

Socially Fair and Hierarchical Facility Location Problems

Swati Gupta¹, Jai Moondra¹, and Mohit Singh¹

¹Georgia Institute of Technology

{swatig, jmoondra3}@gatech.edu, mohit.singh@isye.gatech.edu

December 12, 2022

Abstract

The classic facility location problem seeks to open a set of facilities to minimize the cost of opening the chosen facilities and the total cost of connecting all the clients to their nearby open facilities. Such an objective may induce an unequal cost over certain socioeconomic groups of clients (i.e., total distance traveled by clients in such a group). This is important when planning the location of socially relevant facilities such as emergency rooms. In this work, we consider a *fair* version of the problem by minimizing the Minkowski p -norm of the total distance traveled by clients across different socioeconomic groups and the cost of opening facilities, to penalize high access costs to open facilities across r groups of clients. This generalizes classic facility location ($p = 1$) and the minimization of the maximum total distance traveled by clients in any group ($p = \infty$). We show that there is a small portfolio of solutions where for any norm, at least one of the solutions is a constant-factor approximation with respect to any p -norm. We also give a lower bound on the cardinality of such portfolios. We also introduce the notion of *weak* and *strong* refinements for the facility location problem, that imposes further structural properties across solutions. The former requires that the set of facilities open for a lower p -norm is a superset of those open for higher p -norms, and the latter further imposes a partition refinement over the assignment of clients to open facilities in different norms. We give an $O(1)$ -approximation for weak refinements, $\text{poly}(r^{1/\sqrt{\log r}})$ -approximation for strong refinement in general metrics and $O(\log r)$ -approximation for the tree metric, where r is the number of (disjoint) client groups.

Our techniques also generalize to hierarchical versions of the facility location problem. Here we are given multiple instances of facility location problems on the same network that differ only in facility opening costs that decrease in the hierarchy. The goal is to obtain a family of solutions, one for each instance, where an opened facility in a solution remains open in instances with lower cost. We give a $\text{poly}(e^{\sqrt{l}})$ -approximation for the problem with arbitrary metrics where l is the number of instances in the hierarchy. We also give $O(l^2)$ -approximation for the problem on line and tree metrics.

1 Introduction

In recent decades, optimization and algorithm design have revolutionized numerous industrial sectors – ranging from supply chains and network design to food production and finance. Novel optimization techniques have led to many industries achieving previously inconceivable levels of efficiency. The race to efficiency, however, has led to an inequitable distribution of resources among different segments of the population. For instance, the goal of profit maximization by grocery stores has led to the formation of food deserts in poorer parts of the United States [16]. Furthermore, a continuation of this trend can potentially lead to the formation of medical deserts [42]. It is clear that the pursuit of maximizing a singular objective can lead to crucial social and ethical problems, and therefore, it is imperative that optimization research studies a wider range of objectives that are important from a fairness viewpoint. In recent years, fairness has been extensively studied from an optimization and algorithm design perspective [7, 33, 35, 23], in both theory and application.

There are two broad ways explored in the current literature to make constrained optimization problems fairer: one is to incorporate equity or fairness metrics as the optimization objective, and the other is to constrain solutions so they satisfy a minimum desired level of fairness. Further, there are broadly two different notions of fairness in optimization – individual and group fairness (e.g., [18, 17]). In this work, we focus on various fairness objectives under the same set of constraints for the (uncapacitated) facility location problem. We are concerned with group fairness, wherein each member of the population belongs to a single group.

The uncapacitated facility location problem is one of the most well-studied combinatorial optimization problems. In the classic version of the problem, we have a set of demands or clients D , a set of facilities F , and distances d_{ij} between each facility $i \in F$ and client $j \in D$ that satisfy the triangle inequality. Opening facility $i \in F$ incurs a cost c_i . The objective is to open a set of facilities and assign each client to some open facility such that the total cost of opening the facilities and the distances of clients from their assigned facilities is minimized. The facility location problem models various network design situations such as opening offices, warehouses, hospitals, proxy servers, transport hubs etc. The simple and elegant facility location problem has received significant attention in theory and practice [29, 15, 43, 12, 13, 36].

A solution to the uncapacitated facility location problem can be disproportionately bad for a certain set of clients while being optimal with respect to the overall cost. As a simple example, consider clients on a line where a facility of cost $m \in \mathbb{Z}_+$ can be opened anywhere on the line, and there are m clients at 0 and $m - 1$ clients at 1; the optimal solution to the classic facility location opens a single facility at 0. This disproportionately affects the set of clients at 1.

As discussed, this is particularly undesirable in social applications where underprivileged groups of clients can be disadvantaged by the optimal solution. As a further example, a recent study on emergency rooms in Alameda County showed that the closure of the Alta Bates Emergency Room would disproportionately displace the uninsured population (clients in facility location terminology)

and people of color [14].

To model such phenomenon, we study the *socially fair* version of the facility location problem, where we are given several *client groups* that partition the client set, and the total cost of a solution is a function of the total distances of each group. Formally, the socially fair facility location problem is defined as the following:

Definition (Fair facility location). We are given a metric space $\mathcal{M} = (X, d)$, client set $D \subseteq X$, and non-empty set of facilities $F \subseteq X$ with non-negative facility costs $c : F \rightarrow \mathbb{R}_+$. Furthermore, suppose we are given non-empty client groups D_1, \dots, D_r that partition D . For $p \in [1, \infty]$, the p -norm fair facility location problem requires opening a subset of facilities $F' \subseteq F$ and providing a client to facility assignment $\Pi : D \rightarrow F$ so as to minimize the cost of open facilities and the Minkowski p -norm of average group distances, i.e.,

$$\min \sum_{i \in F'} c_i + \left[\sum_{s \in [r]} \left(\frac{1}{|D_s|} \sum_{j \in D_s} d(j, \Pi(j)) \right)^p \right]^{1/p}.$$

For $p = 1$ and singleton groups (i.e., $|D_s| = 1$ for all $s \in [r]$), the fair facility location problem reduces to the aforementioned classic facility location problem. For $p = \infty$, the objective corresponds to minimizing the maximum distance across all groups, weighted by their group size. This variant has been studied generally [10, 41] and in specific contexts (for example, emergency rooms placement [14]).

We remark that while we have defined our objective for *average* distance to open facilities for each group, our results hold for a much larger class of objectives, including total distance to open facilities for each group.

Our primary goal is not restricted to getting a constant-factor approximation for the p -norm fair facility location problem: indeed, that can be achieved by a relatively straightforward generalization of Shmoys *et al.*'s algorithm [43] for the classic facility location problem. Instead, we aim to understand how approximate solutions change as p varies, and if common combinatorial structure can be preserved between these approximate solutions for various norms. As we shall see, this is useful from a practical policy design perspective, and interesting from a theoretical point of view. This is also an interesting question from a multi-criteria optimization viewpoint: our problem is an instance of the broader question about what common structures do optimal (or approximate) solutions maintain for related objectives.

Contributions. As a warm-up to the fair facility location, we show that the 4-approximation algorithm of Shmoys *et al.* [43] for the classic facility location problem generalizes to a large class of convex objective functions, and in particular gives a 4-approximation to the p -norm fair facility

location problem for each $p \in [1, \infty]$; see Section 1.3 and Appendix A*. This observation is useful in showing our key contributions, explained next.

1. **Portfolio of Solutions.** A challenge in the fair facility location problem is deciding what p -norm to choose to model a fair objective. The chosen norm allows to weigh the connection cost paid by different socioeconomic groups and appropriate weighing may not be clear even to the algorithm designer. Indeed, for classification algorithms, it has been shown that different fairness objectives may not be simultaneously satisfied [33]. Building portfolios of small solutions can be useful for policymakers, whose decision is not really to decide a fairness criterion, but rather to *weigh the properties of suggested portfolio of options* before making a choice of the solution. For example, United Network for Organ Sharing (UNOS) selects the yearly policy for kidney transplants after carefully considering a portfolio of candidate policies and evaluating them on multiple axes of interest [1]. This problem of portfolio design for multi-criteria approximations has not received enough attention from the theory community.

Therefore, given an instance of the fair facility location problem, a natural question we consider is the following: *is an optimal solution to the p -norm fair facility problem a good approximation to the q -norm fair facility problem for different $p, q \in [1, \infty]$?* This is not always the case; indeed, the cost of an optimal solution to the p -norm problem can be high ($\Omega(r)$, where r is the number of client groups) for the q -norm problem. This raises the following question: *is there a small set of candidate solutions such that no matter what norm is chosen, one of the solutions in the set is a good approximation corresponding to that norm.* Empirical studies for the facility location problem suggest that this might possibly be true even for an infinite class of objective functions [28].

Upper bound: We show in Section 2, that when there are r client groups, there exist (approximately) $\log_2 r$ solutions such that for each $p \in [1, \infty]$, one of these solutions is an 8-approximation for the p -norm fair problem (Theorem 2); our proof builds on ideas similar to those in [26]. This is important in context of policy design: the task of deciding what is ‘fair’ can be left to policy makers along with a small set of candidate solutions that are a good approximation irrespective of what definition of fair is chosen.

Lower bound: Moreover, we show such a dependence on r is *necessary*. In particular, we show that there exist problem instances where at least $\tilde{\Omega}(\sqrt{\log_2 r})^\dagger$ different solutions are needed such that one of them is an $O(1)$ -approximation for any p -norm (Theorem 3).

2. **Refinements.** Additionally, we introduce the notion of *refinements* which establish further structure between approximate solutions across different p -norms, $p \in [1, \infty]$.

Given an instance of the facility location problem, for weak refinements we demand that for any two norms $p < q$, we must have $F_q \subseteq F_p$ where F_p, F_q are the set of facilities in the solution

*We remark that several other improved algorithms for the classic facility location problem do not generalize for more general convex objectives because they crucially use the linearity of the objective.

[†]We say $f(r) = \tilde{\Omega}(\sqrt{\log_2 r})$ if and only if $\sqrt{\log_2 r} = O(f(r) \cdot \text{poly}(\log \log r))$, i.e., we suppress $\log \log r$ terms.

	Arbitrary metric	Line metric	Tree metric [‡]
weak refinement	16		
strong refinement	$\text{poly}(r^{\frac{1}{\sqrt{\log r}}})$	$O(\log r)$	$O(\log r)$

Table 1: A summary of approximation guarantees for refinements for the socially fair facility location problem in Section 3. Here r is the number of client groups in the problem instance.

for p, q -norm fair problems respectively. This also establishes a decreasing subset structure for open facilities across solutions for p -norms as p varies from 1 to ∞ . Therefore, if the fairness criterion changes at some point, a completely different set of facilities does not have to be open or closed.

For strong refinements, we additionally demand that the set of assigned clients to any facility in F_p must be assigned to a single open facility in F_q ($p < q$). Thus the partition of client set obtained by their assignment to open facilities in solution for norm q must be a refinement of the partition obtained in the solution for norm p . From a planning perspective, strong refinements induce a nice hierarchical structure over the potential set of solutions which can eventually help policy makers in growing the number of facilities that serve a set of customers, as the need arises.

We formalize this notion in 1.1; Section 3 deals with approximation algorithms for refinements. For arbitrary metrics, we give an $O(1)$ -approximation for weak refinement (Theorem 4) and a $\text{poly}\left(r^{\frac{1}{\sqrt{\log r}}}\right)$ -approximation for strong refinement (Section 3.1, Corollary 2). For the special cases of line and tree metrics, we improve the approximation factor to $O(\log r)$ for strong refinement (Section 3.2, Corollary 3 and Appendix E, Corollary 5 respectively). A summary of these results is presented in Table 1.

3. **Trade-off Between Facility Opening Costs and Connection Costs:** So far, we considered an objective that adds the cost of opening facilities and the p -norm of client group assignment cost as a way to balance the two costs. However, as p changes, different trade-offs between the two costs might be desirable. In Section 4, we introduce a more general objective that models this trade-off as a normalizing factor $g(p; r)$ for the group assignment costs. Formally, given a norm p , the cost of a solution with open facility set F' and assignment Π for r client groups D_1, \dots, D_r is given by

$$\min \sum_{i \in F'} c_i + g(p; r) \left[\sum_{s \in [r]} \left(\frac{1}{|D_s|} \sum_{j \in D_s} d(j, \Pi(j)) \right)^p \right]^{1/p}.$$

We show that there the objective function is nonincreasing in p when g is non-increasing in p , and that it is nondecreasing when $r^{1/p}g$ is nondecreasing in p ; further, these are the only cases when monotonicity is guaranteed (see Theorem 7).

4. **Hierarchical Facilities.** Inspired by the notion of refinements, we next introduce a version

of the *hierarchical facility location* problem. In several applications of the facility location problem such as the placement of corporate offices, judicial courts, hospitals, public works offices, facilities need to be opened at several hierarchical levels with additional constraints across those levels [19, 50]. For instance, public works offices usually operate at several levels, with the office at the lowest level being the first point of contact for an applicant (i.e., a client in facility location terms). For l hierarchical levels, our version of the hierarchical facility location problem demands solutions for each of the l levels of the given instance of the problem that satisfy certain hierarchic constraints motivated by the discussion above, while also being a good approximation for the independent facility location problems at their respective levels. We formalize this definition in Section 1.1. In Section 5, we show that our techniques for refinements also lead to results for hierarchical facility location, giving an $\tilde{O}(e^{3\sqrt{l}})$ -approximation for general metrics, and an $O(l^2)$ -approximation for tree metrics (Theorem 10).

We define the various problems formally in Section 1.1. We give a brief literature survey for classic, fair, and hierarchical facility location problems in Section 1.2. Finally, we include the generalization of Shmoys *et al.*'s algorithm [43] as a preliminary in Section 1.3 before proceeding to the key results.

1.1 Problem definitions and notation

Before we consider the special case of the fair facility location problem, we define a more general version of the facility location problem, where the objective is defined by a suitable convex function. Given a convex, non-decreasing function $f : \mathbb{R}_+^D \rightarrow \mathbb{R}_+$, we minimize the sum of facility costs and f applied to individual client group distances. The classic facility location corresponds to the case where $f(x) = \sum_j x_j$ for $x \in \mathbb{R}_+^D$, i.e., the total assignment cost is the sum of distances of individual clients. Furthermore, the p -norm facility location problem

Definition (Generalized facility location). We are given a metric space $\mathcal{M} = (X, d)$, client set $D \subseteq X$, and non-empty facility set $F \subseteq X$ with non-negative facility costs $c : F \rightarrow \mathbb{R}$. Furthermore, we are given a non-decreasing convex function $f : \mathbb{R}_+^D \rightarrow \mathbb{R}_+$. The general facility location problem requires opening a subset of facilities $F' \subseteq F$ and providing an assignment $\Pi : D \rightarrow F'$ so as to minimize

$$\underbrace{\sum_{i \in F'} c_i}_{\text{Facility cost } C_{F'}} + f\left(\underbrace{\left(d(j, \Pi(j))\right)_{j \in D}}_{\text{Assignment cost } C_{\Pi}}\right),$$

where $\left(d(j, \Pi(j))\right)_{j \in D}$ is the vector in \mathbb{R}^D consisting of individual client distances.

For any instance of the facility location problem, a solution \mathcal{S} is an ordered pair (F', Π) where $F' \subseteq F$ is the set of open facilities and $\Pi : D \rightarrow F'$ is the assignment function. The *total cost* of the solution, $C_{\mathcal{S}}$ is the sum of the *facility cost* $C_{F'}$ and the *assignment cost* C_{Π} , that is, $C_{\mathcal{S}} = C_{F'} + C_{\Pi}$.

If f is non-decreasing, then each client is assigned to its nearest open facility for any optimal solution. Therefore, a solution is sometimes identified just by the set of open facilities in facility location literature. However, we will deal extensively with different assignments and always explicitly give the assignment function.

We discuss two different notions of refinements for the p -norm fair facility location problem. *Weak (or facility) refinements* preserve the structure for open facility sets across p , by ensuring that any facility that was open for a higher p -norm remains open for lower p -norms as well. *Strong (or partition) refinements* are weak refinements that also preserve structure for the corresponding assignments, by ensuring that any customer that was mapped to a certain facility in a lower p -norm, must be mapped to the same facility (if it is open) in the higher p -norm. We can formalize these notions as follows:

Definition (Refinements). We are given a metric space $\mathcal{M} = (X, d)$, client set $D \subseteq X$, and non-empty facility set $F \subseteq X$ with non-negative facility costs $c : F \rightarrow \mathbb{R}$. We are given a metric space $\mathcal{M} = (X, d)$, client set $D \subseteq X$, and non-empty facility set $F \subseteq X$ with non-negative facility costs $c : F \rightarrow \mathbb{R}$. Furthermore, suppose we are given non-empty client groups D_1, \dots, D_r that partition D . Given a set of norms $P \subseteq [1, \infty]$, a solution set $S = \{\mathcal{S}_p = (F_p, \Pi_p) : p \in P\}$ is called

1. a *weak refinement* if for all $p, q \in P$ with $p \leq q$, $F_p \supseteq F_q$; that is, the facilities form a subset structure.
2. a *strong refinement* if for all $p, q \in P$ with $p \leq q$ and for each facility $i \in F_p$, there is some facility $i' \in F_q$ such that all clients assigned to i under Π_p are assigned to i' under Π_q , that is,

$$\{j \in D : \Pi_p(j) = i\} \subseteq \{j \in D : \Pi_q(j) = i'\}.$$

If for each norm $p \in P$, the solution \mathcal{S}_p is an α -approximation for the p -norm fair problem, then we say that the refinement is an α -approximate refinement.

One can think of refinements as dynamic solutions to an instance of the fair facility location problem that change with p . As $p \in P$ decreases, weak refinements *open* more facilities but do not close any open facilities. For strong refinements, this can be interpreted in terms of the partition of D that Π_p induces. As p decreases and new facilities are potentially added, this partition of D is refined. See Figure 1 for a toy example. As described earlier, refinements demand some common structure across different p -norms (i.e., across different fairness criteria).

Finally, we discuss the hierarchical facility location problem. Recall the assumptions: we wish to solve several facility location problems at different levels in a hierarchy. The cost of opening a facility at a level increasing with the level. Furthermore, each open facility at a given level is also open at a lower level. Finally, all clients of a facility at a given level must be clients of the same facility at a higher level - this condition is often true for most hierarchical structures such as corporate offices and speciality hospitals. We present the formal definition.

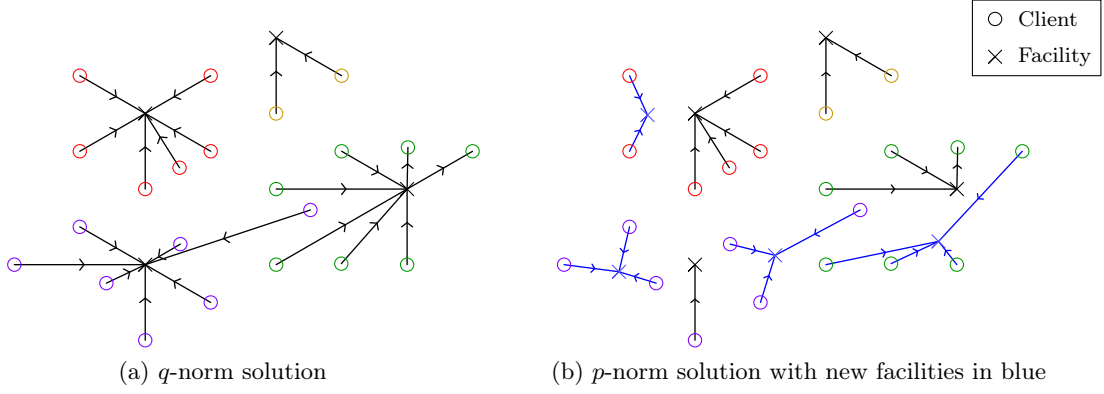


Figure 1: A toy example to illustrate refinements. Consider $P = \{p, q\}$ with $q > p$. (a) A solution for the q -norm problem with open facilities marked as crosses and the assignment function indicated by arrows; clients assigned to the same facility have the same color for clarity. (b) A solution for the p -norm problem: this solution refines the q -norm solution, with new facilities and new assignment arrows in blue. Note that no facility has been deleted from the q -norm solution, and all clients assigned to a new facility are assigned to the same facility in the q -norm solution (i.e., all clients assigned to the same facility still have the same color).

Definition (Hierarchical facility location). We are given a metric space $\mathcal{M} = (X, d)$, and non-empty facility set $F \subseteq X$ and demand set $D \subseteq X$, and l different general facility location problems for some positive integer l , one for each *level* in $[l]$. The l problems are specified by uniform facility opening costs $c_1, \dots, c_l \in \mathbb{R}_+$ satisfying $c_1 \leq \dots \leq c_l$; and a non-negative, non-decreasing, convex function $f : \mathbb{R}_+^D \rightarrow \mathbb{R}$. In the hierarchical facility location problem, we are asked to provide solutions $\{\mathcal{S}_k = (F_k, \Pi_k) : k \in [l]\}$ such that for all $k, t \in [1, l]$ with $t \geq k$:

1. $F_k \supseteq F_t$, and
2. for all $i \in F_k$, there exists $i' \in F_t$ such that

$$\{j \in D : \Pi_p(j) = i\} \subseteq \{j \in D : \Pi_q(j) = i'\}.$$

The objective is to minimize the worst approximation ratio among l levels, that is, $\max_{k \in [l]} \frac{C_{\mathcal{S}_k}}{OPT_k}$, where OPT_k is the optimal solution value for the generalized facility location problem with facility costs c_k . We call this an l -level hierarchical facility location problem.

It should be clear that the hierarchical facility location is similar to the notion of refinements for the fair facility location problem: the constraints on open facility sets and assignments are similar. This is precisely why our algorithms work for both refinements and hierarchical facility location.

Throughout, we denote $n = |F \cup D|$. We assume that the input always has size at least n : the points in F and D always need to be explicitly specified; the distance function d can be specified implicitly or explicitly. For instance, when the metric space \mathcal{M} is the Euclidean space \mathbb{R}^2 , the points in F and D need to be specified as 2-tuples of real numbers while the distance d is implicit. Therefore, all algorithms that run in polynomial time in n are polynomial in the input size.

We usually denote facilities using the letter i and occasionally using the letters f, g, h . Clients are usually denoted by j . The letters k, l usually denote indices for different p -norms or levels in the hierarchical facility location problem; occasionally letters s, t are also used for this. Solutions (i.e., pairs of facilities and assignments) are usually denoted by letters \mathcal{S}, \mathcal{R} . Subsets of the facility set F are denoted by letters F, G . Assignments of clients to facilities (or facilities to facilities at different levels in a hierarchical problem) are denoted by letters Π and θ .

1.2 Literature review

We present a brief review of the literature for classic, fair, and hierarchical facility location problems. The classic facility location problem has been extensively studied. An $O(\log n)$ -approximation algorithm was given by Hochbaum [29]; notably, this algorithm does not assume the metric property. The first constant-factor approximation was given by Shmoys *et al.* [43] who gave a deterministic 4-approximation algorithm. They also give a randomized 3.16-approximation algorithm. Since then, a steady stream of algorithms has led to frequent improvements in this approximation factor [34, 12, 13, 30, 38, 11]. The state-of-the-art is a 1.488-approximation algorithm of Li [36]; while the problem is inapproximable to within a factor 1.463 unless $P = NP$ [27, 45]. The textbook by Shmoys and Williamson [48] gives an overview of and more details about many of the techniques used.

Several fairness objectives for the facility location problem have been studied [31, 21, 46]; for an extensive list see the survey articles [40, 6]. To the best of our knowledge, none of these fairness objectives have been studied from an approximation algorithm viewpoint for the uncapacitated facility location problem. However, several approximation algorithms are known for the fair versions of the related clustering problem where a fixed number of facilities are allowed to open [31, 23, 3, 39, 24, 26, 25].

Minkowski p -norm objectives have been widely considered in combinatorial optimization literature as a model for fairness and as interesting theoretical questions. Golovin *et al.* [26] develop a general technique for p -norm optimization that is closely related to our technique in section 2.1. Other methods are often problem-specific: for instance, k -clustering [23, 24, 25, 26], traveling salesman problem [20], set cover, scheduling and other problems [26] have all been studied with the p -norm objective.

There are a very large number of models for hierarchical facility location problems; see survey articles [50, 19]. Many of these models have not been studied from an approximation algorithm viewpoint. However, there are several models with an approximation guarantee for the overall cost [2, 9, 4, 5, 49, 32, 22, 47]. Almost all of these models aim to minimize some notion of the total cost or maximize some notion of client coverage that is linear in open facilities and assignments. To the best of our knowledge, the model with a facility location problem at each level with minimization of the worst approximation ratio across levels has not been proposed so far.

1.3 Preliminaries

We prove the following theorem, which asserts that the generalized facility location problem can be 4-approximated for a large class of functions f . Specifically, we show that Shmoys *et al.*'s algorithm [43] – which uses the filtering technique introduced by Lin and Vitter [37] – extends to this class of functions. For completeness, we include the algorithm and the proof of the theorem in Appendix A.

Before we state the theorem, we recall the definition of *sublinear* functions: f is sublinear if for all x, y in its domain (1) $f(cx) = cf(x)$ for all $c \geq 0$ and (2) $f(x + y) \leq f(x) + f(y)$. All norms are sublinear and all sublinear functions are convex. We are now ready to state the theorem:

Theorem 1. *If $f : \mathbb{R}_+^D \rightarrow \mathbb{R}$ is a differentiable sublinear function that is non-negative and non-decreasing, then there is 4-approximation to the generalized facility location problem with a polynomial number of oracle calls to $f, \nabla f$.*

Since all p -norm functions are sublinear, the theorem immediately implies the following corollary:

Corollary 1. *There is a polynomial-time algorithm that gives a 4-approximation for the p -norm fair facility location problem for any $p \in [1, \infty]$.*

2 Portfolio of solutions for fair facility location

In this section, we give present an upper bound (Theorem 2) and a lower bound (Theorem 3) on the number of solutions such that at least one of the solutions is a constant factor approximation for every norm.

Definition (Approximate solution sets). Given an instance of the fair facility location problem and an approximation factor $\alpha \geq 1$, an α -approximate solution set is a set S of solutions such that for all norms $p \in [1, \infty]$, there is some solution in S that is an α -approximation to the p -norm fair facility location problem.

2.1 Upper bound

The key idea is to use the relationship between different p -norms in \mathbb{R}^r . Before we prove the theorem, we state such a relationship that follows from Hölder's inequality [44].

Lemma 1. *For $p, q \in [1, \infty]$ with $p \leq q$ and any vector $x \in \mathbb{R}^r$, $\|x\|_q \leq \|x\|_p \leq r^{\frac{1}{p} - \frac{1}{q}} \|x\|_q$.*

We remark that the above bound is sharp; for all p, q , there exist simple examples where either of the two inequalities hold.

The next lemma states that approximate solutions for one norm can be translated into approximate solutions for a different norm, incurring an additional factor in the approximation ratio. The proof follows by a simple application of Lemma 1.

Lemma 2. *Given an instance of the fair facility location problem with r groups and $1 \leq p < q \leq \infty$, any α -approximate solution to the p -norm fair problem is an $(\alpha r^{\frac{1}{p}-\frac{1}{q}})$ -approximate solution to the q -norm fair problem, and conversely, any α -approximate solution to the q -norm fair problem is an $(\alpha r^{\frac{1}{q}-\frac{1}{p}})$ -approximate solution to the p -norm fair problem.*

Recall that OPT_p is a decreasing function of p . Define sequence q_1, \dots, q_T as follows: $q_1 = 1$, q_k is the unique norm satisfying $\frac{OPT_{q_0}}{OPT_{q_k}} = 2^{k-1}$, and $T = \lceil \log_2 \left(\frac{OPT_1}{OPT_\infty} \right) \rceil$. Also define $q_{T+1} = \infty$. From Lemma 1, we get that $OPT_\infty \leq r OPT_1$, so that $T \leq \lceil \log_2 r \rceil$. We are ready to prove the main result for this section.

Theorem 2. *Given any instance of the facility location problem, there exists an 8-approximate solution set of cardinality at most $\lceil \log_2 r \rceil$. This solution set can be obtained in polynomial time.*

Proof. Notice that

$$\frac{OPT_{q_T}}{OPT_{q_{T+1}}} = \frac{OPT_1}{2^{T-1} OPT_\infty} \leq \frac{2}{\frac{OPT_1}{OPT_\infty}} \cdot \frac{OPT_1}{OPT_\infty} = 2.$$

Corollary 1 allows us to obtain 4-approximate solutions for any p -norm for the given instance of the problem. We will claim that a 4-approximate solution $\mathcal{S}_k = (F_k, \theta_k)$ to the q_k -norm fair problem is an 8-approximate solution to the p -norm problem for any $p \in [q_k, q_{k+1}]$. Since $\cup_{k=1}^T [q_k, q_{k+1}] = [1, \infty]$, this implies that $\{\mathcal{S}_1, \dots, \mathcal{S}_T\}$ is an 8-approximate solution set for the given instance of the facility location problem. Since $|\{\mathcal{S}_1, \dots, \mathcal{S}_T\}| = T \leq \lceil \log_2 r \rceil$, this implies the theorem.

We prove the claim now. Let $V_{\theta_k} \in \mathbb{R}^r$ denote the cost vector for assignment θ_k . Then, the cost of solution \mathcal{S}_k for a norm $p \in [q_k, q_{k+1}]$ is

$$C_{F_k} + \|V_{\theta_k}\|_p \leq C_{F_k} + \|V_{\theta_k}\|_{q_k} \leq 4OPT_{q_k} \leq 8OPT_p.$$

The first inequality follows from Lemma 1, the second inequality follows since \mathcal{S}_k is a 4-approximate solution to the q_k -norm fair problem, and the final inequality follows since $OPT_{q_k} \geq OPT_p \geq OPT_{q_{k+1}} \geq \frac{1}{2}OPT_{q_k}$. This proves the claim. \square

Definition. Given an instance of the facility location problem, we will call this set of norms $\{q_1, \dots, q_T\}$ the *representative norm set*. Getting α -approximate solutions corresponding to these norms is sufficient to get 2α -approximate solutions for each norm in $[1, \infty]$.

2.2 Lower bound

We now provide a lower bound on the number of solutions required.

Theorem 3. *For any large positive integer r , there is an instance of the fair facility location problem with r clients such that at any $O(1)$ -approximate solution set contains at least $\tilde{\Omega}(\log_2 r)$ distinct solutions.*

Proof. We consider r clients located at the central vertex of a star. We let $r = t^k$, where t and k are parameters that we will choose later; also let $\epsilon = \frac{1}{k}$. The star has k edges with leaves f_1, \dots, f_k , leaf f_j has facility with opening cost t^j and is at distance $t^{(k-j)\epsilon}$ from the central vertex, for each $j \in [k]$. Each client is in its own group. This is presented in Figure 2.

For $i \in [k]$, let $p_i = \frac{k}{i}$. For appropriate choices of t, k , we will show that for distinct i, i' , no $O(1)$ -approximate solution to the p_i -norm fair problem is an $O(1)$ -approximate solution to the $p_{i'}$ -norm fair problem. Therefore, any $O(1)$ -approximate solution set for this problem instance contains at least k distinct solutions.

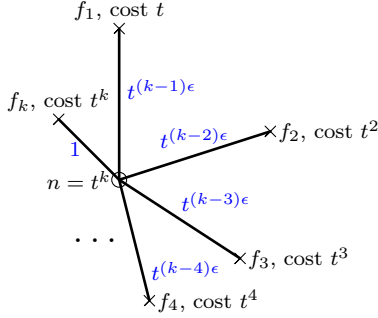


Figure 2: The star graph instance in Theorem 3. There are $r = t^k$ clients at the central vertex and no clients at any of the leaf vertices. Leaf vertex f_i has facility with cost t^i and is at a distance $t^{(k-i)\epsilon}$ from the central vertex. The distances are marked in blue.

Fix $i \in [k]$. We show that any $O(1)$ -approximate solution to the p_i -norm fair problem must have (a) the facility at f_i open and (b) none of the facilities at f_{i+1}, \dots, f_k open. This is sufficient to prove our claim.

If only f_i is open, the solution cost for p_i -norm is

$$\begin{aligned} C_{p_i, f_i} &= t^i + \|t^{(k-i)\epsilon} \mathbf{1}_r\|_{p_i} = t^i + r^{\frac{1}{p_i}} t^{(k-i)\epsilon} \\ &= t^i + t^{i+(k-i)\epsilon} = \Theta(t^{i+(k-i)\epsilon}). \end{aligned}$$

Since $\epsilon = \frac{1}{k}$, this cost is of the order $\Theta(t^{i+\frac{k-1}{k}}) = \Theta(t^{(i+1)-\frac{1}{k}})$.

For a solution \mathcal{S} with any of the facilities f_{i+1}, \dots, f_k open, the solution cost $C_{\mathcal{S}} = \Omega(t^{i+1})$, which is higher than C_{p_i, f_i} by a factor $\Omega(t^{\frac{1}{k}})$. For $k = o(\log t)$, this factor is $\omega(1)$, and therefore \mathcal{S} is not an $O(1)$ -approximate solution to the p_i -norm fair problem.

Further, consider a solution \mathcal{S} with none of the facilities f_i, \dots, f_k open. At least one of the facilities in f_1, \dots, f_{i-1} must be open, and even if all clients are assigned to the nearest open facility, the distance of each client from the nearest open facility is at least $t^{(k-(i-1))\epsilon}$. Therefore, the solution cost

$$C_{\mathcal{S}} = \Omega\left(\|t^{(k-i+1)\epsilon} \mathbf{1}_r\|_{p_i}\right) = \Omega\left(\|t^{(k-i+1)\epsilon} \mathbf{1}_r\|_{p_i}\right) = \Omega\left(t^{i+(k-i+1)\epsilon}\right).$$

This is higher than the cost of only opening f_i , i.e., C_{p_i, f_i} by a factor of $\Omega\left(t^{\frac{1}{k}}\right)$. For $k = o(\log t)$, this factor is $\omega(1)$, and therefore \mathcal{S} is not an $O(1)$ -approximate solution to the p_i -norm fair problem.

Finally, we show that for $k = \sqrt{\frac{\log_2 r}{\log_2 \log_2 r}}$ and $k^t = r$, we have $k = o(\log t)$, finishing the proof. $t^k = r$ gives $\sqrt{\frac{\log_2 r}{\log_2 \log_2 r}} \log_2 t = \log_2 r$ so that $\log_2 t = \sqrt{\log_2 r \cdot \log_2 \log_2 r}$, so that $k = o(\log_2 t)$. \square

3 Refinements

In this section, we will present our algorithms for weak and strong refinements to the facility location problem (see Section 1.1 for the definition of refinements). The key idea is to use the local relationship between approximate solutions, i.e., when two norms are relatively close, a good approximation for one is a good approximation for the other (see Lemma 2, Theorem 2). We will build on this idea to provide solution sets where the set of open facilities changes in a structured way, as required for strong and weak refinements.

We give 16-approximate weak refinements for arbitrary metrics in Theorem 4, Section 3.1.1. Strong refinements require a structural relationship between solutions for different norms not only in terms of facilities, but also in terms of assignments. We give an approximation algorithm for strong refinements for $P = [1, \infty]$ for arbitrary metrics in Section 3.1.2. We improve the approximation ratio for the special case of line metric in Section 3.2, and for tree metric in Appendix E.

From here on, we are primarily concerned by the order of our approximation ratios in terms of r . For convenience, we abuse the notation slightly and assume $\log_2 r \simeq \lceil \log_2 r \rceil$.

3.1 Arbitrary metric

We give approximation algorithms for weak and strong refinements for arbitrary metrics in this section. The main results in this section are Theorems 4, 5, and Corollary 2. Theorem 4 gives $O(1)$ -approximate weak refinement for all norms, i.e., the norm set $P = [1, \infty]$. Theorem 5 gives an $O(e^{3\sqrt{l}})$ -approximate strong refinement for any finite norm set $P = \{p_1, \dots, p_l\}$ and Corollary 2 gives a $\text{poly}(r^{\frac{1}{\sqrt{\log_2 r}}})$ -approximate strong refinement for the norm set $P = [1, \infty]$. Note that $e^{3\sqrt{l}}$ is *subexponential* in l and $r^{\frac{1}{\sqrt{\log_2 r}}}$ is *sublinear* in r : in fact, $r^{\frac{1}{\sqrt{\log_2 r}}} = 2^{\sqrt{\log_2 r}} = O(r^\delta)$ for all $\delta > 0$. These peculiar functions arise as bounds on the terms of a recurrence relation that occurs naturally in our problem (see Lemmas 4, 5).

3.1.1 Weak refinement

The key idea to get weak refinements is very simple: given norms $p < q$, we can augment an approximate solution $\mathcal{S}_p = (F_p, \theta_p)$ for the p -norm problem by including facilities F_q , where $\mathcal{S}_q = (F_q, \theta_q)$ is an approximate solution for the q -norm problem. Since $\text{OPT}_p \geq \text{OPT}_q$, the cost of the augmented solution $\mathcal{R}_p = (F_p \cup F_q, \theta_p)$ is at most twice the cost of \mathcal{S}_p , and the solution set $\{\mathcal{R}_p, \mathcal{S}_q\}$ forms a weak refinement for $P = \{p, q\}$. This idea can be extended to larger norm sets. The *representative norm set* defined in Section 2 allows us to pick a norm set of cardinality $\log_2 r$ to get a facility refinement for $P = [1, \infty]$.

Theorem 4. *Given any instance of the facility location problem, there exists a 16-approximate weak refinement for the norm set $[1, \infty]$. These weak refinements can be obtained in polynomial-time.*

Proof. Recall the representative norm set $P = \{q_1, \dots, q_T\}$ defined for Theorem 2: $q_1 = 1$, $T = \lceil \log_2 \frac{OPT_1}{OPT_\infty} \rceil \leq \lceil \log_2 r \rceil$ and $OPT_{q_k} = \frac{OPT_{q_1}}{2^{k-1}}$ for all $k \in [l]$. Additionally, $q_{T+1} = \infty$.

We showed that a 4-approximate solution for the q_k -norm fair problem is an 8-approximate solution for the p -norm fair problem for all $p \in [q_k, q_{k+1}]$. Therefore, an α -approximate facility refinement for $P = \{q_1, \dots, q_T\}$ is a 2α -approximate facility refinement for $[1, \infty]$. We will obtain an 8-approximate facility refinement for P , implying the theorem.

For all $k \in [T]$, let $\mathcal{S}_k = (F_k, \theta_k)$ be a 4-approximate solution to the q_k -norm fair problem, obtained using Corollary 1. Define G_k as the union of all facilities from F_k through F_T , i.e., $G_k = \bigcup_{s \in [k, T]} F_s$. Then $G_1 \supseteq G_2 \supseteq \dots \supseteq G_T$. Further, define solution $\mathcal{R}_k = (G_k, \theta_k)$. Then,

$$\begin{aligned} C_{\mathcal{R}_k} &= C_{G_k} + C_{\theta_k} \leq \left(\sum_{s \in [k, T]} C_{F_s} \right) + C_{\theta_k} = \left(\sum_{s \in [k+1, T]} C_{F_s} \right) + (C_{F_k} + C_{\theta_k}) \\ &\leq \left(\sum_{s \in [k+1, T]} 4 OPT_{q_s} \right) + 4 OPT_{q_k} = \left(\sum_{s \in [k+1, T]} 4 \frac{OPT_{q_k}}{2^{s-k}} \right) + 4 OPT_{q_k} \\ &= 4 OPT_{q_k} \left(\sum_{s \in [k, T]} \frac{1}{2^{s-k}} \right) \leq 4 OPT_{q_k} \left(\sum_{s \geq k} \frac{1}{2^{s-k}} \right) = 8 OPT_{q_k}. \end{aligned}$$

Therefore, \mathcal{R}_k is an 8-approximate solution for the q_k -norm fair problem, implying that the set $\{\mathcal{R}_1, \dots, \mathcal{R}_T\}$ is an 8-approximate facility refinement for P , and consequently a 16-approximate facility refinement for $[1, \infty]$. \square

3.1.2 Strong refinement

We proceed to discuss strong refinement. Given a finite $P = \{p_1, \dots, p_l\}$, we give a polynomial-time algorithm that gives an $O(e^{3\sqrt{l}})$ -approximate strong refinement for P . The first step is to get a weak refinement, as in Theorem 4. A weak refinement may not be a strong refinement, and so a natural question is if we can we reassign the clients to facilities in each solution (i.e., for each $k \in [l]$) so that we get a strong refinement. Such a reassignment may increase the assignment cost for each $k \in [l]$. Notice that we do not open any facilities in any such reassignment. The key idea in this section is to give a reassignment such that the distance of each client from its assigned facility increases by at most a factor β , and this enables us to convert the bound of β into an approximate factor for strong refinement. We will show that $\beta = O(e^{3\sqrt{l}})$ for our algorithm. The main results for arbitrary metrics are given in Theorem 5 and Corollary 2, which we will improve upon this for the line metric in Section 3.2. Before we present the main algorithm in this section, we develop intuition using the greedy algorithm, and show why it fails.

Greedy algorithm and why it fails. From Theorem 4, we can obtain an $O(1)$ -weak refinement $\{(G_k, \Pi_k) : k \in [l]\}$ with $G_1 \supseteq \dots \supseteq G_l$. We denote by G_0 the set of clients and the set of facilities

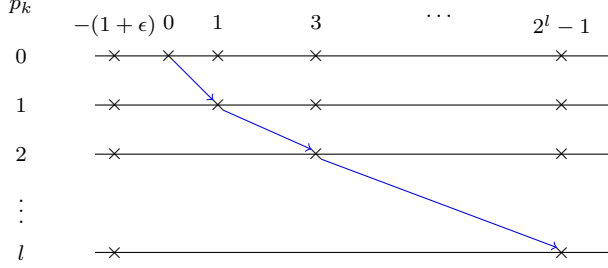


Figure 3: A toy example to illustrate that the greedy algorithm for strong refinements can reassign clients to far away facilities. The underlying metric is a line metric, and elements of sets G_0, \dots, G_l are denoted as a cross at their respective locations, with a copy of the line for each $k \in [l]$ for clarity. For $\epsilon > 0$, the closest facility to $x = 0$ in G_1 is at $x = 1$, and so the greedy algorithm assigns $\Pi_1(0) = 1$. Further, the closest facility to $x = 1$ in G_2 is at $x = 3$, so that $\Pi_2(0) = 3$. Continuing this way, $\Pi_l(0) = 2^l - 1$. However, the closest facility to $x = 0$ in G_l is at $-(1 + \epsilon)$. For $\epsilon \simeq 0$, this reassignment gives a factor $\beta \simeq 2^l - 1$ for the client at $x = 0$.

open for the 1-norm solution, i.e., $G_0 = D \cup G_1$ for convenience, so that $G_0 \supseteq G_1 \supseteq \dots \supseteq G_l$. Let us first consider a natural greedy algorithm: for each $j \in G_0$, assign $\Pi_1(j) = \arg \min_{f \in G_1} d(j, f)$, that is, the facility in G_1 closest to j . Call this facility f_1 . Under Π_2 , assign j to the facility in G_2 closest to f_1 in G_2 , i.e., $\Pi_2(j) = \arg \min_{f \in G_2} d(f_1, f)$, and so on. Recursively, for client j , assign $\Pi_{k+1}(j) = \arg \min_{f \in G_{k+1}} d(\Pi_k(j), f)$ for all $k \geq 0$. It is not difficult to see that this algorithm produces a strong refinement, albeit with a higher approximation factor. Note that for higher k , this can lead to some clients being reassigned to facilities very far away; see Figure 3 for an example where the increase in the approximation factor can be very large for a client. Therefore, the greedy algorithm fails to provide a factor better[§] than 2^l .

Our approach through DISCOUNTEDLOOKAHEAD. One can think of the greedy algorithm as looking ‘one step’ ahead: for each $k \in [0, l - 1]$ and each $f \in G_k$, map $\sigma(f) = \arg \min_{h \in G_{k+1}} d(f, h)$. Then, for each client $j \in G_0$, the greedy algorithm assigns $\Pi_k(j) = \sigma^k(j)$. A natural question we consider here is if we can improve this algorithm by looking *more than* one step ahead at a time? Specifically, for client $j \in G_0$, instead of looking only at facility $f_1 = \arg \min_{f \in G_1} d(j, f)$, we can also look at facilities $h_k = \arg \min_{h \in G_k} d(j, h)$ for $k \in [l]$ (that is, h_k is the facility closest to j in G_k , so that $f_1 = h_1$). We pick some ‘discounting factor’ $\gamma > 1$ to be chosen later, and instead of assigning $\Pi_1(j) = h_1$, we assign $\Pi_{s^*}(j) = h_{s^*}$ where

$$s_* = \arg \min_{k \in [l]} \frac{d(j, h_k)}{\gamma^k}.$$

Since $G_k \supseteq G_{s^*}$ for all $k \leq s^*$, we can assign $\Pi_k(j) = h_{s^*}$, and for $k > s^*$, we can do this process recursively for $G_{s^*}, G_{s^*+1}, \dots, G_l$. Algorithm 1 does precisely this, and we show that this leads to a *subexponential* approximation factor in l .

For each $k \in [0, l - 1]$ and each $t \in [k + 1, l]$, the algorithm constructs an assignment $\Pi_t^{(k)} : G_k \rightarrow G_t$ from facilities at a lower level k to facilities at a higher level t . The final assignments on clients $G_0 = D$ are $\Pi_k^{(0)} : G_0 \rightarrow G_k$ for $k \in [l]$.

Lemma 3. *The assignments $\Pi_1^{(0)}, \dots, \Pi_l^{(0)}$ output by Algorithm 1 form a strong refinement.*

[§]We can prove that no client can be assigned *too* far away under the greedy algorithm. We can prove the bound $\beta \leq 2^l$, so that the greedy algorithm gives an $O(2^l)$ strong refinement.

Algorithm 1 DISCOUNTEDLOOKAHEAD

- 1: Obtain solutions $\{\mathcal{S}_k = (F_k, \Pi_k) : k \in [l]\}$ such that \mathcal{S}_k is a 4-approximation to the p_k -norm fair facility location problem.
- 2: Denote $G_k = \cup_{t \in [k, l]} F_t$ for $k \in [l]$
- 3: For any $k, t \in [0, l]$ and any $f \in G_k$, denote $h_t(f) = \arg \min_{h \in G_t} d(f, h)$
- 4: **for** $k = l - 1$ **to** 0 **do**
- 5: **for** each $f \in G_k$ **do**

$$s^* = \arg \min_{t \in [k+1, l]} \frac{d(f, h_t(f))}{\gamma^t},$$

where s^* is the least such number in case of ties. Let h be the corresponding facility, i.e., $h = h_{s^*}(f)$.

- 6: Set

$$\Pi_t^{(k)}(f) = \begin{cases} h & \text{if } t \in [k+1, s^*]; \\ \Pi_t^{(s^*)}(h) & \text{if } t \in [s^*+1, l] \end{cases}$$

- 7: Output solutions $(G_1, \Pi_1^{(0)}), \dots, (G_l, \Pi_l^{(0)})$
-

Proof. First, we use induction on $l - k$ to prove the following: if $f \in G_t$, then $\Pi_s^{(k)}(f) = f$ for all $k < s \leq t$. When $k = l - 1$, this is trivially true since $d(f, f) = 0$. When $k < l - 1$, $s^* = s + 1$ and $h_{k+1}(f) = f$ in step 5 since $d(f, f) = 0$ and $f \in G_{k+1}$. By induction, $\Pi_t^{(k+1)}(f) = f$, and since by step 6, we have $\Pi_t^{(k)}(f) = \Pi_t^{(k+1)}(h_{k+1}(f))$, we get $\Pi_t^{(k)}(f) = f$.

For convenience, define $\Pi_k^{(k)} : G_k \rightarrow G_k$, where $\Pi_k^{(k)}(f) = f$ for all $f \in G_k$. We prove the following claim: for all $0 \leq k < s \leq t \leq l$ and for all (open) facilities $f \in G_k$ for $k \geq 1$ (and clients in G_0 for $k = 0$),

$$\Pi_t^{(k)}(f) = \Pi_t^{(s)}\left(\Pi_s^{(k)}(f)\right).$$

We prove this using induction on $l - k$. For $k = l - 1$, $s = t = l$, and by definition $\Pi_l^{(l)}\left(\Pi_l^{(k)}(f)\right) = \Pi_l^{(k)}(f)$.

Suppose $l - k > 1$, i.e., $k < l - 1$. As in Algorithm 1, let $s^* = \arg \min_{s' \in [k+1, l]} \gamma^{-s'} d(f, h_{s'}(f))$ and denote $h = h_{s^*}(f)$ for brevity in the proof that follows.

Case I: $s \leq t \leq s^*$. In this case, by step 6, $\Pi_t^{(k)}(f) = \Pi_s^{(k)}(f) = h$. Since $h \in G_t$, by our earlier claim, $\Pi_t^{(s)}(h) = h$, proving the desired equality.

Case II: $s < s^* \leq t$. Since $h \in G_{s^*}$, by our earlier claim we have $\Pi_{s^*}^{(s)}(h) = h$. Since $l - s^* < l - k$, using the induction hypothesis

$$\Pi_t^{(s)}(h) = \Pi_t^{(s^*)}\left(\Pi_{s^*}^{(s)}(h)\right) = \Pi_t^{(s^*)}(h).$$

By step 6, $\Pi_s^{(k)}(f) = h$ while $\Pi_t^{(k)}(f) = \Pi_t^{(s^*)}(h)$, so that

$$\Pi_t^{(s)}\left(\Pi_s^{(k)}(f)\right) = \Pi_t^{(s)}(h) = \Pi_t^{(s^*)}(h) = \Pi_t^{(k)}(f).$$

Case III: $s^* \leq s \leq t$. Since $l - s < l - k$, by the induction hypothesis,

$$\Pi_t^{(s^*)}(h) = \Pi_t^{(s)}\left(\Pi_s^{(s^*)}(f)\right). \quad (1)$$

In this case, $\Pi_t^{(k)}(f) = \Pi_t^{(s^*)}(h)$ and $\Pi_s^{(k)}(f) = \Pi_s^{(s^*)}(h)$, thus proving the desired equality and consequently the claim.

Facility sets $G_1 \supseteq \dots \supseteq G_l$ already form a weak refinement. To prove that assignments $\Pi_1^{(0)}, \dots, \Pi_l^{(0)}$, we need to show that for all $k \in [1, l-1]$ and for all $f \in G_k$, there exists some $f' \in G_{k+1}$ such that each client assigned to f under $\Pi_k^{(0)}$ is assigned to f' under $\Pi_{k+1}^{(0)}$, that is:

$$\left\{j \in G_0 : \Pi_k^{(0)}(j) = f\right\} \subseteq \left\{j \in G_0 : \Pi_{k+1}^{(0)}(j) = f'\right\}.$$

We claim that $f' = \Pi_{k+1}^{(k)}(f)$ works.

Suppose $j \in G_0$ is such that $\Pi_k^{(0)}(j) = f$. Then j must have been assigned to f under $\Pi_k^{(0)}$ in one of the two cases in step 6. Let s^* be the corresponding index chosen when j is assigned.

If $k < s^*$, then $f = \Pi_k(j) = \Pi_{k+1}(j) = h_{s^*}(j)$. Since $h_{s^*}(j) \in G_{k+1}$, $\Pi_{k+1}^{(k)}(h_{s^*}(j)) = h_{s^*}(j)$.

If $k \geq s^*$, then $f = \Pi_k^{(0)}(j) = \Pi_k^{(s^*)}(h_{s^*}(j))$ and $\Pi_{k+1}^{(0)}(j) = \Pi_{k+1}^{(s^*)}(h_{s^*}(j))$. By eqn. (1),

$$\Pi_{k+1}^{(0)}(j) = \Pi_{k+1}^{(s^*)}(h_{s^*}(j)) = \Pi_{k+1}^{(k)}\left(\Pi_k^{(s^*)}(h_{s^*}(j))\right) = \Pi_{k+1}^{(k)}(f).$$

□

Before we prove our main theorem for this section, we need the following technical lemmas; we include their proofs in Appendix C.

Lemma 4. *Given $\gamma > 1$ and some $l \in \mathbb{Z}_+$, consider the set of recursively defined numbers $\{u_{k,t} : t \in [l], k \in [0, t-1]\}$: for each $t \in [l]$, and for $k = t-1, \dots, 0$ define*

$$u_{k,t} = \begin{cases} \gamma^{l-t} & \text{if } k = t-1, \\ \max\{\gamma^{l-t}, \max_{k < s < t} \gamma^{s-t} + u_{s,t}(1 + \gamma^{s-t})\} & \text{if } k \in [0, t-2]. \end{cases} \quad (2)$$

Then $\max_{k,t} u_{k,t} \leq e^{\frac{2\gamma}{\gamma-1}} \gamma^{l-1}$.

Lemma 5. $\min_{\gamma > 1} e^{\frac{2\gamma}{\gamma-1}} \gamma^{l-1} \leq e^2 \cdot e^{3\sqrt{l}}.$

We are ready to prove the approximation guarantee for Algorithm 1:

Theorem 5. *Given a finite $P = \{p_1, \dots, p_l\}$ of norms, there is a polynomial-time algorithm that gives an $O(e^{3\sqrt{l}})$ -approximate strong refinement for P .*

Proof. Lemma 3 shows that Algorithm 1 gives a strong refinement for P . It remains to determine the choice of the parameter γ and prove the desired bound on the approximation ratio.

Assume without loss of generality that $p_1 < \dots < p_l$; we know that $OPT_{p_1} \geq \dots \geq OPT_{p_l}$. In step 2 of the algorithm,

$$C_{G_k} \leq \sum_{t \in [k, l]} C_{F_t} \leq 4 \sum_{t \in [k, l]} OPT_{p_t} \leq 4 \sum_{t \in [k, l]} OPT_{p_k} \leq 4l OPT_{p_k}.$$

If $\theta_k : D \rightarrow G_k$ assigns every client to its closest vertex in G_k , we have $C_{\theta_k} \leq C_{\Pi_k}$ since $G_k \supseteq F_k$. We will show that assignments $\Pi_k^{(0)}$ output by the algorithm satisfy $C_{\Pi_k^{(0)}} \leq O(e^{3\sqrt{l}})C_{\theta_k}$, and therefore that

$$C_{(G_k, \Pi_k^{(0)})} = C_{G_k} + C_{\Pi_k^{(0)}} \leq 4l OPT_{p_k} + O(e^{3\sqrt{l}})C_{\Pi_k} \leq \left(l + O(e^{3\sqrt{l}})\right) OPT_{p_k} = O(e^{3\sqrt{l}}) OPT_{p_k}.$$

To this end, it is sufficient to show that the assignments $\Pi_1^{(0)}, \dots, \Pi_1^{(l)}$ satisfy

$$d\left(j, \Pi_k^{(0)}(j)\right) \leq O(e^{3\sqrt{l}}) d(j, h_k(j)), \quad \forall j \in D. \quad (3)$$

Instead of proving (3), we prove the following stronger inductive statement:

$$d\left(f, \Pi_t^{(k)}(f)\right) \leq u_{k,t} d(f, h_t(f)), \quad \forall f \in G_k, \forall k \in [0, l-1], t \in [k+1, l]. \quad (4)$$

where u is the sequence defined in Lemma 4. Lemmas 4, 5 then imply (3).

We induct on $l - k$. Base case: $l - k = 1$. Consider any $f \in G_k$, and let s^* be the index in $[k+1, l]$ chosen in step 5. As in the algorithm, we denote $h = h_{s^*}(f)$. Then $s^* = l$, $h = h_l(f)$, and $\Pi_l^{(k)}(f) = h_l(f)$, so that (4) is trivially true.

Suppose $l - k > 1$. Consider any $f \in G_k$. We have two cases.

Case I: $t \leq s^*$. By definition, $\Pi_t^{(k)}(f) = h_{s^*}(f) := h$. By the choice of s^* , $\gamma^{-t}d(h_t(f), f) \geq \gamma^{-s^*}d(h, f)$, so that $d(h, f) \leq \gamma^{s^*-t}d(h_t(f), f) \leq \gamma^{l-t}d(h_t(f), f) \leq u_{k,t}d(h_t(f), f)$.

Case II: $t > s^*$. Since $s^* > k$, $l - s^* < l - k$, and so we have by the induction hypothesis that

$$d\left(h, \Pi_t^{(s^*)}(h)\right) \leq u_{s^*,t} d(h, h_t(h)) \leq u_{s^*,t} d(h, h_t(f)),$$

where the last inequality follows since by definition $h_t(h)$ is the closest facility to h in G_t . Therefore, we have

$$\begin{aligned} d\left(f, \Pi_t^{(k)}(f)\right) &\leq d(f, h) + d\left(h, \Pi_t^{(k)}(f)\right) = d(f, h) + d\left(h, \Pi_t^{(s^*)}(h)\right) \\ &\leq d(f, h) + u_{s^*,t} \cdot d(h, h_t(f)) \leq d(f, h) + u_{s^*,t} \cdot (d(f, h) + d(f, h_t(f))). \end{aligned}$$

The first and final inequalities are triangle inequalities. The equality is by definition of $\Pi_t^{(k)}(f)$ and s^* .

Now, since $s^* = \arg \min_{t \in [k+1, l]} d(f, h_t(f))$, we have $d(f, h) \leq \gamma^{s^*-t} d(f, h_t(f))$, so that the above inequality becomes

$$d\left(f, \Pi_t^{(k)}(f)\right) \leq d(f, h_t(f)) \times \left(\gamma^{s^*-t} + u_{s^*,t} \cdot (\gamma^{s^*-t} + 1)\right).$$

Since

$$u_{k,t} \geq \max_{k < s < t} \left(\gamma^{s-t} + u_{s,t} \cdot (\gamma^{s-t} + 1)\right) \geq \gamma^{s^*-t} + u_{s^*,t} \cdot (\gamma^{s^*-t} + 1),$$

we get that $d\left(f, \Pi_t^{(k)}(f)\right) \leq d(f, h_t(f)) \times u_{k,t}$. \square

Using the representative norm set, we can choose a finite norm set of size at most $\log_2 r$ to represent the norm set $P = [1, \infty]$ to get the following corollary:

Corollary 2. *There is a $\text{poly}(r^{\frac{1}{\sqrt{\log_2 r}}})$ -approximate strong refinement for $P = [1, \infty]$ for arbitrary metrics.*

3.2 Line metric

Consider the special case when the underlying metric $\mathcal{M} = (X, d)$ is a line metric, i.e., $X = \mathbb{R}$. We will denote by $x(f) \in \mathbb{R}$ the location of facility $f \in F$. In this section, we give an algorithm for an $O(\log_2 r)$ -approximate strong refinement for $P = [1, \infty]$. We prove the following theorem for a finite norm set, and obtain the result for $P = [1, \infty]$ as a corollary using the *representative norm set* defined in Section 2.

Theorem 6. *Given a finite set of norms P , there is a polynomial-time algorithm that gives an $O(|P|)$ -approximate strong refinement for P for the line metric.*

Corollary 3. *There is a polynomial-time algorithm that gives an $O(\log_2 r)$ -approximate strong refinement for $P = [1, \infty]$ for the line metric.*

We now develop the algorithm for $P = \{p_1, \dots, p_l\}$. Assume without loss of generality that $p_1 < \dots < p_l$, so that $OPT_{p_1} \geq \dots \geq OPT_{p_l}$. Obtain a 4-approximate solution $\mathcal{S}_k = (F_k, \theta_k)$ for each $k \in [l]$ using Corollary 1. A weak refinement can be obtained in a manner that we have deployed before: by combining facilities at all higher norms for each p_k . That is, define $G_k = \bigcup_{t \in [k, l]} F_t$. We can bound the facility costs G_k for each k :

$$C_{G_k} \leq \left(\sum_{t \geq k} C_{F_t}\right) \leq \left(\sum_{t \geq k} OPT_{p_t}\right) \leq 4(l - (k - 1)) OPT_{p_k} \leq 4l OPT_{p_k}. \quad (5)$$

However, assignments $\theta_k, k \in [l]$ may not form a partition refinement. Suppose we construct assignments $\Pi_k : D \rightarrow G_k, k \in [l]$ so that

1. (Strong refinement condition) the solution set $T = \{(G_k, \Pi_k) : k \in [l]\}$ forms a strong refinement for $\{p_1, \dots, p_l\}$, and

2. (Cost upper bound) assignments Π_k assign each client to a facility within a factor $1/\alpha$ of its nearest facility in G_k for some fixed $\alpha \in (0, 1]$. That is, for all $k \in [l]$,

$$d(j, \Pi_k(j)) \leq \frac{1}{\alpha} \min_{f \in G_k} d(j, f) \quad \forall j \in D. \quad (6)$$

From the second condition, we have that $d(j, \Pi_k(j)) \leq \frac{1}{\alpha} \min_{f \in G_k} d(j, f) \leq \frac{1}{\alpha} d(j, \theta_k(j))$ for all clients $j \in D$. From eqn. (5), we further have that $C_{G_k} \leq 4l \text{OPT}_{p_k}$. Therefore, since solution $\mathcal{S}_k = (F_k, \theta_k)$ is a 4-approximate solution to the p_k -norm problem for each k , we can bound the cost of solution (G_k, Π_k) for norm p_k :

$$C_{(G_k, \Pi_k)} = C_{G_k} + C_{\Pi_k} \leq 4l \text{OPT}_{p_k} + \frac{1}{\alpha} C_{\theta_k} \leq 4l \text{OPT}_{p_k} + \frac{4}{\alpha} \text{OPT}_{p_k} = \left(4l + \frac{4}{\alpha}\right) \text{OPT}_{p_k},$$

implying that if assignments $\Pi_k, k \in [l]$ satisfy the two conditions above and if $1/\alpha = O(l)$, then the solution set $T = \{(G_k, \Pi_k) : k \in [l]\}$ forms an $O(l)$ -approximate strong refinement for $\{p_1, \dots, p_l\}$.

Our algorithm achieves precisely these two conditions with $\alpha = \frac{1}{2l}$, implying Theorem 6. We first introduce some notation for this section. Then, we establish some structure on strong refinements on a line and give an outline of the algorithm.

Notation: We will refer to $k \in \{1, \dots, l\}$ as problem *levels*. Recall that $[l]$ indexes the set of norms we will simultaneously approximate for. For example, $k = 1$ may refer to the L_1 -norm and $k = l$ approximates higher norms such as L_∞ , so that as the levels increase from $k = 1$ to l , a smaller number of facilities are opened. For technical reasons, we introduce an auxiliary level $(l + 1)$, with $G_{l+1} = \{f_0\}$, f_0 being an arbitrary facility in G_l and all clients being mapped to f_0 under any assignment for level $(l + 1)$.

Since a facility may be present in multiple levels G_k , it will be convenient to distinguish between these copies. We define \overline{G}_k to be facility-level pairs in G_k , i.e., $\overline{G}_k = G_k \times \{k\} = \{(f, k) : f \in G_k\}$ for all $k \in [l + 1]$. We will also denote $\overline{G} = \overline{G}_1 \cup \dots \cup \overline{G}_{l+1}$ to be the union of these sets; notice that this union is disjoint, even though $G_1 \supseteq \dots \supseteq G_l \supseteq G_{l+1}$.

Since clients can be identified by their locations on \mathbb{R} , we will think of any assignment of clients to facilities in G_k as instead a function from \mathbb{R} to G_k . Further, for such an assignment $\Pi_k : \mathbb{R} \rightarrow G_k$ and for any facility $f \in G_k$, we denote $A(f, k)$ to be the set of points in \mathbb{R} assigned to f at level k using assignment Π_k , i.e., $A(f, k) = \{j \in \mathbb{R} : \Pi_k(j) = f\}$.

If f, f' are consecutive facilities in G_k with $x(f) < x(f')$, we define $b_\alpha(f, k) = \alpha x(f') + (1 - \alpha)x(f)$, that is, the point dividing interval $[x(f), x(f')]$ in ratio $\alpha/(1 - \alpha)$. If f is the rightmost facility in G (i.e., if f' does not exist), we define $b_\alpha(f, k) = +\infty$. Similarly, we define $a_\alpha(f', k) = (1 - \alpha)x(f') + \alpha x(f)$, and if f' is the leftmost facility in G_k , then we define $a_\alpha(f', k) = -\infty$. We define intervals $I_\alpha(f, k) = [a_\alpha(f, k), b_\alpha(f, k)]$ for all $f \in G_k$. We omit the subscript α when it is clear from context. Figure 4 gives an example to provide geometric intuition for these definitions.

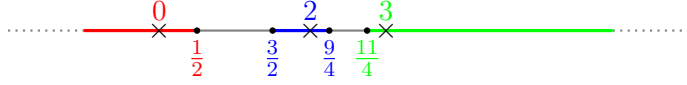


Figure 4: An example illustrating intervals $I(f, k)$ for $\alpha = 1/4$ and a fixed k . G_k consists of three facilities, located at 0, 2, 3 respectively. $I(0, k) = [a(0, k), b(0, k)] = [-\infty, 1/2]$, $I(2, k) = [3/2, 9/4]$, and $I(3, k) = [11/4, +\infty]$.

Satisfying the strong refinement condition: Suppose the sets $A(f, k)$ are intervals of \mathbb{R} for all $(f, k) \in \overline{G}$. If the assignments Π_1, \dots, Π_{l+1} form a strong refinement, then for each $k \in [l]$ and $f \in G_k$, there exists some $h \in G_{k+1}$ such that $A(f, k) \subseteq A(h, k+1)$ [¶].

Definition (Hierarchy tree). Form the directed graph $H = (\overline{G}, E)$ with the edge set $E = \{((h, k+1), (f, k)) : f \in G_k, h \in G_{k+1}, A(f, k) \subseteq A(h, k+1)\}$. When Π_1, \dots, Π_{l+1} form a strong refinement, then H is tree rooted at $(f_0, l+1)$, and we call it a *hierarchy tree*.

The hierarchy tree satisfies the following properties:

1. (Interval tree) (a) If (h, k') is the parent of (f, k) , then $A(h, k') \supseteq A(f, k)$, and (b) if (f, k) and (f', k') are siblings, then $A(f, k) \cap A(f', k') = \emptyset$.
2. (Immediate parent) The parent of (f, k) is always $(h, k+1)$ for some $h \in G_{k+1}$. That is, there are $l+1$ levels in this tree, corresponding to the sets $\overline{G}_{l+1}, \dots, \overline{G}_1$ ordered from the top to the bottom of the tree.
3. (Completeness) If the children of $(h, k+1)$ are $(f_1, k), \dots, (f_t, k)$, then the intervals $A(f_1, k), \dots, A(f_t, k)$ partition $A(h, k+1)$. Under the above two conditions, this means that no point in \mathbb{R} is unassigned at any level.

Conversely, if there exists a rooted tree H on vertex set \overline{G} for Π_1, \dots, Π_{l+1} that satisfies these conditions, then it is not difficult to see that these assignments form a strong refinement. We design our algorithm so that the assignments it outputs satisfy these conditions.

Satisfying the cost upper bound: Consider some assignment $\Pi_k : \mathbb{R} \rightarrow G_k$ at level k . Let f, f' be two consecutive facilities in G_k with $x(f) < x(f')$. Consider the interval $[x(f), x(f')]$. For some $\alpha \leq 1/2$; suppose Π_k assigns each point in $[x(f), b_\alpha(f, k)] = I_\alpha(f, k) \cap [x(f), x(f')]$ to f and each point in $[a_\alpha(f', k), x(f')] = I_\alpha(f', k) \cap [x(f), x(f')]$ to f' . Then, irrespective of whether Π_k assigns points in $(b_\alpha(f, k), a_\alpha(f', k))$ to f or f' , we get that for all points $j \in [x(f), x(f')]$:

$$d(j, \Pi_k(j)) \leq \frac{1-\alpha}{\alpha} \min_{h \in G_k} d(j, h).$$

If this is done for all pairs of facilities in G_k and for all $k \in [l]$, then we satisfy the cost upper bound (6), yielding a $(4l + 4/\alpha)$ -approximation.

[¶]Recall that $G_{l+1} = \{f_0\}$, implying $A(f_0, l+1) = \mathbb{R}$, so this condition is trivially true when $k = l$, by choosing $h = f_0$.

This is equivalent to the following: the interval assigned to facility f at level k , $A(f, k)$ must contain $I_\alpha(f, k)$, that is,

$$A(f, k) \supseteq I_\alpha(f, k) \quad \forall (f, k) \in \overline{G}. \quad (7)$$

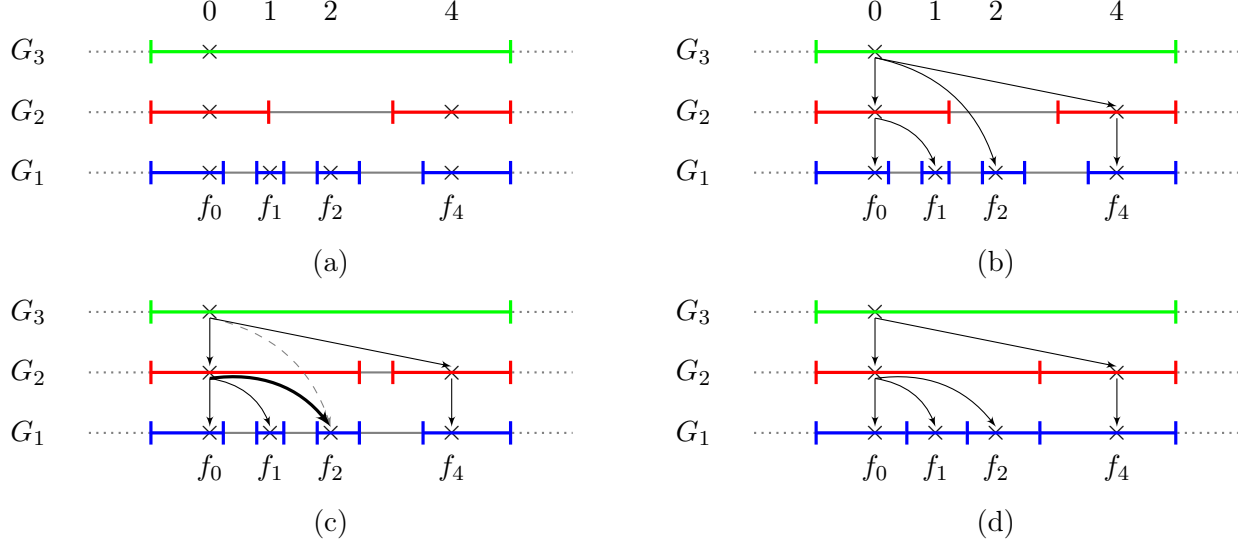


Figure 5: An example to illustrate Algorithm 2. Facilities are drawn as crosses, and the facilities f_0, f_1, f_2, f_4 are at $x = 0, 1, 2, 4$ respectively. $l = 2$, and so α is set to $1/4$. Intervals A for facilities in G_1, G_2 are drawn in blue, red respectively. The auxiliary facility in G_3 is f_0 and $A(f_0, 3)$ is in green. The edges of the hierarchy tree H is drawn in black. (a) The intervals $A(f, k)$ are initialized to $I_\alpha(f, k)$ for all $(f, k) \in \overline{G}$. (b) Step 1: at level $k = 1$, none of the sets $A(f_i, 1), i \in \{0, 1, 2, 4\}$ is updated. At level $k = 2$, since $A_\alpha(f_0, 2)$ intersects $A(f_1, 1) = I_\alpha(f_1, 1)$, $A(f_0, 2)$ is updated to $A(f_0, 2) \cup A(f_1, 1)$. The edges in H are also added. (c) Step 2: the only problematic edge that prevents H from satisfying the immediate parent condition 2 is $((f_0, 3), (f_2, 0))$. To fix this, look at the child of $(f_0, 3)$ at level 2, i.e., $(f_0, 2)$ and $(f_4, 2)$, and select the one closest to $(f_2, 1)$. Since both of these are at the same distance from $(f_2, 1)$, the tie is broken arbitrarily $(f_0, 2)$ is selected. The edge $((f_0, 3), (f_2, 1))$ is removed and add edge $((f_0, 2), (f_2, 1))$ is added to H . $A(f_0, 2)$ is expanded to $\text{conv}(A(f_0, 2) \cup A(f_2, 1))$. (d) Step 3: Intervals A are expanded recursively to cover all points at all levels.

Outline of the algorithm: We choose $\alpha = \frac{1}{2l}$ with some foresight. Our algorithm (Algorithm 2, EXPANDINTERVALS) starts by assigning $A(f, k) = I_\alpha(f, k)$ for all $(f, k) \in \overline{G}$. It maintains the invariant that $A(f, k) \supseteq I_\alpha(f, k)$ at all times, thus satisfying (7) (and therefore also satisfying the cost upper bound (6)) when the algorithm ends (we record this in Lemma 7). This is done by making sure that each update to $A(f, k)$ only makes it larger, never smaller. There are three major steps in the algorithm:

1. Step 1 modifies intervals $A(f, k)$ and forms a hierarchy tree H that is an interval tree, i.e., satisfying the interval tree condition (1). This is done by expanding intervals $A(f, k)$ iteratively from lower levels to higher levels $k = 1, \dots, l + 1$ and adding edges to the hierarchy tree H so that interval tree property (1a) is satisfied. Our choice of α ensures that these intervals are not too large, so that property (1b) is also satisfied. We prove this in Lemma 9.
2. Step 2 rearranges H so that it satisfies the immediate parent condition (2). The problematic edges are of the form $((f, k), (h, k'))$ with $k - k' > 2$. The algorithm first finds an appropriate

Algorithm 2 EXPANDINTERVALS(G_1, \dots, G_l) for strong refinement over line metric

input: Sets of facilities $\{G_k : l \in [l+1]\}$ that are open at level k , s.t. $G_{k-1} \supseteq G_k$ for all k and $|G_{l+1}| = 1$, with corresponding (facility, level) pairs \overline{G}_k

output: Strong refinement $\Pi_k : D \rightarrow G_k, k \in [l+1]$ such that the level- k facility that serves any client j is at most α factor away from the closest facility on level k : $d(j, \Pi_k(j)) \leq \frac{1}{\alpha} \min_{i \in G_k} d(i, j)$ for all clients $j \in \mathbb{R}$

- 1: Set $\alpha = 1/2l$, initialize the partial mapping $\sigma : \overline{G} \rightarrow \overline{G}$ to be empty
- 2: Represent the current mapping σ as a directed graph $H = (V, E_\sigma = \emptyset)$ with $V = \overline{G} = \bigcup_{k \in [l+1]} \overline{G}_k$.
- 3: Calculate sets $I(f, k)$ and initialize $A(f, k) = I(f, k)$ for all $(f, k) \in \overline{G}$
invariant: $A(f, k) \supseteq I(f, k)$ for all $(f, k) \in \overline{G}$ (Lemma 7)
step 1: Populate σ so that H is an interval tree (Condition 1)
- 4: **for** $k = 2$ to $l+1$ **do**:
- 5: **for** each facility $f \in G_k$ **do**
- 6: Define the set of facilities $S(f, k) = \{(h, k') : k' < k, \sigma(h, k') \text{ is unassigned}, A(h, k') \cap A(f, k) \neq \emptyset\}$ and update the interval

$$A(f, k) \leftarrow A(f, k) \cup \left(\bigcup_{(h, k') \in S(f, k)} A(h, k') \right)$$

- 7: Update the mapping $\sigma(h, k') = (f, k)$ for all facilities $(h, k') \in S(f, k)$
- 8: Add edge $((f, k), (h, k'))$ to H

Lemma 9: H is an interval tree after step 1

step 2: rearrange H to satisfy immediate parent property (Condition 2)

- 9: **for** $k = l+1$ to 2 **do**
- 10: **for** each facility $f \in G_k$ **do**
- 11: **for** each child (h, k') of (f, k) in H with $k' < k-1$ **do**
- 12: Let $(g, k-1) \in \overline{G}_{k-1}$ be the child of (f, k) closest to (h, k')
- 13: Remove edge $((f, k), (h, k'))$ from H and add edge $((g, k-1), (h, k'))$ to H
- 14: Update $A(g, k-1) \leftarrow \text{conv}(A(g, k-1) \cup A(h, k'))$

Lemma 10: H is an interval tree and satisfies the immediate parent condition after Step 2

step 3: Assign any unassigned intervals to satisfy completeness (Condition 3)

- 15: COMPLETEASSIGNMENT($(f_0, l+1), H$)
 - 16: **return** assignments Π_1, \dots, Π_l induced by intervals $A(f, k)$ for $(f, k) \in \overline{G}$
-

Algorithm 3 COMPLETEASSIGNMENT($(f, k), H$)

- 1: **input:** Facility, level pair (f, k) and rooted tree H on vertices \overline{G}
 - 2: Let $(f_1, k-1), \dots, (f_t, k-1)$ be the children of (f, k) in H . Expand intervals $A(f_1, k-1), \dots, A(f_t, k-1)$ arbitrarily so that they partition $A(f, k)$.
 - 3: **for** children $f_s \in \{f_1, \dots, f_t\}$ **do**
 - 4: COMPLETEASSIGNMENT($((f_s, k-1), H)$)
 - 5: **return**
-

child of (f, k) of the form $(g, k-1)$ and rearranges the tree H so that $(g, k-1)$ is the new parent of (h, k') . Doing this iteratively from higher levels to lower levels $k = l+1, \dots, 2$ ensures

that H satisfies the immediate-parent condition while still being an interval tree. We prove this in Lemma 10.

3. Finally, step 3 fills any ‘gaps’, i.e., assigns any points not covered for any $k \in [l]$ by $\bigcup_{f \in G_k} A(f, k)$ by expanding these intervals recursively, thus satisfying the completeness condition (3) as well. We record this in Lemma 11. By our discussion above, this implies that the assignments now form an $O(l)$ -approximate strong refinement.

These steps are illustrated using a minimal working example in Figure 4. We defer the lemmas and their proofs to Appendix D.

,=

4 Trade-offs between opening and connection costs

In this section, we discuss trade-offs between opening and connection costs as a function of p by considering a weighting coefficient for the connection cost. We characterize the monotonicity of OPT as a function of p in terms of this weighting coefficient (Theorem 7). We also provide an alternate model for fair facility location motivated by this analysis. We provide analogous results for the number of solutions and refinements for this proposed new model (Corollary 4, Theorems 8, 9).

Consider the following simple example: the underlying metric is the line segment $[0, 1]$. There are n clients at $x = 0$; each client is the only one in its group. There are two facilities, at $x = 0$ and $x = 1$. The facility at $x = 0$ has opening cost \sqrt{n} and the facility at $x = 1$ has opening cost 1.

The optimal 1-norm solution opens the facility at $x = 0$ with total cost \sqrt{n} . Note that the assignment cost is 0.

This is, however, a bad approximation for ∞ -norm objective. Indeed, it is easy to verify that the *only* $O(1)$ -approximate solution for ∞ -norm is to open the facility at $x = 1$, with the total cost 2. However, each client is now at a greater distance from its nearest open facility. Thus, fairness is being achieved by making everyone worse off.

There is an explanation for this behavior: for a given solution, as norm p increases, the assignment cost decreases while the facility cost stays the same. Therefore, there is higher incentive to minimize the facility cost as opposed to the assignment cost for higher p .

4.1 Normalized fair facility location

We can introduce a normalizing factor to balance the facility and assignment costs as p varies: for a solution $\mathcal{S} = (F', \Pi)$ and client groups D_1, \dots, D_r , we have the assignment cost vector

$$V_{\Pi} = \left(\frac{1}{|D_1|} \sum_{j \in D_1} d(j, \Pi(j)), \dots, \frac{1}{|D_r|} \sum_{j \in D_r} d(j, \Pi(j)) \right) \in \mathbb{R}^r.$$

With a nonnegative *normalizing factor* $g(p; r)$, we define the p -norm cost function to be

$$C_S = C_{F'} + g(p; r) \|V_\Pi\|_p,$$

and the objective is to minimize this cost function across solutions.

A following question is the choice of the function $g(p; r)$. We first characterize the monotonicity of OPT as a function of p , which then leads to a natural choice of $g(p; r)$ that resolves the aforementioned unwanted behavior.

Theorem 7. 1. *OPT is a nonincreasing function of p if $g(p; r)$ is a nonincreasing function of p . Further, if $g(p; r)$ is not a nonincreasing function of p , then there exist problem instances where OPT is not a nonincreasing function of p .*

2. *OPT is a nondecreasing function of p if $r^{\frac{1}{p}}g(p; r)$ is a nondecreasing function of p . Further, if $r^{\frac{1}{p}}g(p; r)$ is not a nondecreasing function of p , then there exist problem instances where OPT is not a nondecreasing function of p .*

Proof. 1. Suppose $g(p; r)$ is a nonincreasing function of p . We show that $OPT_q \leq OPT_p$ for all norms $q > p$ for all problem instances with r groups. Given such a problem instance, let F_p, V_p respectively denote the set of open facilities and the assignment cost vector for the optimal solution for the p -norm problem. Then,

$$OPT_q \leq F_p + g(q; r) \|V_p\|_q \leq F_p + g(p; r) \|V_p\|_q \leq F_p + g(p; r) \|V_p\|_p = OPT_p.$$

The first inequality holds by definition of OPT_q , the second inequality holds since g is a nonincreasing function, and the final inequality holds by Lemma 1 and since g is nonnegative.

Suppose $g(p; r)$ is not nonincreasing, then there exist norms $p, q \in [1, \infty]$ with $p < q$ and $g(p; r) < g(q; r)$. Consider the line segment $[0, 1]$. There is a single facility at $x = 0$ with opening cost 0, with $r - 1$ clients at $x = 0$ and 1 client at $x = 1$. The only possible solution is to open the facility at $x = 0$, with $OPT_p = 0 + g(p; r) \times 1 = g(p; r)$ and similarly $OPT_q = g(q; r)$. Since $g(p; r) < g(q; r)$, we have $OPT_p < OPT_q$, so that OPT is not nonincreasing in p .

2. Suppose $r^{\frac{1}{p}}g(p; r)$ is a nondecreasing function of p . We show that $OPT_q \geq OPT_p$ for all norms $q > p$ for all problem instances with r groups. Given such a problem instance, let F_q, V_q respectively denote the set of open facilities and the assignment cost vector for the optimal solution for the q -norm problem. Then,

$$OPT_p \leq F_q + g(p; r) \|V_q\|_p \leq F_q + g(p; r) r^{\frac{1}{p} - \frac{1}{q}} \|V_q\|_q \leq F_q + g(q; r) \|V_q\|_q = OPT_q.$$

The first inequality holds by definition of OPT_p , the second inequality holds by Lemma 1 and since g is nonnegative, and the final inequality holds since $r^{\frac{1}{p}}g(p; r)$ is a nondecreasing function.

Suppose $r^{\frac{1}{p}}g(p; r)$ is not nondecreasing, then there exist norms $p, q \in [1, \infty]$ with $p < q$ and

$r^{\frac{1}{p}}g(p; r) > r^{\frac{1}{q}}g(q; r)$. Consider the star graph with r leaf vertices and unit distances between the central vertex and any any leaf. There is a single facility at $x = 0$ with opening cost 0. There is one client each at each of the leaf vertices, with each client in a singleton group. The only possible solution is to open the facility at $x = 0$, with $OPT_p = 0 + g(p; r) \times \|(1, \dots, 1)\|_p = r^{\frac{1}{p}}g(p; r)$ and similarly $OPT_q = r^{\frac{1}{q}}g(q; r)$. Since $r^{\frac{1}{p}}g(p; r) > r^{\frac{1}{q}}g(q; r)$, we have $OPT_p > OPT_q$, so that OPT is not nondecreasing in p . \square

This suggests $g(p; r) = r^{1-\frac{1}{p}}$ as a natural choice for the normalizing factor. Lemma 1, which gives a fundamental relationship between different p -norms in Cartesian spaces suggests that the factor $r^{\frac{1}{p}-\frac{1}{q}}$ can offset the decrease in assignment cost as the norm is increased. We can normalize this with respect to the 1-norm and recover our choice of the normalizing factor.

Formally, we define the *normalized p -norm fair facility location problem* as an instance of the facility location problem with objective

$$\min_{(F', \Pi)} \sum_{i \in F'} c_i + r^{1-\frac{1}{p}} \left\| \left(\frac{1}{|D_s|} \sum_{j \in D_s} d(j, \Pi(j)) \right)_{s \in [r]} \right\|_p,$$

that is, the problem seeks to minimize across all solutions the sum of facility costs and the p -norm of the vector of group distances weighted by $r^{1-1/p}$.

With our notation for facility and assignment costs, the cost for solution $\mathcal{S} = (F', \Pi)$ is $C_{\mathcal{S}} = C_{F'} + r^{1-\frac{1}{p}}C_{\Pi}$. Notice that for $p = 1$ and singleton groups (i.e., each client is in a separate group), this reduces to the classic facility location problem.

Theorem 1 immediately implies that this problem is 4-approximable:

Corollary 4. *There is a polynomial-time algorithm that gives a 4-approximation for the normalized p -norm fair facility location problem for any $p \in [1, \infty]$.*

As with our original model, we can establish more structure across approximate solutions for different p -norms. Theorem 7 already tells us that OPT_p is a nondecreasing function of p . Analogous to Theorem 2, we can once again get a portfolio of $\lceil \log_2 r \rceil$ solutions such that for each norm $p \in [1, \infty]$, one of these solutions is an 8-approximation to the p -norm fair problem. We omit the proof as it is similar to the proof of Theorem 2.

Theorem 8. *Given any instance of the facility location problem, there exists an 8-approximate solution set of cardinality at most $\lceil \log_2 r \rceil$ for the normalized fair facility location problem. This solution set can be obtained in polynomial time.*

The advantage of the normalized problem is most significant for refinements. Since OPT_p is an increasing function of p , we expect that more facilities open as norm p increases, as opposed to less facilities being open for our original model. In terms of fairness, this implies that fairness is achieved

(i.e. the norm increases) by opening more facilities instead of closing them. This motivates the definition of *increasing refinements*:

Definition (Increasing refinements). We are given a metric space $\mathcal{M} = (X, d)$, client set $D \subseteq X$, and non-empty facility set $F \subseteq X$ with non-negative facility costs $c : F \rightarrow \mathbb{R}$. We are given a metric space $\mathcal{M} = (X, d)$, client set $D \subseteq X$, and non-empty facility set $F \subseteq X$ with non-negative facility costs $c : F \rightarrow \mathbb{R}$. Furthermore, suppose we are given non-empty client groups D_1, \dots, D_r that partition D . Given a set of norms $P \subseteq [1, \infty]$, a solution set $S = \{\mathcal{S}_p = (F_p, \Pi_p) : p \in P\}$ is called

1. a *weak increasing refinement* if for all $p, q \in P$ with $p \leq q$, $F_p \subseteq F_q$; that is, the facilities form a subset structure.
2. a *strong increasing refinement* if for all $p, q \in P$ with $p \leq q$ and for each facility $i \in F_q$, there is some facility $i' \in F_p$ such that all clients assigned to i under Π_q are assigned to i' under Π_p , that is,

$$\{j \in D : \Pi_q(j) = i\} \subseteq \{j \in D : \Pi_p(j) = i'\}.$$

If for each norm $p \in P$, the solution \mathcal{S}_p is an α -approximation for the p -norm fair problem, then we say that the refinement is an α -approximate increasing refinement.

Simple changes to our algorithms lead to analogous results for increasing refinements for the normalized fair facility location problem:

- Theorem 9.**
1. *There is 16-approximate increasing weak refinement for the norm set $[1, \infty]$ for the normalized fair facility location problem.*
 2. *There is a $\text{poly}(r^{\frac{1}{\sqrt{\log_2 r}}})$ -approximate strong increasing refinement for the norm set $[1, \infty]$ for the normalized fair facility location problem for arbitrary metrics.*
 3. *There is an $O(\log_2 r)$ -approximate strong increasing refinement for the norm set $[1, \infty]$ for the normalized fair facility location problem on the line metric.*
 4. *There is an $O(\log_2 r)$ -approximate strong increasing refinement for the norm set $[1, \infty]$ for the normalized fair facility location problem on the tree metric, assuming uniform facility cost and that a facility can be opened anywhere on the tree.*

These increasing refinements can be obtained in polynomial time.

Remark: We can also consider the related clustering model where at most k facilities can be opened and the objective is to minimize the p -norm of the assignment vector, generalizing problems such as k -median, k -means, and k -center. Analogous to Theorem 2, we can show that at most $\lceil \log_2 r \rceil$ solutions are required such that for any p , one of these solutions is a constant-factor approximation. By combining these solutions, we get a solution with $O(k \log r)$ open facilities such that it is an $O(1)$ -approximation for all p -norms simultaneously; this is already known ([25, 26]).

5 Hierarchical facility location problem

In this section, we extend the algorithms for refinements to the hierarchical facility location problem defined in Section 1.1. We give approximation algorithms for various metrics; the results are presented in Theorem 10.

Given an instance of the hierarchical facility location problem with l levels, 4-approximations for the individual facility location problem at each level $k \in [l]$ can be obtained using Theorem 1 when the function f is sublinear and polynomial-time oracles for f and ∇f are available. Suppose $\mathcal{S}_k = (F_k, \theta_k)$ is a 4-approximation to the level k facility location problem (i.e., with facility costs $c_k \in \mathbb{R}_+$).

Note that since $c_1 \leq \dots \leq c_l$, we must have $OPT_1 \leq \dots \leq OPT_l$, where OPT_k is the value of the optimal solution to the level k problem. This allows us to get a subset structure in facility sets as in Theorem 4. That is, let $G_k = \cup_{t \in [k, l]} F_t$ and consider solutions $\mathcal{R}_k = (G_k, \theta_k)$ for the level k problem, $k \in [l]$. Solutions \mathcal{R}_k satisfy constraint (1) in the definition of hierarchical facility location. We will show that $C_{\mathcal{R}_k} \leq 4l C_{\mathcal{S}_k} \leq 16l OPT_k$ for all k .

For any $t \geq k$, since \mathcal{S}_t is a 4-approximation to the level t problem and since \mathcal{S}_k is a feasible solution to the level t problem, we have

$$C_{\mathcal{S}_t} = c_t |F_t| + C_{\Pi_t} \leq 4 OPT_t \leq 4(c_t |F_k| + C_{\Pi_k}),$$

implying that $|F_t| \leq 4|F_k| + \frac{4}{c_t} C_{\Pi_k}$. Therefore, using the monotonicity assumption $c_k \leq c_t$ for all $t \geq k$, we get

$$\begin{aligned} C_{\mathcal{R}_k} &= c_k |G_k| + C_{\Pi_k} \leq c_k \sum_{t \geq k} |F_t| + C_{\Pi_k} \\ &= C_{\mathcal{S}_k} + c_k \sum_{t > k} |F_t| \leq C_{\mathcal{S}_k} + c_k \sum_{t > k} \left(4|F_k| + \frac{4}{c_t} C_{\Pi_k} \right) \\ &\leq C_{\mathcal{S}_k} + \sum_{t > k} 4(c_k |F_k| + C_{\Pi_k}) \leq 4l C_{\mathcal{S}_k}. \end{aligned}$$

If we can change the assignments $\Pi_k, k \in [r]$ so that they satisfy constraint (2) in the definition, we will get a feasible solution to hierarchical facility location problem. We claim that our algorithms for refinements are sufficient for this reassignment: indeed, none of the strong refinement algorithms DISCOUNTEDLOOKAHEAD, EXPANDINTERVALS, and BRANCHANDLINEARIZE use the norm structure of p -norms for reassigning clients. If the reassignment increases the cost of Π_k by a factor at most β for each k , we get that \mathcal{R}_k is an $O(\beta l)$ -approximation for the level k problem, and therefore the set of solutions $R = \{\mathcal{R}_k : k \in [l]\}$ forms an $O(\beta l)$ -approximation for the hierarchical facility location problem. Therefore, we have the following theorem:

Theorem 10. *Given an instance of the hierarchical facility location problem with l levels and a differentiable sublinear function f with polynomial-time oracles for f and ∇f ,*

1. there is a polynomial-time algorithm that is an $\tilde{O}(e^{3\sqrt{l}})$ -approximation for arbitrary metrics,
2. there is a polynomial-time algorithm that is an $O(l^2)$ -approximation for the line metric, and
3. there is a polynomial-time algorithm that gives an $O(l^2)$ -approximation for the tree metric, assuming that a facility can be opened anywhere on the tree.

6 Open questions

A natural open question is about the existence of a solution set with cardinality lesser than $\lceil \log_2 r \rceil$ (Theorem 2, Section 2) that is constant-factor approximate for $P = [1, \infty]$. In particular, given any instance of the fair facility location problem, what is the order of the smallest number of solutions required: is it $\Theta(\sqrt{\log_2 r})$, or $\Theta(\log_2 r)$, or something between the two?

Another open question is whether the subexponential factor $e^{3\sqrt{l}}$ for strong refinement in Theorem 5 (Section 3.1) can be reduced to a polynomial factor. We believe that this should be true (potentially by searching a larger local neighborhood instead of a one-step for the Greedy algorithm):

Conjecture 1. *Given a finite set $P = \{p_1, \dots, p_l\}$ of norms, there is a polynomial-time algorithm that gives a $\text{poly}(l)$ -approximation to the strong refinement problem for arbitrary metrics.*

This can potentially (but not necessarily) lead to an improvement for the corresponding factor for the hierarchical facility location problem in Section 5. For strong refinements, it is natural to ask if the algorithm for line metric generalizes to Cartesian spaces \mathbb{R}^d , $d > 1$, after perhaps losing an additional factor that depends on d (2^d , for instance).

Finally, it is natural to ask the basic question about the approximability of the general facility location problem (see Theorem 1): can it be approximated to a factor better than 4 in polynomial time? Several improved approximation factors are known for the classic facility location problem [48], including the 1.488-approximation algorithm of Li. Most of these algorithms crucially use the linearity of the objective, and therefore do not generalize.

References

- [1] UNOS policy: OPTN policy changes and organ allocation guidelines, December 2021.
- [2] K. Aardal, F.A. Chudak, and D.B. Shmoys. A 3-approximation algorithm for the k-level uncapacitated facility location problem. *Information Processing Letters*, 72(5):161–167, 1999.
- [3] M. Abbasi, A. Bhaskara, and S. Venkatasubramanian. Fair clustering via equitable group representations. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT, 2021.

- [4] A.A. Ageev. Improved approximation algorithms for multilevel facility location problems. *Operations Research Letters*, 30(5):327–332, 2002.
- [5] A.A. Ageev, Y. Ye, and J. Zhang. Improved combinatorial approximation algorithms for the k-level facility location problem. *SIAM Journal on Discrete Mathematics*, 18(1):207–217, 2004.
- [6] M. Barbati and C. Piccolo. Equality measures properties for location problems. *Optimization Letters*, 10, June 2016.
- [7] D. Bertsimas, V. Farias, and N. Trichakis. The price of fairness. *Operations Research*, 59:17–31, February 2011.
- [8] S. Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3–4):231–357, 2015.
- [9] A. Bumb. An approximation algorithm for the maximization version of the two level uncapacitated facility location problem. *Operations Research Letters*, 29(4):155–161, 2001.
- [10] R.E. Burkard, J. Krarup, and P.M. Pruzan. Efficiency and optimality in minisum, minimax 0-1 programming problems. *Journal of the Operational Research Society*, 33(2):137–151, 1982.
- [11] J. Byrka. An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem. APPROX ’07/RANDOM ’07, page 29–43, Berlin, Heidelberg, 2007. Springer-Verlag.
- [12] M. Charikar and S. Guha. Improved combinatorial algorithms for the facility location and k-median problems. In *In Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science*, pages 378–388, 1999.
- [13] F.A. Chudak and D.B. Shmoys. Improved approximation algorithms for the uncapacitated facility location problem. *SIAM Journal on Computing*, 33(1):1–25, 2003.
- [14] J. Corburn, A. Fukutome, M.R. Asari, J. Jarin, and V. Villaverde. Rapid health impact assessment: Proposed closure of Alta Bates Campus Berkeley, CA. Technical report, Institute of Urban and Regional Development University of California Berkeley, 2018.
- [15] G. Cornuejols, G. Nemhauser, and L. Wolsey. The uncapacitated facility location problem. Technical report, Cornell University Operations Research and Industrial Engineering, 1983.
- [16] P. Dutko, M.V. Ploeg, and T. Farrigan. Characteristics and influential factors of food deserts. *Economic Research Report*, 140, 2012.
- [17] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS, page 214–226, New York, NY, USA, 2012. Association for Computing Machinery.

- [18] C. Dwork and C. Ilvento. Group fairness under composition. In *Proceedings of the 2018 Conference on Fairness, Accountability, and Transparency (FAT*)*, 2018.
- [19] R.Z. Farahani, M. Hekmatfar, B. Fahimnia, and N. Kazemzadeh. Hierarchical facility location problem: Models, classifications, techniques, and applications. *Computers & Industrial Engineering*, 68:104–117, 2014.
- [20] M. Farhadi, A. Toriello, and P. Tetali. The traveling firefighter problem. In *SIAM Conference on Applied and Computational Discrete Algorithms (ACDA21)*, pages 205–216. SIAM, 2021.
- [21] C. Filippi, G. Guastaroba, D.L. Huerta-Muñoz, and M.G. Speranza. A kernel search heuristic for a fair facility location problem. *Computers & Operations Research*, 132:105292, 2021.
- [22] A.F. Gabor and Jan-Kees C.W. van Ommeren. A new approximation algorithm for the multilevel facility location problem. *Discrete Applied Mathematics*, 158(5):453–460, 2010.
- [23] M. Ghadiri, S. Samadi, and S.S. Vempala. Socially fair k-means clustering. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT ’21, page 438–448, New York, NY, USA, 2021. Association for Computing Machinery.
- [24] M. Ghadiri, M. Singh, and S.S. Vempala. Constant-factor approximation algorithms for socially fair k -clustering. *arXiv*, 2022.
- [25] A. Goel and A. Meyerson. Simultaneous optimization via approximate majorization for concave profits or convex costs. *Algorithmica*, 44(4):301–323, 2006.
- [26] D. Golovin, A. Gupta, A. Kumar, and K. Tangwongsan. All-Norms and All- L_p -Norms Approximation Algorithms. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 199–210, Dagstuhl, Germany, 2008.
- [27] S. Guha and S. Khuller. Greedy strikes back: Improved facility location algorithms. In *Journal of Algorithms*, pages 649–657, 1998.
- [28] S. Gupta, A. Jalan, G. Ranade, H. Yang, and S. Zhuang. Too many fairness metrics: Is there a solution? In *Ethics of Data Science Conference EDSC*, 2020.
- [29] D.S. Hochbaum. Heuristics for the fixed cost median problem. *Math. Program.*, 22(1):148–162, dec 1982.
- [30] K. Jain, M. Mahdian, E. Markakis, A. Saberi, and V.V. Vazirani. Greedy facility location algorithms analyzed using dual fitting with factor-revealing lp. *J. ACM*, 50(6):795–824, nov 2003.
- [31] C. Jung, S. Kannan, and N. Lutz. A center in your neighborhood: Fairness in facility location. *arXiv*, abs/1908.09041, 2019.

- [32] E. Kantor and D. Peleg. Approximate hierarchical facility location and applications to the bounded depth steiner tree and range assignment problems. *Journal of Discrete Algorithms*, 7(3):341–362, 2009. Special Issue on the 6th Italian Conference on Algorithms and Complexity (CIAC 2006).
- [33] J. Kleinberg, S. Mullainathan, and M. Raghavan. Inherent trade-offs in the fair determination of risk scores. In *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*.
- [34] M.R. Korupolu, C.G. Plaxton, and R. Rajaraman. Analysis of a local search heuristic for facility location problems. *J. Algorithms*, 37:146–188, 1998.
- [35] J.G. Lee, Y. Roh, H. Song, and S.E. Whang. Machine learning robustness, fairness, and their convergence. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD ’21, page 4046–4047. Association for Computing Machinery, 2021.
- [36] S. Li. A 1.488 approximation algorithm for the uncapacitated facility location problem. *Information and Computation*, 222:45–58, 2013. 38th International Colloquium on Automata, Languages and Programming (ICALP 2011).
- [37] J.H. Lin and J.S. Vitter. Approximation algorithms for geometric median problems. *Information Processing Letters*, 44(5):245–249, 1992.
- [38] M. Mahdian, Y. Ye, and J. Zhang. Approximation algorithms for metric facility location problems. *SIAM Journal on Computing*, 36(2):411–432, 2006.
- [39] Y. Makarychev and A. Vakilian. Approximation algorithms for socially fair clustering. In *Proceedings of Thirty Fourth Conference on Learning Theory*, pages 3246–3264. PMLR, 2021.
- [40] M.T. Marsh and D.A. Schilling. Equity measurement in facility location analysis: A review and framework. *European Journal of Operational Research*, 74(1):1–17, 1994.
- [41] Włodzimierz Ogryczak. On the lexicographic minimax approach to location problems. *European Journal of Operational Research*, 100(3):566–585, 1997.
- [42] M. Repko. Walgreens strikes deal with primary-care company to open doctor offices in hundreds of drugstores. *CNBC*, 2020.
- [43] D.B. Shmoys, É. Tardos, and K. Aardal. Approximation algorithms for facility location problems (extended abstract). In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, STOC ’97, page 265–274, New York, NY, USA, 1997. Association for Computing Machinery.
- [44] J.M. Steele. *The Cauchy-Schwarz master class: an introduction to the art of mathematical inequalities*. Cambridge University Press, 2004.

- [45] M. Sviridenko. An improved approximation algorithm for the metric uncapacitated facility location problem. *Proceedings of the 9th International IPCO Conference on Integer Programming and Combinatorial Optimization*, 1997.
- [46] C. Wang, X. Wu, M. Li, and H. Chan. Facility’s perspective to fair facility location problems. In *Thirty-Fifth AAAI Conference on Artificial Intelligence*. The AAAI Press, 2021.
- [47] Z. Wang, D. Du, A.F. Gabor, and D. Xu. An approximation algorithm for the k-level stochastic facility location problem. *Operations Research Letters*, 38(5):386–389, 2010.
- [48] D.P. Williamson and D.B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011.
- [49] J. Zhang. Approximating the two-level facility location problem via a quasi-greedy approach. SODA 2004. Society for Industrial and Applied Mathematics, 2004.
- [50] G. Şahin and H. Süral. A review of hierarchical facility location models. *Computers & Operations Research*, 34(8):2310–2331, 2007.

A Proof of Theorem 1

We can formulate the problem as an integer program: for each facility $i \in F$, variable $y_i \in \{0, 1\}$ indicates whether facility i is opened and for each client $j \in D$, variable $x_{ij} \in \{0, 1\}$ indicates whether j has been assigned to i . The feasible region is

$$\begin{aligned} \sum_{i \in F} x_{ij} &\geq 1, & \forall j \in D, \\ x_{ij} &\leq y_i, & \forall i \in F, j \in D, \\ x_{ij}, y_i &\in \{0, 1\}, & \forall i \in F, j \in D, \end{aligned}$$

and the objective is to minimize the function $\sum_{i \in F} c_i y_i + f\left(\left(\sum_{i \in F} x_{ij} d(i, j)\right)_{j \in D}\right)$.

In the relaxation, we allow variables x, y to take values in $[0, 1]$:

$$\min_{x, y} \sum_{i \in F} c_i y_i + f\left(\left(\sum_{i \in F} x_{ij} d(i, j)\right)_{j \in D}\right) \quad \text{subject to} \quad (\text{CP})$$

$$\sum_{i \in F} x_{ij} \geq 1, \quad \forall j \in D, \quad (8)$$

$$x_{ij} \leq y_i, \quad \forall i \in F, j \in D, \quad (9)$$

$$0 \leq x, y. \quad (10)$$

Let us denote the underlying polytope by P and the objective function by $\bar{f} : P \rightarrow \mathbb{R}$. Since P

Algorithm 4 FILTER(x, y)

```
1: Set  $\alpha = \frac{1}{4}$ 
2: for  $j \in D$  do
3:   Let  $S_j = \{i \in F : x_{ij} > 0\}$ 
4:   Let  $\sigma$  be the permutation arranging facilities in  $S_j$  such that  $d_{\sigma(1)j} \leq d_{\sigma(2)j} \leq \dots$ 
5:   Let  $i_j = \arg \min_k \sum_{k' \in [k]} x_{\sigma(k')j} \geq \alpha$ , and let  $d_j(\alpha) = d_{\sigma(i_j),j}$ 
6: for  $i \in F$  do
7:   Set  $\bar{y}_i \leftarrow \frac{1}{\alpha} y_i$ 
8:   for  $j \in D$  do
9:     Set  $\bar{x}_{ij}$  to  $\frac{1}{\alpha} x_{ij}$  if  $d_{ij} \leq d_j(\alpha)$  and 0 otherwise
10: return  $(\bar{x}, \bar{y})$ 
```

has a polynomial number of constraints, the Ellipsoid algorithm (see [8]) can be used to obtain the following theorem:

Theorem 11. *For a differentiable convex function $f : \mathbb{R}^D \rightarrow \mathbb{R}$, an optimal solution to (CP) can be obtained in a polynomial number of calls to f and ∇f .*

For the classical facility location problem with a linear objective, Shmoys *et al.* [43] use the filtering technique first introduced by Lin and Vitter [37] to round a fractional solution to (CP), giving a 4-approximation. We show that their rounding algorithm generalizes to a much larger class of convex functions:

Lemma 6. *If f is a sublinear function, then a fractional solution (x, y) to (CP) can be rounded to an integral solution (x^*, y^*) with cost $\bar{f}(x^*, y^*) \leq 4\bar{f}(x, y)$ in polynomial time.*

This, along with Theorem 11 implies Theorem 1 for the general facility location problem and the corollary for p -norm socially fair facility location problem.

We present the algorithm now and then proceed to prove Lemma 6.

Proof of Lemma 6. Consider the solution (\bar{x}, \bar{y}) output by FILTER. Since $\bar{x} \leq \frac{1}{\alpha} x$ and $\bar{y} = \frac{1}{\alpha} y$, it satisfies the feasibility constraints (9) and (10). Further,

$$\sum_{i \in F} \bar{x}_{ij} = \sum_{i: d_{ij} \leq d_j(\alpha)} \frac{1}{\alpha} x_{ij} = \frac{1}{\alpha} \sum_{i: d_{ij} \leq d_j(\alpha)} x_{ij} \geq \frac{1}{\alpha} \times \alpha = 1$$

by definition of $d_j(\alpha)$. Therefore, \bar{x}, \bar{y} is feasible.

Furthermore

$$\sum_{i \in F} d_{ij} x_{ij} \geq \sum_{i: d_{ij} \geq d_j(\alpha)} d_{ij} x_{ij} \geq d_j(\alpha) \sum_{i: d_{ij} \geq d_j(\alpha)} x_{ij} \geq d_j(\alpha)(1 - \alpha). \quad (11)$$

Let the solution (x^*, y^*) output by Algorithm ROUND. For the facility i_k and client j_k chosen in

Algorithm 5 ROUND(\bar{x}, \bar{y})

- 1: Initialize $D' \leftarrow D$
 - 2: Form a bipartite graph G with vertex sets D', F' with $ji \in E(G)$ iff $\bar{x}_{ij} > 0$ for all $i \in F, j \in D$
 - 3: For $j \in D$, denote by $N_G(j) = \{i \in F : \bar{x}_{ij} > 0\}$ the neighborhood of j in G and by $N_G^2(j)$ the set of all $j' \in D$ that share a neighbor with j in G , i.e., the set of all $j' \in D$ with $N_G(j') \cap N_G(j) \neq \emptyset$
 - 4: $k = 0$ ▷ Loop counter
 - 5: **while** $D' \neq \emptyset$ **do**
 - 6: Choose $j_k = \arg \min_{j \in D'} g_j$
 - 7: Choose $i_k = \arg \min_{i \in N_G(j_k)} f_i$
 - 8: Open i_k and assign all $j \in N_G^2(j_k)$ to i_k
 - 9: $D' \leftarrow D' \setminus (N_G^2(j_k) \cup \{j_k\})$, update G
 - 10: Set $y_i^* = 1$ for all opened facilities i , set $x_{ij}^* = 1$ if j was assigned to i , and set all other variables to 0
 - 11: **return** (x^*, y^*)
-

iteration k of loop 5, by step 7 and since $\sum_{i \in F} \bar{x}_{ij_k} \geq 1$,

$$c_{i_k} \leq c_{i_k} \sum_{i \in F} \bar{x}_{ij_k} \leq \sum_{i \in N_G(j_k)} c_i \bar{x}_{ij_k} \leq \sum_{i \in N_G(j_k)} c_i \bar{y}_i.$$

Furthermore, since each client in set $N_G^2(j_k)$ is assigned to i_k and then removed from G , the client sets $N_G(j_1), N_G(j_2), \dots$ are disjoint. Along with the above, this implies

$$\sum_k c_{i_k} \leq \sum_k \sum_{i \in N_G(j_k)} c_i \bar{y}_i \leq \sum_{i \in F} c_i \bar{y}_i = \frac{1}{\alpha} \sum_{i \in F} c_i y_i. \quad (12)$$

Suppose client j was assigned to i_k . Then $j \in N_G^2(j_k)$ and by step 6, $d_j(\alpha) \geq d_{j_k}(\alpha)$. Since $j \in N_G^2(j_k)$, there exists some $i \in N_G(j) \cap N_G(j_k)$. By metric property of d , $d_{i_k j} \leq d_{i_k j_k} + d_{i j_k} + d_{i j}$. Further, by step 9 and since $\bar{x}_{ij}, \bar{x}_{ij_k}, \bar{x}_{i_k j_k} > 0$, we must have $d_{ij} \leq d_j(\alpha)$, $d_{ij_k} \leq d_{j_k}(\alpha)$ and $d_{i_k j_k} \leq d_{j_k}(\alpha)$, so that

$$d_{i_k j} \leq 2d_{j_k}(\alpha) + d_j(\alpha) \leq 3d_j(\alpha).$$

Eqn. (11) then implies that $d_{i_k j} \leq 3d_j(\alpha) \leq \frac{3}{1-\alpha} \sum_{i \in F} x_{ij} d_{ij}$, or that $d_{i_k j} = \sum_{i \in F} x_{ij}^* d_{ij} \leq \frac{3}{1-\alpha} \sum_{i \in F} x_{ij} d_{ij}$. Since f is nondecreasing and sublinear,

$$f \left(\left(\sum_i x_{ij}^* d_{ij} \right)_{j \in D} \right) \leq f \left(\left(\frac{3}{1-\alpha} \sum_i x_{ij} d_{ij} \right)_{j \in D} \right) = \frac{3}{1-\alpha} f \left(\left(\sum_i x_{ij} d_{ij} \right)_{j \in D} \right).$$

By eqn. (12),

$$\sum_{i \in F} c_i y_i^* = \sum_k c_{i_k} \leq \frac{1}{\alpha} \sum_{i \in F} c_i y_i.$$

Together, for $\alpha = \frac{1}{4}$, this implies that $\bar{f}(x^*, y^*) \leq 4f(x, y)$. □

B Missing proofs from Section 2

Proof of Lemma 2. Let OPT_p, OPT_q denote the corresponding optimum values. Let $\mathcal{S}_p = (F_p, \Pi_p)$ be an α -approximate solution to the p -norm fair problem, and let C_p denote the assignment cost vector for it (so that the p -norm assignment cost of Π_p is $\|C_p\|_p$). Let $\mathcal{S}_q^* = (F_q^*, \Pi_q^*)$ be an optimum solution to the q -norm fair problem, with assignment cost vector C_q^* .

We have from definition that $C_{F_p} + \|C_p\|_p \leq \alpha OPT_p$. Now,

$$\begin{aligned} C_{F_p} + \|C_p\|_q &\leq C_{F_p} + \|C_p\|_p \leq \alpha OPT_p \\ &\leq \alpha (C_{F_q^*} + \|C_q^*\|_p) \leq \alpha (C_{F_q^*} + r^{\frac{1}{p}-\frac{1}{q}} \|C_q^*\|_q) \\ &\leq \alpha r^{\frac{1}{p}-\frac{1}{q}} (C_{F_q^*} + \|C_q^*\|_q) = \alpha r^{\frac{1}{p}-\frac{1}{q}} OPT_q. \end{aligned}$$

where the first inequality follows from Lemma 1 and since $q > p$, the second inequality is true since s_p is α -approximate, the third inequality follows from definition, the fourth inequality follows from Lemma 1, the fifth inequality is true since $\frac{1}{p} - \frac{1}{q} > 0$, and the final equality is true since \mathcal{S}_q^* is an optimal solution to the q -norm fair problem. Therefore, \mathcal{S}_p is a $\alpha r^{\frac{1}{p}-\frac{1}{q}}$ -approximate solution to q -norm fair problem.

Similarly let $\mathcal{S}_q = (F_q, \Pi_q)$ be an α -approximate solution to the q -norm fair problem with cost vector C_q . Then, using the fact that $OPT_p \geq OPT_q$,

$$C_{F_q} + \|C_q\|_p \leq C_{F_q} + r^{\frac{1}{p}-\frac{1}{q}} \|C_q\|_q \leq r^{\frac{1}{p}-\frac{1}{q}} (C_{F_q} + \|C_q\|_q) \leq \alpha r^{\frac{1}{p}-\frac{1}{q}} OPT_q \leq \alpha r^{\frac{1}{p}-\frac{1}{q}} OPT_p.$$

Therefore, \mathcal{S}_q is an $\alpha r^{\frac{1}{p}-\frac{1}{q}}$ -approximation to the p -norm fair problem □

C Missing proofs from Section 3

Proof of Lemma 4. First, note that for any $t \in [1, l]$, since $\gamma > 1$, we have

$$u_{t-1,t} \leq \dots \leq u_{1,t} \leq u_{0,t},$$

so that it is enough to prove that $\max_t u_{0,t} \leq e^{\frac{2\gamma}{\gamma-1}} \gamma^{l-1}$. Further, since $u_{t-1,t} = \gamma^{l-t}$, this implies

$$u_{k,t} = \max_{k < s < t} \gamma^{s-t} + u_{s,t} (1 + \gamma^{s-t}) \quad \forall k \in [0, t-2].$$

First, a change of indices is in order: if we define $w_{k,t} = u_{t-1-k,t}$ for $k \in [0, t-1]$, we get $w_{0,t} = \gamma^{l-t}$ and

$$w_{k,t} = \max_{s \in [0, k-1]} \gamma^{-(s+1)} + w_{s,t} (1 + \gamma^{-(s+1)}) \quad \forall k \in [1, t-1].$$

This makes it clear that in fact

$$w_{k,t} = \gamma^{-k} + w_{k-1,t}(1 + \gamma^{-k}) \quad \forall k \in [1, t-1].$$

Finally, we bound $w_{t-1,t} = u_{0,t}$. Since $w_{0,t} = \gamma^{l-t} \geq 1$, notice that

$$\frac{w_{k,t}}{w_{k-1,t}} \leq \frac{\gamma^{-k}}{w_{k-1,t}} + 1 + \gamma^{-k} \leq 1 + 2\gamma^{-k}.$$

Therefore, taking product from $k = 1$ to $t - 1$, we get

$$\frac{w_{t-1,t}}{w_{0,t}} \leq \prod_{k=1}^{t-1} (1 + 2\gamma^{-k}).$$

Denote $P_t = \prod_{k=1}^{t-1} (1 + 2\gamma^{-k})$. Using the standard bound $1 + x \leq \exp(x)$,

$$\prod_{k=1}^{t-1} (1 + 2\gamma^{-k}) \leq \prod_{k=1}^{t-1} \exp(2\gamma^{-k}) = \exp\left(2 \sum_{k=1}^{t-1} \gamma^{-k}\right) \leq \exp\left(2 \sum_{k=1}^{\infty} \gamma^{-k}\right) = \exp\left(\frac{2\gamma}{\gamma-1}\right).$$

Therefore,

$$u_{0,t} = w_{t-1,t} \leq \exp\left(\frac{2\gamma}{\gamma-1}\right) w_{0,t} = \exp\left(\frac{2\gamma}{\gamma-1}\right) \gamma^{l-1}.$$

□

Proof of Lemma 5. Choose $\gamma = 1 + \frac{1}{\sqrt{l}}$. Then $e^{\frac{2\gamma}{\gamma-1}} = e^{2\sqrt{l}(1+\frac{1}{\sqrt{l}})} = e^{2\sqrt{l}} \cdot e^2$. Also, $\gamma^{l-1} < \gamma^l = \left(1 + \frac{1}{\sqrt{l}}\right)^l \leq \left(e^{\frac{1}{\sqrt{l}}}\right)^l = e^{\sqrt{l}}$, implying that $e^{\frac{2\gamma}{\gamma-1}} \gamma^{l-1} \leq e^2 \cdot e^{3\sqrt{l}}$. □

D Missing proofs in Section 3.2

Lemma 7. For each $k \in [l]$ and each $(f, k) \in \overline{G}$, the interval $A(f, k)$ satisfies $A(f, k) \supseteq I(f, k)$ at the end of Algorithm 2. Consequently, for each client $j \in D$ and each level $k \in [l]$, we have $d(j, \Pi_k(j)) \leq 2l \cdot \min_{f \in G_k} d(j, f)$.

Proof. $A(f, k)$ is initialized to $I(f, k)$ in line 3. A is only updated in lines 6, 14, and 15, and A is expanded in each of these steps. □

We show that H is an interval tree at the end of step 1 in Lemma 9. For all facilities $(f, k) \in \overline{G}$, let the interval $A(f, k)$ be denoted by $[c(f, k), d(f, k)]$. We need the following structural lemma first:

Lemma 8. For all $k \in [l]$, if f and f' are consecutive facilities in G_k with f' to the right of f , then,

at the end of step 1,

$$d(f, k) - x(f) \leq k\alpha x, \quad x(f') - c(f', k) \leq k\alpha x, \quad c(f', k) - d(f, k) \geq (1 - 2k\alpha)x, \quad (13)$$

where $x = x(f') - x(f)$ is the distance between f and f' .

Proof. The third inequality is implied by the first two; we prove the first two inequalities by induction on k . For $k = 1$, $A(f, k) = I(f, k)$ and $A(f', k) = I(f', k)$, so that $d(f, k) = b(f, k)$ and $c(f', k) = a(f', k)$, and the result follows by definition of intervals I .

Assume the result is true for all $k' < k$. Let $(g, k_1) = \arg \max_{(h, k') \in S(f, k)} d(h, k')$. Then, by line 6 in the algorithm, we have $d(f, k) = \max \{b(f, k), d(g, k_1)\}$. Since $k_1 < k$, we have $G_k \subseteq G_{k_1}$ and so $f, g \in G_{k_1}$. By the induction hypothesis at level k_1 on facility pairs f, g and g, f' , we have

$$x(g) - c(g, k_1) \leq k_1\alpha(x(g) - x(f)), \quad d(g, k_1) - x(g) \leq k_1\alpha(x(f') - x(g)).$$

so that $d(g, k_1) - c(g, k_1) \leq k_1\alpha x$. Since $A(f, k) \cap I(h, k_1) \neq \emptyset$, we get $c(g, k_1) \leq b(f, k) = x(f) + \alpha x$. Then we get that $d(g, k_1) - x(f) \leq (k_1 + 1)\alpha x \leq k\alpha x$. Since $d(f, k) = \max \{b(f, k), d(g, k_1)\}$, this proves the first inequality in (13). The second inequality is analogously proven. \square

Observe that if $((f, k), (h, k')) \in E(H)$, then $k' < k$ by line 6 in the algorithm.

Lemma 9. *The hierarchy graph H is an interval tree at the end of step 1 in Algorithm 2.*

Proof. For each $k \in [1, l + 1]$, let H_k denote the subgraph of H induced by the vertex set $\bigcup_{k' \in [k]} G_{k'}$, i.e., facilities at level k or lower. We induct on k to prove the following (stronger) statement: each component of H_k is an interval tree with respect to intervals A , and furthermore, for two components C_1, C_2 of H_k with roots r_1, r_2 , then $A(r_1)$ and $A(r_2)$ are internally disjoint.

For $k = 1$, each component of H_1 is an isolated vertex and further, $A(f, 1) = I(f, 1)$ and $A(g, 1) = I(g, 1)$ are disjoint intervals for $f \neq g$, implying the claim.

Assume now that $k > 1$ and that the claim is true for all $k' < k$. Let C be a component of H_k rooted at some facility (f, k') , $k' \leq k$. We first prove that C is an interval tree. If $k' < k$, then C is also a component of H_{k-1} and therefore C is an interval tree by the induction hypothesis. Suppose $k' = k$. The only parent-child pairs present in $H_k \cap C$ but not in $H_{k-1} \cap C$ are (f, k) and one of its children. And the only sibling pairs present in $H_k \cap C$ but not in $H_{k-1} \cap C$ are two children of (f, k) .

Let (g_1, k_1) be some child of (f, k) in C . By definition of H , $\sigma(g_1, k_1) = (f, k)$, and therefore $A(g_1, k_1) \subseteq A(f, k)$ by line 6 in the algorithm. Let (g_2, k_2) be some other child of (f, k) in C . Since $(f, k) \notin V(H_{k-1})$, both $(g_1, k_1), (g_2, k_2)$ are the roots of their corresponding components in H_{k-1} . By the induction hypothesis, $A(g_1, k_1) \cap A(g_2, k_2)$ are internally disjoint. This proves that C is an interval tree.

Let C_1, C_2 be two components in H_k , rooted at $(f_1, k_1), (f_2, k_2)$ respectively. we claim that $A(f_1, k_1)$ and $A(f_2, k_2)$ are internally disjoint.

If both $k_1, k_2 < k$, then $A(f_1, k_1)$ and $A(f_2, k_2)$ are internally disjoint by the induction hypothesis. If $k_1 = k_2 = k$, then Lemma 8 implies that $A(f_1, k_1)$ and $A(f_2, k_2)$ are internally disjoint. Suppose $k_1 = k$ and $k_2 < k$. Since (f_2, k_2) is a root of a component in H_k , (f_2, k_2) must have been unassigned at the end of iteration k in loop 4 in the algorithm. Therefore, since $\sigma(f_2, k_2) \neq A(f_1, k_1)$, $(f_2, k_2) \notin S(f_1, k_1)$ and therefore $A(f_1, k_1) \cap A(f_2, k_2) = \emptyset$ by the definition of $S(f_1, k_1)$ in line 6. This proves the inductive statement.

Finally, since $A(f_0, l+1) = I(f_0, l+1) = \mathbb{R}$, by line 6, $H = H_{l+1}$ is a tree with root f_0 , and therefore an interval tree by our claim. \square

Lemma 10. *The tree H is an interval tree at the end of step 2 in Algorithm 6. Further, H satisfies the immediate parent condition (2).*

Proof. We first claim that H stays an interval tree at all times in step 2. By Lemma 9, each $A(f, k)$ is an interval at the start of step 2. The only updates to intervals A occur on line 14. Since convex hulls on \mathbb{R} are intervals, sets A are still intervals after step 2.

Fix an iteration of the loops in step 2, and let $(f, k), (h, k')$ and $(g, k-1)$ be the corresponding facilities in this iteration. Before the update in line 14, $A(g, k-1), A(h, k') \subseteq A(f, k)$, and therefore $\text{conv}(A(g, k-1), A(h, k')) \subseteq A(f, k)$. Therefore, after the update, $A(g, k-1)$ is still a subset of $A(f, k)$. Further, by construction, $A(h, k')$ is a subset of $A(g, k-1)$ after the update, so that interval tree condition (1a) is satisfied.

We now show that interval tree condition (1b) is also satisfied. Since H is an interval tree before the update, the intersection of $A(g, k-1)$ and $A(h, k')$ is empty before the update. Therefore, after the update, the subtree of H rooted at $(g, k-1)$ is an interval tree. Further, for any other child $(g', k-1)$ of (f, k) , $A(g', k-1), A(g, k-1)$ and $A(h, k')$ are disjoint sets before the update. Since $(g, k-1)$ is the child of (f, k) closest to (h, k') , this means that $A(g', k-1)$ and $\text{conv}(A(g, k-1) \cup A(h, k'))$ are disjoint before the update, or that $A(g', k-1)$ and $A(g, k-1)$ are disjoint after the update. Since the only children of (f, k) are of the form $(g', k-1)$ after the update, this implies that (1b) is also satisfied after the update.

Finally, we show that the immediate parent condition (2) is satisfied by H . We show the following stronger statement by induction on level k : after the k th iteration of the outer loop in step 2 (on line 9), the subtree of H induced by $\overline{G}_k \cup \overline{G}_{k+1} \cup \dots \cup \overline{G}_{l+1}$ satisfies the immediate parent condition. In iteration k , all edges of the form $((f, k), (h, k'))$, $k' < k-1$ are removed from H , and so the only children of (f, k) remaining are of the form $(g, k-1)$. Since so edges containing any vertex in $\overline{G}_{k+1} \cup \dots \cup \overline{G}_{l+1}$ are modified in iteration k , this implies the claim for level k using the induction hypothesis. \square

The following lemma follows by a simple induction on level k ; we omit its proof:

Lemma 11. *H satisfies the completeness condition (3) at the end of Algorithm 2.*

E Strong refinement for tree metric

Consider the case when the underlying metric is induced by a tree, i.e., there is a tree $T = (V, E, w)$ with vertices $V = D \cup F$, edges E , and nonnegative edge weights $w : E \rightarrow \mathbb{R}$ with the distance $d(j, j')$ defined as the sum of edge weights on the unique path from j to j' for all $j, j' \in V$. Under the assumptions that the facility costs are uniform (i.e., each facility incurs the same opening cost) and that a facility can be opened at any vertex on the tree, we give an $O(|P|)$ -approximate strong refinement for a finite norm set P (Theorem 12), and therefore an $O(\log r)$ -approximate strong refinement for all norms, i.e., norm set $P = [1, \infty]$ (Corollary 5) using the *representative norm set* defined in Section 2:

Theorem 12. *Given a finite set of norms P , there is a polynomial-time algorithm that gives an $O(|P|)$ -approximate strong refinement for P for the tree metric, assuming uniform facility costs and that a facility can be opened at any vertex of the tree.*

Corollary 5. *There is a polynomial-time algorithm that gives an $O(\log_2 r)$ -approximate strong refinement for $P = [1, \infty]$ for the tree metric, assuming uniform facility costs and that a facility can be opened at any vertex of the tree.*

Given a tree T , a vertex $v \in V(T)$ is called a *branch vertex* if $\deg_T(v) \geq 3$. Given a set of vertices $S \subseteq V(T)$, let T_S denote the subtree of T induced by all vertices in S and those in the paths connecting any two vertices in S (i.e., the maximal subtree of T with all leaf vertices in set S). Two vertices $u, v \in S$ are called *consecutive vertices of S* if the unique path in T (and in T_S) joining u, v does not contain any other vertex of S . If S contains all the branch vertices of T_S , then the edges of T_S can be uniquely partitioned into paths between pairs of consecutive vertices of S . We will also use the following observation that follows from the handshaking lemma:

Fact 1. The number of branch vertices is at most the number of leaves in any tree.

We now sketch our algorithm for getting a strong refinement on a tree $T = (V, E, w)$. Suppose we are given a finite norm set $P = \{p_1, \dots, p_l\}$ with $p_1 < \dots < p_l$. First, using Corollary 1, we get solutions $(F_k, \theta_k), k \in [l]$ that are 4-approximate for norms $p_k, k \in [l]$ respectively. As we have repeatedly done, we first get a weak refinement by opening facility sets $G_k = \bigcup_{t \in [k, l]} F_t$ for norm p_k , with the facility costs for these sets upper bounded by $C_{G_k} \leq 4l \text{OPT}_{p_k}$. However, unlike before, we now open more facilities: let G'_k be the set of all facilities in G_k and on the branch vertices of T_{G_k} , i.e.,

$$G'_k = G_k \cup \{v \in T : v \text{ is a branch vertex in } T_{G_k}\} \quad \forall k \in [l].$$

Notice that since $G_k \supseteq G_{k+1}$ for all k , $T_{G_{k+1}}$ is subgraph of T_{G_k} , and therefore $G'_k \supseteq G'_{k+1}$ for all k , i.e., facility sets G'_1, \dots, G'_l also form a weak refinement. Additionally, the cost of opening

Algorithm 6 BRANCHANDLINEARIZE(T, t, G_1, \dots, G_t) for strong refinement over trees

input: tree $T = (V(T), E(T), w)$, level $t \geq 0$, facility sets $G'_1, \dots, G'_t \subseteq V$ satisfying (a) $G'_1 \supseteq \dots \supseteq G'_t \neq \emptyset$, and (b) all branch vertices of $T_{G'_k}$ are in G'_k for all $k \in [t]$
output: Assignments Π_1, \dots, Π_t on $V(T)$ that form a strong refinement and satisfy $d(j, \Pi_k(j)) \leq 2t \min_{f \in G'_k} d(j, f)$ for all $j \in D$, $k \in [t]$

- 1: Initialize empty mappings $\Pi_1, \dots, \Pi_t = \emptyset$
- 2: **for** each path p between two consecutive vertices of G'_k in $T_{G'_k}$ **do**
- 3: Treat path p as a line with lengths determined by edge weights w , and obtain assignments

$$\Pi'_1, \dots, \Pi'_t \leftarrow \text{EXPANDINTERVALS}(G'_1, \dots, G'_t),$$
 where each assignment $\Pi'_k : V(p) \rightarrow V(p) \cap G'_k$, $k \in [t]$
- 4: Update assignments $\Pi_k \leftarrow \Pi_k \cup \Pi'_k$ for $k \in [t]$
- 5: **for** each component C of $T \setminus T_{G'_t}$ **do** \triangleright To assign remaining vertices; i.e., those in $T \setminus T_{G'_t}$
- 6: Let f be the vertex that connects C to $T_{G'_t}$
- 7: Let $s \in [t]$ be the least level such that C has no facility from G'_s , i.e., $V(C) \cap G'_s = \emptyset$
- 8: **for** $k \in [s, t]$ **do**
- 9: Assign $\Pi_k(j) = \Pi_k(f)$ for all $j \in V(C)$ \triangleright Assign j to the facility assigned to f
- 10: $\Pi'_1, \dots, \Pi'_{s-1} = \text{BRANCHANDLINEARIZE}(C \cup \{f\}, s-1, G'_1, \dots, G'_{s-1})$ \triangleright Recurse
- 11: Update $\Pi_k \leftarrow \Pi_k \cup \Pi'_k$ for all $k \in [s-1]$

return Assignments Π_1, \dots, Π_t

these additional facilities is not very high: by our observation above, $|G'_k| \leq 2|G_k|$; and so by our assumption of uniform facility costs, we get $C_{G'_k} \leq 2C_{G_k} \leq 8l \text{OPT}_{p_k}$.

Similar to Sections 3.1.2 and 3.2 (for arbitrary and line metrics respectively), it is now sufficient to give assignments $\Pi_k : V \rightarrow G'_k$ for $k \in [l]$ that (a) form a strong refinement, and (b) satisfy the cost guarantee that for each client $j \in V(T)$,

$$d(j, \Pi_k(j)) \leq O(l) \cdot \min_{f \in G'_k} d(j, f) \quad \forall k \in [l]. \quad (14)$$

To do this, notice that we can decompose edges in $T_{G_l} = T_{G'_l}$ into edge-disjoint paths between facilities in G'_l , and so we can call the strong partition algorithm for line, EXPANDINTERVALS from Section 3.2 on each of these paths. This will give a $2l$ -approximate strong refinement on T_{G_l} . It remains to give assignments on $T \setminus T_{G_l}$. Each component C of $T \setminus T_{G_l}$ is a tree, and connected to T_{G_l} by a unique vertex $f \in G_l$. Since there is no facility in G'_l in C , we can assign $\Pi_l(j) = \Pi_l(f)$ for all $j \in V(C)$. For a lower level $k \in [l-1]$, there are two cases. If $V(C) \cap G'_k$ is empty, then the natural choice again is to assign $\Pi_k(j) = \Pi_k(f)$ for all $j \in V(C)$. If $V(C) \cap G'_k$ is nonempty, then we can show that f must also be in G'_k , and we can recurse on C .

We input tree T , level l and facility sets $G'_1 \supseteq \dots \supseteq G'_l$ to Algorithm 6. We prove in Lemma 13 that the assignments $\Pi_k, k \in [l]$ output by the algorithm form a strong refinement, and in Lemma 14 that they satisfy the cost guarantee in Lemma (14), finishing the proof of Theorem 12. But first, we prove that they are well-defined:

Lemma 12. *Algorithm 6 produces well-defined assignments Π_1, \dots, Π_l , i.e., that each Π_k is a function from $V(T)$ to G'_k .*

Proof. We first show that each vertex is covered at least once (i.e., assignments are not partial): each vertex in $T_{G'_l}$ is covered by some path p in the loop at line 2. Each vertex in $T \setminus T_{G'_l}$ is covered in the loop at line 5. We next show by induction that when some assignment $\Pi_k(j)$ is defined multiple times for some $j \in V(T)$, then j is a facility in G'_k and therefore the algorithm always assigns $\Pi_k(j) = j$ (since EXPANDINTERVALS always assigns an open facility at a level to itself by Lemma 7). There are only the following two cases when $\Pi_k(j)$ is defined multiple times:

Case 1: j is the endpoint of multiple paths connecting consecutive vertices of G'_l ; $\Pi_k(j)$ is then defined multiple times in the loop at line 2. But each endpoint of such paths is in G'_l , so $j \in G'_k$ since $G'_l \subseteq G'_k$.

Case 2: Some component C of $T \setminus T_{G'_l}$ is connected to $T_{G'_l}$ at j . In this case, $\Pi_k(j)$ is first defined for some path(s) in the loop at line 2, and is defined again potentially again in line 10. If $j \in G'_l$, then $j \in G'_k$. Otherwise, $\deg_{T_{G'_l}}(j) = 2$ and so by the condition in line 7, $V(C) \cap G'_k \neq \emptyset$, and so j is a branch vertex in $T_{G'_k}$, implying that $j \in G'_k$. \square

Lemma 13. *Assignments $\Pi_k : V \rightarrow G'_k$, $k \in [l]$ output by Algorithm 6 form a strong refinement.*

Proof. Any vertex v in a tree induces two subtrees T_1, T_2 on either side that only intersect at v and $T_1 \cup T_2 = T$. It is sufficient to prove that the assignments Π_1, \dots, Π_l form a strong refinement when restricted to each of the two subtrees induced by v . By generalizing this argument, it is sufficient to show that these assignments form a strong refinement on each of the paths that decompose G'_l , and for each component of $T \setminus T_{G'_k}$.

We prove this using induction on level l . For $l = 1$, the claim is trivially true. Suppose $l > 1$, and that the result is true for all levels $k \leq l - 1$ on all trees.

If p is one of the paths that decompose $T_{G'_l}$, the assignments on p are defined in line 3 in the algorithm, using EXPANDINTERVALS. By the corresponding guarantee on the line metric (Theorem 6), Π_1, \dots, Π_l form a strong refinement when restricted to p .

Fix a component C of $T \setminus T_{G'_l}$, connected to $T_{G'_l}$ at f . Let s be the level chosen in line 7. Note that s exists because $V(C) \cap G'_l = \emptyset$ since C is a subtree of $T \setminus T_{G'_l}$. Assignments Π_s, \dots, Π_t are defined on C in line 9, and assignments Π_1, \dots, Π_{s-1} are defined on $C \cup \{f\}$ recursively in line 10.

By the induction hypothesis, Π_1, \dots, Π_{s-1} form a strong refinement on $C \cup \{f\}$. Therefore, it remains to show that for each $k \in [s - 1, t - 1]$ and each facility $g \in G'_k$, there is some facility $g' \in G'_{k+1}$ such that all clients in $V(C)$ assigned to g under Π_k are assigned to g' under Π_{k+1} . But since $k + 1 \in [s, t]$, all clients $j \in V(C) \cup \{f\}$ are assigned to $\Pi_{k+1}(f)$ under Π_{k+1} by line 9, and so the choice $g' = f$ works no matter what g is. \square

Lemma 14. *Assignments $\Pi_k : V \rightarrow G'_k$, $k \in [l]$ output by Algorithm 6 satisfy the cost guarantee in (14), i.e., for all $j \in V(T)$ and each $k \in [l]$,*

$$d(j, \Pi_k(j)) \leq 2l \cdot \min_{f \in G'_k} d(j, f) \leq 2l \cdot \min_{f \in G_k} d(j, f).$$

Proof. We use induction on level l . As noted above, we can decompose T into $T_{G'_l}$ and components of $T \setminus T_{G'_l}$, and $T_{G'_l}$ can be further decomposed into edge-disjoint paths between pairs of consecutive vertices in G'_l .

Suppose a client j lies on one such path p . The endpoints of p are both in G'_l , and therefore in G'_k for all $k \in [l]$. Therefore, for each client $j \in V(p)$, the nearest facility at each of the levels $k \in [l]$ lies on this path. Since $\Pi_1(j), \dots, \Pi_l(j)$ are assigned on line 3 using EXPANDINTERVALS, the result then holds for j by the corresponding claim for line metric (Lemma 7).

If j is not on such a path, let C be the component of $T \setminus T_{G_k}$ containing j , and let C be connected to T_{G_k} at vertex f . Let s be the number defined in line 7, i.e., s is the least level in $[l]$ such that $V(C) \cap G'_s = \emptyset$.

If $k < s$, then $V(C) \cap G'_k \neq \emptyset$, and so f is either in G'_l or is a branch vertex in G'_k and therefore itself in G'_k . In this case, the result holds by the induction hypothesis.

If $k \geq s$, then $V(C) \cap G'_k = \emptyset$, and therefore the facility in G'_k closest to j is the facility in G'_k closest to f ; call this facility g . Since f lies on some path connecting two consecutive vertices in G'_l , we have by our earlier claim that $d(f, \Pi_k(f)) \leq 2l \cdot d(f, g)$. Now, since $\Pi_k(j) = \Pi_k(f)$ by construction in line 9, we get $\frac{d(j, \Pi_k(j))}{d(j, g)} = \frac{d(j, \Pi_k(f))}{d(j, g)} = \frac{d(j, f) + d(f, \Pi_k(f))}{d(j, f) + d(f, g)} \leq \max\left(\frac{d(j, f)}{d(j, g)}, \frac{d(f, \Pi_k(f))}{d(f, g)}\right) \leq 2l$. \square