as functions of ρ under rounding are Lipschitz, they are also Lipschitz with slack δ. We are left with verifying that these functions are also $[-\lambda - 5\delta, \gamma + 4\delta]$-bounded, η and φ are compatible, $R_{t'}$ is a $(\lambda + 5\delta)$-near feasible $(\gamma + 5\delta)$-close solution to $K_{t'}$ for each $t' \in \mathsf{chld}(t)$, where $K_{t'} = K_{t'}(\eta_{t'})$, and that the postulated lower bound on $\mathsf{cost}(R; K_t)$ holds. We prove these properties in the following claims.

Claim 4. *For each $t' \in \mathsf{chld}(t)$, the function $\mathsf{req}_{t'}(\cdot)$ is $[-\lambda - 5\delta, \gamma]$-bounded and the function $\mathsf{pred}_{t'}(\cdot)$ is $[0, \gamma + 4\delta]$-bounded.*

PROOF. First, take any $\rho \in \Pi_{t'}$ that is facility-important (as otherwise $\mathsf{req}_{t'}(\rho) = +\infty$ anyway). Then $\mathsf{req}_{t'}(\rho) \geqslant -\lambda - 5\delta$ by definition and $\mathsf{req}_{t'}(\rho) \leqslant \gamma$ by Claim 3. Next, take any $\rho \in \Pi_{t'}$ that is client-important (as otherwise $\mathsf{pred}_{t'}(\rho) = +\infty$ anyway). Then $\mathsf{pred}_{t'}(\rho) \geqslant 2\delta$ by definition and $\mathsf{pred}_{t'}(\rho) \leqslant \gamma + 4\delta$ by Claim 3. ⌟

Claim 5. *It holds that η and φ are compatible.*

PROOF. We first verify condition (C1). Take any $\pi \in \Pi_t$ with $\mathsf{req}(\pi) \neq +\infty$. Since R is a λ-near feasible solution to instance K_t, there exists $f \in R$ such that

$$\mathsf{dist}(\pi, f) \leqslant \mathsf{req}(\pi) + \lambda.$$

Then $f \in R_{t'}$ for some $t' \in \mathsf{chld}(t)$, and in particular $\xi(f) \in L(\beta(t'))$. By Lemma 22, there exists a portal $\rho \in \Pi_{t'}$ such that

$$\mathsf{dist}(\pi, f) \geqslant \mathsf{dist}(\pi, \rho) + \mathsf{dist}(\rho, f) - 2\delta. \qquad (19)$$

In particular ρ is facility-important, so combining the above with the definition of $\mathsf{req}_{t'}(\rho)$ we obtain

$$
\begin{aligned}
\mathsf{req}_{t'}(\rho) &\leqslant \mathsf{dist}(\rho, f) - \lambda - 4\delta \\
&\leqslant \mathsf{dist}(\pi, f) - \mathsf{dist}(\pi, \rho) + 2\delta - \lambda - 4\delta \\
&\leqslant \mathsf{req}(\pi) - \mathsf{dist}(\pi, \rho) - 2\delta;
\end{aligned}
$$

this directly implies (C1).

We now verify condition (C2). Take any $\rho \in \Pi_{t'}$ for any $t' \in \mathsf{chld}(t)$ with $\mathsf{pred}_{t'}(\rho) \neq +\infty$. Then ρ is client-important, so there exists a client $c \in C_{t'}^\diamond$ and either a facility $f \in R$ serving c and satisfying $\mathsf{dist}(c, \rho) + \mathsf{dist}(\rho, f) \leqslant \mathsf{dist}(c, f) + 2\delta$, or a portal $\pi \in \Pi_t$ serving c such that $\mathsf{dist}(c, \rho) + \mathsf{dist}(\rho, \pi) \leqslant \mathsf{dist}(c, \pi) + 2\delta$. We consider these two cases separately.

Suppose the first case holds. Since f serves c in R, for any $\pi' \in \Pi_t$ and $f' \in R$, we have

$$\mathsf{dist}(c, f) \leqslant \mathsf{dist}(c, \pi') + \mathsf{pred}(\pi')$$

and

$$\mathsf{dist}(c, f) \leqslant \mathsf{dist}(c, f').$$

Then we also have

$$
\begin{aligned}
\mathsf{dist}(\rho, f) &\leqslant \mathsf{dist}(c, f) - \mathsf{dist}(c, \rho) + 2\delta \\
&\leqslant \mathsf{dist}(c, \pi') + \mathsf{pred}(\pi') - \mathsf{dist}(c, \rho) + 2\delta \\
&\leqslant \mathsf{dist}(\rho, \pi') + \mathsf{pred}(\pi') + 2\delta,
\end{aligned}
$$

and similarly

$$
\begin{aligned}
\mathsf{dist}(\rho, f) &\leqslant \mathsf{dist}(c, f) - \mathsf{dist}(c, \rho) + 2\delta \\
&\leqslant \mathsf{dist}(c, f') - \mathsf{dist}(c, \rho) + 2\delta \\
&\leqslant \mathsf{dist}(\rho, f') + 2\delta.
\end{aligned}
$$

Therefore, by the definition of $\mathsf{pred}_{t'}(\rho)$, we have

$$\mathsf{pred}_{t'}(\rho) \geqslant \mathsf{dist}(\rho, f).$$

As $f \in R$, there exists $t'' \in \mathsf{chld}(t)$ such that $f \in R_{t''}$. Then, by Lemma 22, there is a portal $\rho' \in \Pi_{t''}$ such that

$$\mathsf{dist}(\rho, f) \geqslant \mathsf{dist}(\rho, \rho') + \mathsf{dist}(\rho', f) - 2\delta.$$

We note that

$$
\begin{aligned}
\mathsf{dist}(c, f) &\geqslant \mathsf{dist}(c, \rho) + \mathsf{dist}(\rho, f) - 2\delta \\
&\geqslant \mathsf{dist}(c, \rho) + \mathsf{dist}(\rho, \rho') + \mathsf{dist}(\rho', f) - 4\delta \\
&\geqslant \mathsf{dist}(c, \rho') + \mathsf{dist}(\rho', f) - 4\delta,
\end{aligned}
$$

implying that ρ' is facility-important. Therefore, by the definition of $\mathsf{req}_{t''}(\rho')$ we infer that

$$\mathsf{req}_{t''}(\rho') \leqslant \mathsf{dist}(\rho', f) - \lambda - 4\delta \leqslant \mathsf{dist}(\rho', f) - 4\delta.$$

Combining all the above we infer that

$$
\begin{aligned}
\mathsf{pred}_{t'}(\rho) &\geqslant \mathsf{dist}(\rho, f) \\
&\geqslant \mathsf{dist}(\rho, \rho') + \mathsf{dist}(\rho', f) - 4\delta \\
&\geqslant \mathsf{dist}(\rho, \rho') + \mathsf{req}_{t''}(\rho'),
\end{aligned}
$$

which establishes (C2) in this case.

Suppose now the second case holds. Since π serves c in R, for any $\pi' \in \Pi_t$ and $f' \in R$, we have

$$\mathsf{dist}(c, \pi) + \mathsf{pred}(\pi) \leqslant \mathsf{dist}(c, \pi') + \mathsf{pred}(\pi')$$

and

$$\mathsf{dist}(c, \pi) + \mathsf{pred}(\pi) \leqslant \mathsf{dist}(c, f').$$

Using the same reasoning as in the first case, but considering expression $\mathsf{dist}(c, \pi) + \mathsf{pred}(\pi)$ instead of $\mathsf{dist}(c, f)$, we infer that

$$\mathsf{pred}_{t'}(\rho) \geqslant \mathsf{dist}(\rho, \pi) + \mathsf{pred}(\pi),$$

which establishes (C2) in this case as well. ⌟

For the next claim, recall that $(C_{t'}^\diamond)_{t' \in \mathsf{chld}(t)}$ form a partition of $C_t^\diamond$.

Claim 6. *Let $c \in C_t^\diamond$ and let $t' \in \mathsf{chld}(t)$ be the unique node satisfying $c \in C_{t'}^\diamond$. Then the following holds.*

$$\mathsf{conn}_{K_{t'}}(c, R_{t'}) \leqslant \mathsf{conn}_{K_t}(c, R) + 5\delta. \qquad (20)$$

PROOF. By the definition of $\mathsf{conn}_{K_t}(c, R)$, there either exists a portal $\pi \in \Pi_t$ such that

$$\mathsf{conn}_{K_t}(c, R) = \mathrm{dist}(c, \pi) + \mathsf{pred}(\pi),$$

or there exists a facility $f \in R$ such that

$$\mathsf{conn}_{K_t}(c, R) = \mathrm{dist}(c, f).$$

Suppose the first case holds. By Lemma 22, there exists a portal $\rho \in \Pi_{t'}$ such that

$$\mathrm{dist}(c, \pi) \geqslant \mathrm{dist}(c, \rho) + \mathrm{dist}(\rho, \pi) - 2\delta.$$

In particular, ρ is facility-important. By the definition of $\mathsf{pred}_{t'}(\rho)$, we have

$$\mathsf{pred}_{t'}(\rho) \leqslant \mathrm{dist}(\rho, \pi) + \mathsf{pred}(\pi) + 3\delta.$$

By combining the above we conclude that

$$
\begin{aligned}
\mathsf{conn}_{K_{t'}}(c, R_{t'}) &\leqslant \mathrm{dist}(c, \rho) + \mathsf{pred}_{t'}(\rho) \\
&\leqslant \mathrm{dist}(c, \rho) + \mathrm{dist}(\rho, \pi) + \mathsf{pred}(\pi) + 3\delta \\
&\leqslant \mathrm{dist}(c, \pi) + \mathsf{pred}(\pi) + 5\delta \\
&= \mathsf{conn}_{K_t}(c, R) + 5\delta;
\end{aligned}
$$

This establishes (20) in this case.

Now suppose the second case holds. Since $(R_{t'})_{t' \in \mathsf{chld}(t)}$ is a partition of R, there exists $t'' \in \mathsf{chld}(t)$ such that $f \in R_{t''}$. If $t'' = t'$, then we have

$$\mathsf{conn}_{K_{t'}}(c, R_{t'}) \leqslant \mathrm{dist}(c, f) = \mathsf{conn}_{K_t}(c, R),$$

so (20) indeed holds in this situation. Assume then that $t'' \neq t'$. By Lemma 22, there exists a portal $\rho \in \Pi_{t'}$ such that

$$\mathrm{dist}(c, f) \geqslant \mathrm{dist}(c, \rho) + \mathrm{dist}(\rho, f) - 2\delta.$$

In particular, ρ is facility-important. By the definition of $\mathsf{pred}_{t'}(\rho)$, we have

$$\mathsf{pred}_{t'}(\rho) \leqslant \mathrm{dist}(\rho, f) + 3\delta$$

By combining the above we conclude that

$$
\begin{aligned}
\mathsf{conn}_{K_{t'}}(c, R_{t'}) &\leqslant \mathrm{dist}(c, \rho) + \mathsf{pred}_{t'}(\rho) \\
&\leqslant \mathrm{dist}(c, \rho) + \mathrm{dist}(\rho, f) + 3\delta \\
&\leqslant \mathrm{dist}(c, f) + 5\delta = \mathsf{conn}_{K_t}(c, R) + 5\delta.
\end{aligned}
$$

Hence, again (20) holds in this case. ⌟

Claim 7. *It holds that*

$$\mathsf{cost}(R; K_t) \geqslant \sum_{t' \in \mathsf{chld}(t)} \mathsf{cost}(R_{t'}; K_{t'}) - 5\delta |C_t^\diamond|.$$

PROOF. The claimed upper bound on $\mathsf{cost}(R; K_t)$ follows by adding the thesis of Claim 6 through all clients $c \in C_t^\diamond$, and adding the opening costs of facilities of R to both sides. ⌟

Claim 8. *For each $t' \in \mathsf{chld}(t)$, $R_{t'}$ is a $(\lambda + 5\delta)$-near feasible $(\gamma + 5\delta)$-close solution to $K_{t'}$.*

PROOF. We first verify the $(\lambda + 5\delta)$-near feasibility. Take any $\rho \in \Pi_{t'}$ with $\mathsf{req}_{t'}(\rho) \neq +\infty$; then ρ is facility-important. By the definition of $\mathsf{req}_{t'}(\rho)$, there exists a facility $f \in R_{t'}$ such that

$$\mathsf{req}_{t'}(\rho) \geqslant \mathrm{dist}(\rho, f) - \lambda - 5\delta,$$

implying

$$\mathrm{dist}(\rho, f) \leqslant \mathsf{req}_{t'}(\rho) + \lambda + 5\delta,$$

as required.

We now verify the $(\gamma + 5\delta)$-closeness. Claim 6 asserts that for each $c \in C_{t'}^\diamond$ we have

$$\mathsf{conn}_{K_{t'}}(c, R_{t'}) \leqslant \mathsf{conn}_{K_t}(c, R) + 5\delta,$$

which by γ-closeness of R implies that

$$\mathsf{conn}_{K_{t'}}(c, R_{t'}) \leqslant \gamma + 5\delta.$$

This is the first condition of the $(\gamma + 5\delta)$-closeness. For the second condition, consider any $\rho \in \Pi_{t'}$ with $\mathsf{req}_{t'}(\rho) \neq +\infty$. In particular, ρ is facility-important, so there exists a facility $f \in R_{t'}$ and either a client $c \in C^\diamond$ served by f such that $\mathrm{dist}(c, \rho) + \mathrm{dist}(\rho, f) \leqslant \mathrm{dist}(c, f) + 4\delta$, or a portals $\pi \in \Pi_t$ served by f such that $\mathrm{dist}(\pi, \rho) + \mathrm{dist}(\rho, f) \leqslant \mathrm{dist}(\pi, f) + 2\delta$. By γ-closeness of R in K, in the first case we have

$$\mathrm{dist}(\rho, f) \leqslant \mathrm{dist}(c, f) - \mathrm{dist}(c, \rho) + 4\delta \leqslant \gamma + 4\delta,$$

while in the second case we have

$$\mathrm{dist}(\rho, f) \leqslant \mathrm{dist}(\pi, f) - \mathrm{dist}(\pi, \rho) + 2\delta \leqslant \gamma + 2\delta.$$

In both cases, we conclude that $\mathrm{dist}(\rho, f) \leqslant \gamma + 5\delta$, as required. ⌟

Claims 4, 5, 7, and 8 conclude the proof. ∎

The algorithm: We are finally ready to present the whole algorithm. First, using the algorithm of Lemma 20 in polynomial time we compute the tree T together with sets $\beta(t)$ for nodes t of T. For each node t we compute the portal set Π_t and the set of functions $\mathcal{N}_t$, as explained before; this takes total time $n^{\mathcal{O}(\varepsilon^{-2}r)}$, since T is of size $n^{\mathcal{O}(1)}$. Sets $\mathcal{N}_t$ give rise to sets $\mathcal{U}_t$ and $\mathcal{W}_t$ as defined before.

The remaining, main part of the algorithm is summarized using pseudo-code as Algorithm `Solve`. We process the nodes of T in a bottom-up manner. For each node t, say at depth i, and each $\eta \in \mathcal{U}_t$, we construct the instance $K_t(\eta)$ and compute an 5ε-near feasible solution $R_{t,\eta}$ to it as follows. If t is a leaf, we use the algorithm of Corollary 26 to compute the least expensive 5ε-near feasible solution $R_{t,\eta}$. Otherwise, we iterate over all $\varphi \in \mathcal{W}_t$ such that η and φ are compatible, and consider all candidate solutions $R(\varphi)$ defined as

$$R = \bigcup_{t' \in \mathsf{chld}(t)} R_{t', \mathsf{restrict}_{t,t'}(\varphi)}.$$

Here, $R_{t', \mathsf{restrict}_{t,t'}(\varphi)}$ is the pre-computed soluton to the instance $K_{t'}(\mathsf{restrict}_{t,t'}(\varphi))$. Out of these candidate solutions we take the least expensive one and we declare it as $R_{t,\eta}$.

Finally, we return $R = R_{t_0,(\emptyset,\emptyset)}$ as computed solution, where t_0 is the root of T. This concludes the description of the algorithm and we are left with analyzing its running time and approximation guarantee.

Algorithm 1: Algorithm `Solve`

Input: Instance J_j, tree T, sets $\mathcal{U}_t, \mathcal{W}_t$ for nodes t of T
Output: Solution R to J_j

> **for each** *node t of T in bottom-up order* **do**
> > **for each** $\eta \in \mathcal{U}_t$ **do**
> > > **if** *t is a leaf* **then**
> > > > $R_{t,\eta} \leftarrow$ minimum-cost 5ε-near feasible solution to $K_{t,\eta}$, computed using Corollary 26
> > >
> > > **else**
> > > > $R_{t,\eta} \leftarrow \perp$
> > > > **for each** $\varphi \in \mathcal{W}_t$ *such that* $\eta \sim \varphi$ **do**
> > > > > $S \leftarrow \bigcup_{t' \in \mathsf{chld}(t)} R_{t',\mathsf{restrict}_{t,t'}(\varphi)}$
> > > > > **if** $R_{t,\eta} = \perp$ *or*
> > > > > $\mathsf{cost}(S, K_{t,\eta}) < \mathsf{cost}(R_{t,\eta}, K_{t,\eta})$
> > > > > **then**
> > > > > > $R_{t,\eta} \leftarrow S$

> $R \leftarrow R_{t_0,(\emptyset,\emptyset)}$, where t_0 is the root of T
> **return** R

Lemma 30. *Algorithm* `Solve` *runs in time* $n^{\mathcal{O}(\varepsilon^{-2}r)}$.

Proof: It suffices to observe that, by Corollary 26 and Lemma 28, the time spent on processing every node of T is bounded by $n^{\mathcal{O}(\varepsilon^{-2}r)}$. Since the number of nodes of T is $n^{\mathcal{O}(1)}$, the total running time follows. $\blacksquare$

Lemma 31. *Algorithm* `Solve` *returns a solution R to the instance J_j satisfying*

$$\mathsf{cost}(R; J_j) \leqslant \mathsf{OPT}(J_j) + 10\varepsilon|C_j|.$$

Proof: Let $D \subseteq F_j$ be an optimum solution to the instance J_j. By Lemma 24, D is also an optimum feasible solution to the instance $K = K_{t_0}((\emptyset,\emptyset))$, where t_0 is the root of T, Furthermore, by Lemma 19 we infer that D is $3r$-close in K.

By applying Lemma 29 in a top-down manner along the tree T, we obtain, for every node t of T, an element $\eta_t \in \widetilde{\mathcal{U}}_t$ and a solution D_t to the instance $K_t(\eta_t)$ such that the following holds:

- whenever t is not a leaf, we have that $\varphi_t = (\eta_{t'})_{t' \in \mathsf{chld}(t)}$ is compatible with η_t;
- D_t is a $(5i\delta)$-near feasible $(3r+5i\delta)$-close solution in $K_t(\eta_t)$, where i is the depth of t in T;
- all request and prediction functions involved in η_t are $(\delta, -5i\delta, 3r+5i\delta, \delta)$-normal, and all prediction functions are nonnegative;

- whenever t is not a leaf, it holds that
$$\mathsf{cost}(D_t; K_t(\eta_t)) \geqslant$$
$$\sum_{t' \in \mathsf{chld}(t)} \mathsf{cost}(D_{t'}, K_{t'}(\eta_{t'})) - 5\delta|C_t^\diamond|. \qquad (21)$$

Recall that T has depth at most $\log n$. Therefore, $5i\delta \leqslant 5\varepsilon$ whenever i is the depth of a node in t, implying that all request and prediction functions involved in elements η_t are $(\delta, -5\varepsilon, 3r+5\varepsilon, \delta)$-normal. We infer that

$$\eta_t \in \mathcal{U}_t \qquad \text{for each node } t. \qquad (22)$$

Recall also that for each non-leaf node t of T, we have that $\{C_{t'}^\diamond : t' \in \mathsf{chld}(t)\}$ form a partition of $C_t^\diamond$. Therefore, by combining inequalities (21) in a bottom-up manner along T we infer that

$$\mathsf{cost}(D; K)$$
$$\geqslant \sum_{t:\,\text{leaf of } T} \mathsf{cost}(D_t, K_t(\eta_t)) - 5\delta \log n |C_j|$$
$$= \sum_{t:\,\text{leaf of } T} \mathsf{cost}(D_t, K_t(\eta_t)) - 5\varepsilon|C_j|. \qquad (23)$$

Again, as $i\delta \leqslant \varepsilon$ whenever $i \leqslant \log n$, for each leaf t of T the solution D_t is 5ε-near feasible in $K_t(\eta_t)$. Hence, due to (22) for each leaf t the algorithm computes an 5ε-near feasible solution R_t to $K_t(\eta_t)$ satisfying

$$\mathsf{cost}(R_t; K_t(\eta_t)) \leqslant \mathsf{cost}(D_t; K_t(\eta_t)). \qquad (24)$$

For each non-leaf node t of T, define solution R_t to instance $K_t(\eta_t)$ by a bottom-up induction: $R_t = \bigcup_{t' \in \mathsf{chld}(t)} R_{t'}$. Then by (22) and the fact that $\eta_t \sim \varphi_t$ for every non-leaf t, we have that for each node t, the algorithm computes a solution to η_t of cost at most $\mathsf{cost}(R_t; K_t(\eta_t))$. In particular, if we denote $R = R_{t_0}$, where t_0 is the root of T, then the solution returned by the algorithm has cost at most $\mathsf{cost}(R; K)$. Hence, we proceed with upper bounding $\mathsf{cost}(R; K)$.

For each node t of T let us define tuples of functions η_t' and φ_t' (here, only when t is not a leaf) as follows:

$$\eta_t' = \eta_t + 5\varepsilon \qquad \text{and} \qquad \varphi_t' = \varphi_t + 5\varepsilon.$$

That is, η_t' is obtained from η_t by adding 5ε to all requests and all predictions on all portals of Π_t, and similarly for φ_t. Note that for each non-leaf node t of T, we still have the following properties:

- $\eta_{t'}' = \mathsf{restrict}_{t,t'}(\varphi_t')$ for each $t' \in \mathsf{chld}(t)$, and
- η_t' and φ_t' are compatible.

However, the 5ε shift in requests and predictions makes the following assertion hold for each leaf t of T:

$$R_t \text{ is a feasible solution to } K_t(\eta_t') \text{ with}$$
$$\mathsf{cost}(R_t; K_t(\eta_t')) \leqslant \mathsf{cost}(R_t; K_t(\eta_t)) + 5\varepsilon|C_t^\diamond|. \qquad (25)$$

That is, we obtained feasibility instead of 5ε-near feasibility at the cost of increasing the cost of the solution.

Denoting $\mathsf{desc}(t)$ the set of leaves of T that are descendants of t, we may now apply Lemma 28 through a bottom-up induction along the tree T to infer the following for each node t of T:

R_t is a feasible solution to $K_t(\eta'_t)$ with

$$\mathsf{cost}(R_t; K_t(\eta'_t)) \leqslant \sum_{t' \in \mathsf{desc}(t)} \mathsf{cost}(R_{t'}; K_{t'}(\eta'_{t'})). \quad (26)$$

In particular, assertion (26) holds for the root t_0 of T. Then, we may use assertions (23), (24), and (25) to infer the following:

$$\begin{aligned}
\mathsf{cost}(R; K) &\leqslant \sum_{t:\, \text{leaf of } T} \mathsf{cost}(R_t; K_t(\eta'_t)) \\
&\leqslant \sum_{t:\, \text{leaf of } T} \mathsf{cost}(R_t; K_t(\eta_t)) + 5\varepsilon|C_j| \\
&\leqslant \sum_{t:\, \text{leaf of } T} \mathsf{cost}(D_t; K_t(\eta_t)) + 5\varepsilon|C_j| \\
&\leqslant \mathsf{cost}(D; K) + 10\varepsilon|C_j|.
\end{aligned}$$

It now suffices to use Lemma 24 to infer that $\mathsf{cost}(R; K) = \mathsf{cost}(R; J_j)$ and $\mathsf{cost}(D; K) = \mathsf{cost}(D; J_j)$; this combined with the above concludes the proof. ∎

We now conclude the proof of Lemma 9. Apply Algorithm Solve to each instance J_j for which C_j is non-empty, yielding a solution R_j. As the number of such instances is at most n, by Lemma 30 this takes total time $n^{\mathcal{O}(\varepsilon^{-2}r)}$. As the final solution return $R = S \cup \bigcup_{j \in \mathbb{N}} R_j$, where we set $R_j = \emptyset$ whenever $C_j = \emptyset$. Then, by Lemmas 18 and 31 we have

$$\begin{aligned}
\mathsf{cost}(R; J') &\leqslant \varepsilon \cdot M + \sum_{j \in \mathbb{N}} \mathsf{cost}(R_j; J_j) \\
&\leqslant \varepsilon \cdot M + 10\varepsilon \cdot |C| + \sum_{j \in \mathbb{N}} \mathsf{OPT}(J_j) \\
&\leqslant \mathsf{OPT}(J') + \varepsilon \cdot M + 10\varepsilon \cdot |C|.
\end{aligned}$$

Finally, we observe that since $\mathrm{dist}(c, f) \geqslant 1$ for each client $c \in \mathsf{cluster}(f)$, we have

$$|C| \leqslant \sum_{f \in D^\circ} \sum_{c \in \mathsf{cluster}(f)} \mathrm{dist}(c, f) \leqslant M.$$

Therefore, we conclude that

$$\mathsf{cost}(R; J') \leqslant \mathsf{OPT}(J') + 11\varepsilon \cdot M.$$

It now remains to apply Corollary 14 to infer the same inequality for instance J instead of J', and to rescale ε by a multiplicative factor of 11.

References

[1] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit, "Local search heuristics for k-median and facility location problems," *SIAM J. Comput.*, vol. 33, no. 3, pp. 544–562, 2004. [Online]. Available: http://dx.doi.org/10.1137/S0097539702416402

[2] D. S. Hochbaum, "Heuristics for the fixed cost median problem," *Math. Program.*, vol. 22, no. 1, pp. 148–162, 1982. [Online]. Available: http://dx.doi.org/10.1007/BF01581035

[3] D. B. Shmoys, É. Tardos, and K. Aardal, "Approximation algorithms for facility location problems," in *STOC 1997*. ACM, 1997, pp. 265–274.

[4] K. Jain and V. Vazirani, "Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation," *J. ACM*, vol. 48, no. 2, pp. 274–296, 2001. [Online]. Available: http://doi.acm.org/10.1145/375827.375845

[5] S. Li, "A 1.488 approximation algorithm for the uncapacitated facility location problem," *Inf. Comput.*, vol. 222, pp. 45–58, 2013.

[6] S. Guha and S. Khuller, "Greedy strikes back: Improved facility location algorithms," *J. Algorithms*, vol. 31, no. 1, pp. 228–248, 1999. [Online]. Available: http://dx.doi.org/10.1006/jagm.1998.0993

[7] V. Cohen-Addad, P. N. Klein, and C. Mathieu, "Local search yields approximation schemes for k-means and k-median in Euclidean and minor-free metrics," in *FOCS 2016*. IEEE Computer Society, 2016, pp. 353–364.

[8] T. M. Chan and S. Har-Peled, "Approximation algorithms for maximum independent set of pseudo-disks," *Discrete & Computational Geometry*, vol. 48, no. 2, pp. 373–392, 2012.

[9] E. Fox-Epstein, P. N. Klein, and A. Schild, "Embedding planar graphs into low-treewidth graphs with applications to efficient approximation schemes for metric problems," in *SODA 2019*, 2019, pp. 1069–1088. [Online]. Available: https://doi.org/10.1137/1.9781611975482.66

[10] D. Eisenstat, P. N. Klein, and C. Mathieu, "Approximating k-center in planar graphs," in *SODA 2014*, 2014, pp. 617–627. [Online]. Available: https://doi.org/10.1137/1.9781611973402.47