A Constant Factor Approximation Algorithm for Fault-Tolerant k-Median

Mohammadtaghi Hajiaghayi [*], Wei Hu [†‡], Jian Li [‡], Shi Li [§] and Barna Saha [¶]

October 9, 2013

Abstract

In this paper, we consider the fault-tolerant k-median problem and give the *first* constant factor approximation algorithm for it. In the fault-tolerant generalization of classical k-median problem, each client j needs to be assigned to at least $r_j \geq 1$ distinct open facilities. The service cost of j is the sum of its distances to the r_j facilities, and the k-median constraint restricts the number of open facilities to at most k. Previously, a constant factor was known only for the special case when all r_js are the same, and a logarithmic approximation ratio was known for the general case. In addition, we present the first polynomial time algorithm for the fault-tolerant k-median problem on a path or a HST by showing that the corresponding LP always has an integral optimal solution.

We also consider the fault-tolerant facility location problem, where the service cost of j can be a weighted sum of its distance to the r_j facilities. We give a simple constant factor approximation algorithm, generalizing several previous results which only work for nonincreasing weight vectors.

1 Introduction

The k-median problem is one of the central problems in approximation algorithms and operation research. The most basic version of the k-median problem is defined as follows. We are given a set of facilities $\mathcal{F}$ and a set of demands (or clients) $\mathcal{D}$ in a metric space. We can open at most k facilities, and then assign each client j to the opened facility that is closest to it. Assigning demand j to facility i incurs an assignment cost of $\mathrm{d}(i,j)$, where $\mathrm{d}(i,j)$ is the distance between i and j. Our goal is to choose at most k facilities so that the sum of the assignment costs is minimized. Lin and Vitter [35] gave a polynomial-time algorithm that, for any $\epsilon > 0$, finds a solution of cost no more than $2+\epsilon$ times the optimum, while using at most $(1+\epsilon)k$ facilities. The first non-trivial approximation algorithm that produces a feasible solution (i.e., open at most k facilities) achieves a logarithmic approximation ratio by combining the metric embedding results [6, 17] and the fact that k-median can be solved in polynomial time in a tree metric. Charikar, Guha, Tardos and Shmoys [11] gave the first constant factor approximation algorithm using LP rounding. This was improved by a series of papers [10, 22, 4, 12] and the current best approximation ratio is $1+\sqrt{3}+\epsilon$ for any $\epsilon > 0$ via pseudo approximation [33]. For the *fault tolerant* version of k-median (FTMed), each client j needs to be assigned to at least $r_j \geq 1$ distinct open facilities. The service cost of j is the sum of its distances to the r_j facilities. A special case of FTMed is when all the r_js are the same. We call such instance as *uniform* FTMed (denoted by Uni-FTMed). For Uni-FTMed, Swamy and Shmoys [41] developed a 4-approximation using the Lagrangian relaxation technique. However, their technique does not work when r_js are not same, even when r_js are either 1 or 2. For general FTMed, where r_js can be non-uniform, the best known result is a logarithmic factor approximation algorithm [2].

In the closely related uncapacitated facility location problem (UFL), there is a facility opening cost f_i for each facility i and our objective is to minimize the sum of the facility opening cost and the total assignment cost. The first constant factor approximation algorithm for UFL was given by Shmoys, Tardos and Aardal [39], using the filtering technique of Lin and Vitter [34]. Subsequently, a variety of techniques in approximation algorithms has been successfully applied to UFL (see e.g., [14, 24, 4, 3, 22, 15, 10, 32]). The current best approximation ratio is 1.488 by Li [32], which is quite close to the best known inapproximability bound of 1.463 due to Guha and Khuller [19]. In this pa-

[*]Department of Computer Science , University of Maryland at College Park, USA. Supported in part by NSF CAREER award 1053605, NSF grant CCF-1161626, ONR YIP award N000141110662, and DARPA/AFOSR grant FA9550-12-1-0423. Email: hajiaghag@cs.umd.edu

[†]Email: huwei9527@gmail.com.

[‡]Institute for Interdisciplinary Information Sciences (IIIS), Tsinghua University, China. Supported in part by the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00301 and the National Natural Science Foundation of China Grant 61202009, 61033001, 61061130540, 61073174. Corresponding author. Email: lijian83@mail.tsinghua.edu.cn

[§]Department of Computer Science, Princeton University. Supported by SPA-MCS 0528414, CCF 0832797, and AF 0916218, and Wallace Memorial Fellowship Email: shili@cs.princeton.edu

[¶]AT&T Research Laboratory, Florham Park, NJ 07932. Email: barna@research.att.com.

per, we study the *fault-tolerant* version of UFL where each client j needs to be assigned to at least $r_j \geq 1$ distinct open facilities. Client j is associated with a weight vector $\mathbf{w}_j = \{w_j^{(1)}, w_j^{(2)}, \ldots, w_j^{r_j}\}$. The service cost of j is the weighted sum of its distances to the r_j facilities, i.e., $\sum_i w_j^{(i)} \mathrm{d}(h_i, j)$ where h_i is the ith closest open facility. It models the situation where each client needs one or more "backup" facilities in case its closest facility fails. The fault-tolerant facility location (FTFL) is a generalization of UFL in which $r_j = 1$ for each client j. FTFL with nonincreasing weight vectors ($w_j^{(1)} \geq w_j^{(2)} \geq \ldots$ for each client j) has been studied extensively. Jain and Vazirani gave a primal-dual based algorithm achieving a logarithmic approximation factor [25]. The first constant factor approximation algorithm with a factor of 2.408 is due to Guha, Meyerson and Munagala [20]. This was later improved to 2.076 by Swamy and Shmoys [41] and 1.7245 by Byrka, Srinivasan and Swamy [7], which is currently the best known ratio. However, nothing is known for FTFL with general positive weight vectors. Measuring service cost using general weight vectors is often a natural choice. For example, in the fault-tolerant k-center problem [26, 13], the service cost of a client is chosen to be its distance to the rth closest facility (this corresponds to the weight vector ($w_j^{(1)} = 0, \ldots, w_j^{(r-1)} = 0, w_j^{(r)} = 1, w_j^{(r+1)} = 0, \ldots$)). Further consider the following application in a wireless sensor network. We need to place hotspots (facilities) to provide wireless services for a designated area. Each hotspot may fail independently with probability p at every time slot. Each client is a sensor that needs to communicate with one hotspot. To ensure that the communication succeeds with probability at least $1 - \delta$ at each time slot, the transmission radius (fixed all the time) of the client needs to be the distance from the client to its $\lceil \log_p \delta \rceil$th closest hotspot. If the communication cost of a client scales linearly with its transmission radius, the problem is exactly FTFL with weight vectors of the form $(0, \ldots, 0, 1, 0, \ldots)$.

1.1 Our Results Our main result is a constant factor approximation algorithm for general FTMed. The current best approximation algorithm for general FTMed achieves a logarithmic approximation ratio [2]. Note that no constant factor approximation algorithm is known even for the case where the demands are either 1 or 2 and no previous techniques for k-median or uniform FTMed [11, 4, 23, 12, 41] seem to be generalizable easily to this case. Our algorithm is built on solving the natural linear programming (LP) relaxation of FTMed. Rounding is involved and proceeds through stages. First, based on the LP solution, we classify the clients into *safe* and *dangerous*. The safe clients are those whose distance to the furthest fractional facility assigned to it can be bounded by a constant factor of the connection cost defined by the LP solution (for the precise definition, see Section 2). Handling such clients is easy and well understood in recent literature on the fault-tolerant facility location problem [41, 7, 43]. In fact, in the fault-tolerant facility location problem, by scaling up the facility variables by a constant factor, one can transform all clients to safe, making it easy to approximate. However, in FTMed, we can not scale the facility variables since scaling would violate the constraint that we can open at most k facilities.

Next, we apply the adaptive clustering algorithm in [43] to produce a family of disjoint sets of facilities that we call *bundles*. However in [43], one can select multiple copies of the same facility. In order to avoid that, we need to keep a new mapping. In the rounding step, we ensure that each bundle contains exactly 1 open facility by randomly selecting an open facility inside it (according to the probabilities suggested by the LP), and we can show that the expected connection cost of a safe client is bounded by a constant times its connection cost in the LP solution. On the other hand, handling the dangerous clients is significantly more challenging and requires new techniques.

We judiciously create a family $\{B_j\}$ of facility sets for each client j choosing from the fractionally open facilities serving j such that B_j is *almost laminar*, that is the two sets are either nearly disjoint, or one is almost contained in the other. This becomes technically challenging primarily for the fact that demands among the clients could be highly skewed. Once we have such a structure, further refinements through filtering and other manipulations, lead to a laminar family of sets of facilities that have the nice property of $y(B_j)$ being very close to r_j. Here $y(B_j)$ is the expected number of fractional facilities in B_j. In the randomized rounding step, in addition to guaranteeing every bundle contains exactly 1 facility, we can also guarantee that every set in the laminar family contains either $\lfloor y(B_j) \rfloor$ or $\lceil y(B_j) \rceil$ open facilities. Since $y(B_j)$ is close to r_j, the rounding procedure opens r_j facilities in B_j with high probability and this suffices to show a constant approximation for the expected service cost of j.

As our second result, we show there is a polynomial time algorithm that can exactly solve general FTMed in a line metric. Unlike for the ordinary k-median problem on a line, which can be easily solved in polynomial time by dynamic programming, it is unclear how to generalize the dynamic program to FTMed (either uniform or non-uniform). Our algorithm is in fact based on a linear program. We show that the LP always has an optimal solution that is integral. We rewrite the LP based on any (fractional) optimal solution and show the new LP matrix is totally unimodular. A similar argument can be used to show that the LP of general FTMed on a hierarchically well separated tree (HST) also has an integral optimal solution. This improves the result in [9] where they showed that the integrality gap of the k-median LP on HSTs is at most 2. [1]

We also consider the fault tolerant version of the facility location problem (FTFL) where the service cost of a client is a weighted sum of the distances to the closest open facility, the 2nd closest open facility and so on. Our main result for this problem is a simple constant factor approximation algorithm for FTFL with a general weight vector for each client. This generalizes several previous results [20, 41, 7], where the weight vectors are nonincreasing. For general weight vectors, the most commonly used ILP formulation does not hold since the optimal integral LP solution may not correspond to a feasible solution. To remedy this, we use an extension of the ILP formulation for facility location proposed by Kolen and Tamir [27]. However, one can easily construct an example where the LP relaxation for this formulation has an unbounded integrality gap (see Section 4). Our approach is based on formulating a strengthened LP relaxation for the problem by adding "knapsack cover constraints" [8, 5].

1.2 Other Related Work Facility location and k-median are central problems in approximation algorithms. Many variants and generalizations have been studied extensively in the literature, including capacitated facility location [37, 30, 40] and k-median [16], multilevel facility location [1], universal facility location [36, 31], matroid median [21, 28, 12], knapsack median [29, 12], just to name a few. A closely related problem is the fault-tolerant k-center problem which has also been studied and constant factor approximation algorithms are known for several of its variants [26, 13]. Recently, Yan and Chrobak studied the fault-tolerant facility placement problem which is almost the same as FTFL except that we can open more than one copy of a facility and they gave a constant factor approximation algorithm based on LP rounding [43].

2 Fault Tolerant k-Median

We use $\mathcal{I} = \left(k, F, C, d, \{r_j\}_{j\in C}\right)$ to denote a FTMed instance. In the instance, $k \geq 1$ is an integer, F is the set of facilities, C is the set of clients, d is a metric over $F \cup C$ and $r_j \in [k]$ is the requirement of j. The solution of $\mathcal{I}$ is a set S of k facilities from F and its cost is the sum, over all clients $j \in C$, of the total distance from j to its closest r_j facilities in S.

The following is the natural LP relaxation for the FTMed:

$$\text{(2.1)} \qquad \text{minimize} \quad \sum_{j\in C}\sum_{i\in F} d(j,i)x_{i,j}$$

$$\begin{aligned}\text{subject to}\quad & y_i - x_{i,j} \geq 0 && \forall i \in F,\ j \in C\\ & \sum_{i\in F} x_{i,j} = r_j && \forall j \in C\\ & \sum_{i\in F} y_i \leq k && \\ & x_{i,j}, y_i \in [0,1] && \forall i \in F,\ j \in C\end{aligned}$$

Throughout the paper, we let y denote the y-vector obtained by solving the above LP. For a subset $S \subseteq F$ of facilities, define the *volume* of S to be $y(S) := \sum_{i\in S} y_i$. W.l.o.g., we assume $y(F) = k$.

We can assume $y_i \leq 1$ and $x_{i,j} \in \{0, y_i\}$ by the following splitting operation. Consider a facility i and a client j such that $x_{ij} < y_i$. We replace i with two facilities i_1, i_2 and let $y_{i_1} = x_{i_1 j} = x_{ij}, y_{i_2} = y_i - x_{i,j}, x_{i_2 j} = 0$. Of course, when we make such clones of a facility, we can only open one of them.

Instead of using (y, x), we use $\left(\{y_i\}_{i\in F}, \{F_j\}_{j\in C}, g\right)$ to denote an LP solution, where $F_j \subseteq F$ and $y(F_j) = r_j$ for every $j \in C$, and g shall be defined later. In this solution, y_i indicates whether to open the facility i. We assume $0 < y_i \leq 1$ for every $i \in F$. Then $i \in F_j$ if and only if $x_{i,j} = y_i$. We also assume F_j contains the closest r_j volume of facilities to j. That is, for any $j \in C, i \in F_j, i' \notin F_j$, we have $d(j,i) \leq d(j,i')$. For some non-empty set $S \subseteq F$ with $y(S) \neq 0$, let

$$d_{\mathsf{av}}(j,S) = \frac{\sum_{i\in S} d(j,i) y_i}{y(S)}$$

be the average distance from j to S. Let $d_{\max}(j,S)$ be the maximum distance from j to any node in S, i.e., $\max_{i\in S} d(j,i)$.

Notice that we can always split a facility i into two facilities i' and i'' with $y_i = y_{i'} + y_{i''}$ arbitrarily (replace any $F_j \ni i$ with $F_j \setminus \{i\} \cup \{i', i''\}$) without changing the value of the LP solution [2]. This turns out to be convenient in the following scenario. Suppose we are given a sequence of facilities $(i_1, i_2, \cdots, i_m)$ such that $\sum_{s=1}^{m} y_{i_s} \geq r$. We are interested in the integer t such that $\sum_{s=1}^{t-1} y_{i_s} < r$ and $\sum_{s=1}^{t} y_{i_s} \geq r$. If $\sum_{s=1}^{t} y_{i_s} > r$, we can split i_t into two facilities i' and i'' with $y_{i'} = r - \sum_{s=1}^{t-1} y_{i_s}$ and $y_{i''} = \sum_{s=1}^{t} y_{i_s} - r$. By splitting, we assume we can always find the integer t such that $\sum_{s=1}^{t} y_{i_s}$ is exactly r. Let $j \in C$ be a client and S be a set of facilities such that $y(S) \geq r$. Sort the facilities of S according to their distances to j, from

[1] It is well known that k-median on trees can be solved in polynomial time by combinatorial methods (e.g., [42]).

[2] Note that we have split the facilities before to ensure $x_{i,j} \in \{0, y_i\}$. Now, if we split a facility y_i (into $y_{i'}$ and $y_{i''}$), we need to split $x_{i,j}$ accordingly for all j: If $x_{i,j} = 0$, let $x_{i',j} = x_{i'',j} = 0$; if $x_{i,j} = y_i$, let $x_{i',j} = y_{i'}$ and $x_{i'',j} = y_{i''}$. It is easy to see that we still have $x_{i,j} \in \{0, y_i\}$ for all i, j after the split.

the closest to the furthest. Let s (resp. t) be the integer such that the first s (resp. t) facilities in the order have volume exactly $r-1$ (resp. r). Then, S_r contains the p-th facility in the sequence for every p from $s+1$ to t. So $y(S')=1$. If y is an integral solution, S' would correspond to the r-th closest facility to j. Define $d^r_{\mathsf{av}}(j,S)=d_{\mathsf{av}}(j,S_r)$ and $d^r_{\max}(j,S)=d_{\max}(j,S_r)$ where S_r is the set defined above.

We observe some simple yet useful facts. Let $j \in C$ be a client and S be a set of facilities with $y(S)=r$ for some integer r. Then, we have that

1. $d^t_{\mathsf{av}}(j,S) \le d^t_{\max}(j,S) \quad \forall t \in [r]$,
2. $d^t_{\max}(j,S) \le d^{t+1}_{\mathsf{av}}(j,S) \quad \forall t \in [r-1]$,
3. $d_{\mathsf{av}}(j,S) = \frac{1}{r}\sum_{t=1}^{r} d^t_{\mathsf{av}}(j,S)$.

In fact, the second inequality holds because $\min_{i\in S_{t+1}}\{d(j,i)\} \le d_{\mathsf{av}}(j,S_{t+1}) \le \max_{i\in S_{t+1}}\{d(j,i)\}$ and $\max_{i\in S_t}\{d(j,i)\} \le \min_{i\in S_{t+1}}\{d(j,i)\}$. The third equation holds since $S_1,\dots,S_r$ is a partition of S.

For ease of notation, we omit the second parameter of d_{av} and $d_{\max}$ if it is F_j. That is, we let $d_{\mathsf{av}}(j) = d_{\mathsf{av}}(j,F_j), d_{\max}(j) = d_{\max}(j,F_j), d^r_{\mathsf{av}}(j) = d^r_{\mathsf{av}}(j,F_j)$ and $d^r_{\max}(j) = d^r_{\max}(j,F_j)$.

In several steps mentioned above, we may split one facility into several copies. In the rounding step, to avoid opening more than one copies for each facility, we need to keep a mapping g where $g(i)$ indicates the original facility co-located with i from which i is split. $g(i)=i$ if i itself is the original facility. Thus, $d(i,g(i))=0$. Keep in mind that we need to make sure in the rounding step that at most 1 facility is open in $g^{-1}(i) := \{i' \in F : g(i')=i\}$ for any $i \in F$.

The high level idea of our algorithm is as follows. We solve LP (2.1) to obtain a fractional solution $\left(\{y_i\}_{i\in F}, \{F_j\}_{j\in C}, g\right)$. Our goal is to output a random set $S \subseteq F$ of size k such that the expected connection cost of j is $O(r_j d_{\mathsf{av}}(j))$ for each client j. We first use the adaptive clustering algorithm of [43] to construct a family $\mathcal{U}$ of disjoint sets of volume 1. If we randomly open 1 facility for each set $U \in \mathcal{U}$, we can show that the expected connection cost of each client $j \in C$ is $O(1)r_j d_{\mathsf{av}}(j) + d_{\max}(j)$. This can handle the clients j with small $d_{\max}(j)/(r_j d_{\mathsf{av}}(j))$ (which we call *safe clients*).

The remaining task is to handle the dangerous clients, i.e., the clients with a large $d_{\max}(j)/d^{r_j}_{\mathsf{av}}(j)$ value (the exact definition will appear later). We first apply a filtering step to select a subset D' of dangerous clients. For each $j \in D'$, we create a set B'_j of facilities such that the set family $\mathcal{B} = \{B'_j : j \in D'\}$ is laminar. Using the laminar family $\mathcal{B}$, we design a process to output a random set S of facilities so that (1) at most 1 facility is open inside $g^{-1}(i)$ for any $i \in F$, (2) each facility i is open with probability exactly y_i; (3) exactly 1 facility in each $U \in \mathcal{U}$ is open and (4) we open either $\lfloor y(B'_j)\rfloor$ or $\lceil y(B'_j)\rceil$ facilities inside each $B'_j \in \mathcal{B}$. With these properties, we can prove the constant approximation for FTMed.

The remainder of this section is organized as follows. We show how to construct $\mathcal{U}$ and $\mathcal{B}$ respectively in Section 2.1 and 2.2. Then, we show how to round the fractional solution based on $\mathcal{U}$ and $\mathcal{B}$ in Section 2.3. Finally, we prove the constant approximation ratio in section 2.4.

2.1 Construction of the Family $\mathcal{U}$ Given a k-median instance defined by $k, F, C, d, \{r_j\}_{j\in C}$ and a fractional solution $(\{y_i\}_{i\in F}, \{F_j \subseteq F\}_{j\in C}, g)$ to the instance, the algorithm of [43] outputs a family $\mathcal{U}$ of disjoint sets of volume 1, which we call *bundles*, as well as a set $\{U_{j,t}\}_{t\in[r_j]}$ of r_j different bundles from $\mathcal{U}$ for each $j \in C$. The algorithm is described in Algorithm 1.

If some U is added to $\mathcal{U}$ at Line 7 of Algorithm 1, we say the *creator* of U is j. We can see that the bundles in $\mathcal{U}$ are mutually disjoint. Moreover, for any $j \in C$, the r_j bundles added to queue_j are all different, since every time we add a bundle U to the queue_j, we removed $\mathcal{U} \cap F'_j$ from F_j.

LEMMA 2.1. *For any client $j \in C$, for any $r \in [r_j]$, we have $d_{\mathsf{av}}(j,U_{j,r}) \le 2d^r_{\max}(j) + d^r_{\mathsf{av}}(j)$.*

Proof. We prove the following statement: when the length of queue_j is $r-1$, we have $d^1_{\mathsf{av}}(j,F'_j) \le d^r_{\mathsf{av}}(j)$ and $d^1_{\max}(j,F'_j) \le d^r_{\max}(j)$. Notice that we only remove facilities from F'_j if we added some set B to queue_j. Moreover, we remove at most 1 volume of facilities from F'_j. Thus, when the length of queue_j is $r-1$, we removed in total at most $r-1$ volume of facilities from F'_j. It is easy to see that in order to maximize $d^1_{\mathsf{av}}(j,F'_j)$ ($d^1_{\max}(j,F'_j)$, resp.), it is the best to remove from F'_j the $r-1$ volume of closest facilities of j, in which case we have $d^1_{\mathsf{av}}(j,F'_j) = d^r_{\mathsf{av}}(j)(d^1_{\max}(j,F'_j) = d^r_{\max}(j)$, resp.). Thus, we proved the statement.

Suppose now the length of queue_j is $r-1$. Clearly, the volume of F'_j is at least 1. Consider the next time when we selected this client j and the correspondent U at Line 4. We know $d_{\mathsf{av}}(j,U) \le d^r_{\mathsf{av}}(j)$ and $d_{\max}(j,U) \le d^r_{\max}(j)$. If there is a $U' \in \mathcal{U}$ such that $U' \cap U \ne \emptyset$, let j' be the creator of U'. Then, we have $d_{\mathsf{av}}(j',U') + d_{\max}(j',U') \le d_{\mathsf{av}}(j,U) + d_{\max}(j,U)$, since we selected j' and U' before we selected j and U. Thus, $d(j,j') \le d_{\max}(j,U) + d_{\max}(j',U')$ and

$$\begin{aligned} d_{\mathsf{av}}(j,U') &\le d(j,j') + d_{\mathsf{av}}(j',U') \\ &\le d_{\max}(j,U) + d_{\max}(j',U') + d_{\mathsf{av}}(j',U') \\ &\le 2d_{\max}(j,U) + d_{\mathsf{av}}(j,U), \end{aligned}$$

which is at most $2d^r_{\max}(j) + d^r_{\mathsf{av}}(j)$.

If such U' does not exist, we added U to $\mathcal{U}$ and queue_j at Line 7, we have $d_{\mathsf{av}}(j,U) \le d^r_{\mathsf{av}}(j)$.

Algorithm 1 Create bundles

Input: A FTMed instance $\mathcal{I} = \left(k, F, C, d, \{r_j\}_{j\in C}\right)$ and a fractional solution $\left(\{y_i\}_{i\in F}, \{F_j\}_{j\in C}, g\right)$ to $\mathcal{I}$

Output: A family $\mathcal{U}$ of disjoint bundles, and a set $\{U_{j,t}\}_{t\in[r_j]}$ of r_j different bundles from $\mathcal{U}$ for each $j \in C$

$\mathcal{U} \leftarrow \emptyset$, $F'_j \leftarrow F_j$ and $\mathsf{queue}_j \leftarrow \emptyset$ for every client $j \in C$;
While there exists a client j such that the length of queue_j is smaller than r_j
  Select such a client j with the minimum $d^1_{\mathsf{av}}(j, F'_j) + d^1_{\max}(j, F'_j)$;
  Let $U \subseteq F'_j$ be the 1 volume of facilities such that $d^1_{\mathsf{av}}(j, F'_j) = d_{\mathsf{av}}(j, U)$ and $d^1_{\max}(j, F'_j) = d_{\max}(j, U)$; $\triangleright$ one might clone facilities in obtaining the set U and g is updated suitably to reflect this.
  If there exists a bundle $U' \in \mathcal{U}$ such that $U' \cap U \neq \emptyset$
  then add U' to the queue_j and remove $U' \cap U$ from F'_j;
  else add U to $\mathcal{U}$, add U to queue_j, and remove U from F'_j;
return $\mathcal{U}$ and $\{U_{j,t}\}_{j\in C, t\in[r_j]}$, where $U_{j,t}$ is the t-th bundle in queue_j.

2.2 Construction of the laminar Family $\mathcal{B}$ We say a client $j \in C$ is *dangerous* if

$$d_{\max}(j) \geq 45 d^{r_j}_{\mathsf{av}}(j).$$

The rest of clients are *safe*. Let D denote the set of dangerous clients. In this section, we first apply a filtering phase to obtain a subset $D' \subseteq D$ of dangerous clients. Then, for each $j \in D'$ we select a set $B'_j \subseteq F_j$ of facilities so that $\mathcal{B} = \{B'_j : j \in D'\}$ form a laminar family.

Filtering: We say two distinct dangerous clients $j, j' \in D$ *conflict* if $r_j = r_{j'}$ and

$$d(j, j') \leq 6 \max\{d_{\mathsf{av}}(j), d_{\mathsf{av}}(j')\}.$$

In the filtering phase, we select a subset $D' \subseteq D$ of dangerous clients such that no two clients in D' conflict each other. Algorithm 2 describes the filtering process.

Algorithm 2 Filtering

$D' \leftarrow \emptyset$;
For $r \leftarrow 1$ to R **do**
  $J = \{j \in D : r_j = r\}$;
  While $J \neq \emptyset$ **do**
    Let j be the client in J with the minimum $d_{\mathsf{av}}(j)$;
    Let J' be the set of clients in J that conflict j;
    Let $J \leftarrow J \setminus J' \setminus \{j\}$ and $D' \leftarrow D' \cup \{j\}$;
return D'.

FACT 2.1. *If $j \in D \setminus D'$, then there must be a client $j' \in D'$ such that $r_{j'} = r_j$, $d_{\mathsf{av}}(j') \leq d_{\mathsf{av}}(j)$ and $d(j, j') \leq 6 d_{\mathsf{av}}(j)$.*

Building a laminar family for dangerous clients For any client $j \in D'$, let $B_j := \mathsf{Ball}(j, d_{\max}(j)/15)$, where $\mathsf{Ball}(j, L) = \{i \in F : d(i, j) \leq L\}$ is the set of facilities that are within a distance L from j. We notice that with the definition of B_j, if a copy of some facility i is in B_j (recall a facility may be split into several copies), all copies of i are in B_j. We first present a few properties of B_j, then show how to construct the laminar family $\mathcal{B}$. The following lemma shows that the volume of B_j is very close to r_j.

LEMMA 2.2. *For a client $j \in D$ with $r_j = r$, we have*

$$r - \frac{15 d^r_{\mathsf{av}}(j)}{d_{\max}(j)} \leq y(B_j) < r.$$

Proof. First, we notice that $d_{\max}(j)/15 \geq d^r_{\mathsf{av}}(j) \geq d^{r-1}_{\max}(j)$. Since j is dangerous, we can see that all clients in $F_j \setminus B_j$ contribute to $d^r_{\mathsf{av}}(j)$. Thus we have

$$d^r_{\mathsf{av}}(j) \geq y(F_j \setminus B_j) d_{\max}(j)/15,$$

which implies that $y(B_j) = r - y(F_j \setminus B_j) \geq r - \frac{d^r_{\mathsf{av}}(j)}{d_{\max}(j)/15}$.

The following corollary follows directly from the definition of dangerous clients and Lemma 2.2.

COROLLARY 2.1. *For a client $j \in D$ with $r_j = r$, we have that $y(B_j) \geq r - 15/45 = r - 1/3$.*

The following lemma shows that two distinct dangerous clients in D' are necessarily far way. A corollary of the lemma which is useful later is that B_j and $B_{j'}$ are disjoint.

LEMMA 2.3. *Let j and j' be two distinct clients in D' such that $r_j = r_{j'} = r$. Then*

$$d(j, j') \geq d_{\max}(j)/10 + d_{\max}(j')/10.$$

Proof. Assume otherwise. First, we can see that $F_j \neq F_{j'}$ (due to the filtering phase). Since j and j' have the same demand r, it is not possible that F_j is strictly contained in $F_{j'}$ or F_j is strictly contained in $F_{j'}$. Combining this fact with triangle inequalities, we can see that

$$|d_{\max}(j) - d_{\max}(j')| \leq d(j, j') < \frac{d_{\max}(j)}{10} + \frac{d_{\max}(j')}{10}.$$

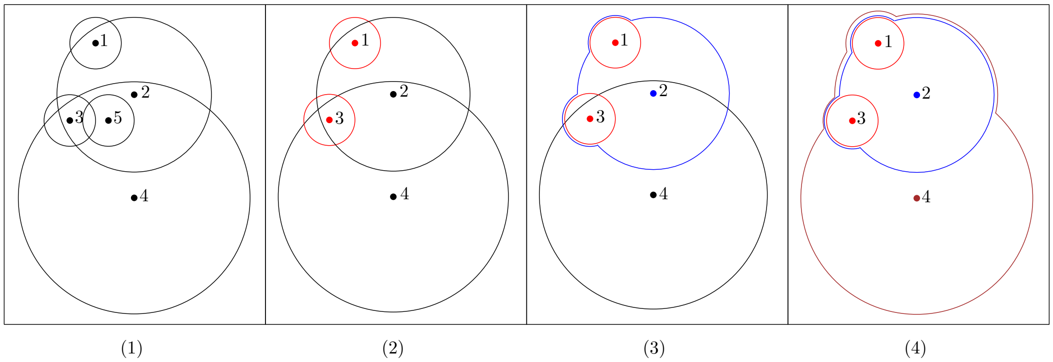

Figure 1: There are five facilities in the graph. (1): Before the filtering phase, 3 and 5 are in conflict and 5 is filtered out. (2)-(4): We build the laminar family in non-decreasing order of r_js.

Thus, we have that

$$\frac{d_{\max}(j')}{d_{\max}(j)} \in \left[\frac{1-1/10}{1+1/10}, \frac{1+1/10}{1-1/10}\right] = \left[\frac{9}{11}, \frac{11}{9}\right].$$

By triangle inequality, we can see that $B_{j'}$ is contained in $\mathsf{Ball}(j, d(j,j') + d_{\max}(j')/15)$ (this can be seen from the fact that every point in $B_{j'}$ is at most $d(j,j') + d_{\max}(j')/15$ distance away from j) and

$$\begin{aligned} d(j,j') &+ \frac{d_{\max}(j')}{15} \\ &\le \frac{1}{10}\left(1+\frac{11}{9}\right) d_{\max}(j) + \frac{11/9}{15} d_{\max}(j) \\ &< 0.5 d_{\max}(j), \end{aligned}$$

we have $B_{j'} \subseteq \mathsf{Ball}(j, 0.5d_{\max}(j))$. Thus, we have $B_j \cup B_{j'} \subseteq \mathsf{Ball}(j, 0.5d_{\max}(j))$, implying $y(B_j \cup B_{j'}) < r$, which, combined with Corollary 2.1, further implies that

$$y(B_j \cap B_{j'}) = y(B_j) + y(B_{j'}) - y(B_j \cup B_{j'}) \ge r - \frac{2}{3}.$$

Then, $d_{\mathsf{av}}(j, B_j \cap B_{j'}) \le r d_{\mathsf{av}}(j)/(r-2/3) \le 3d_{\mathsf{av}}(j)$. Similarly, $d_{\mathsf{av}}(j', B_j \cap B_{j'}) \le 3d_{\mathsf{av}}(j')$. By triangle inequality $d(j,j') \le 3(d_{\mathsf{av}}(j) + d_{\mathsf{av}}(j')) \le 6 \max\{d_{\mathsf{av}}(j), d_{\mathsf{av}}(j')\}$. j and j' can not be both in D' since they conflict each other, leading to a contradiction.

The following lemma shows that if two dangerous clients with different demands are close to each other, the ball for the client with the larger demand is necessarily much larger than the one for the other client.

LEMMA 2.4. *Let j and j' be two clients in D' with $r = r_j > r' = r_{j'}$. Suppose $d(j,j') \le d_{\max}(j)/15 + d_{\max}(j')/10$. Then $d_{\max}(j') \le \frac{1}{6} d_{\max}(j)$.*

Proof. Assume otherwise; then $d_{\max}(j) < 6d_{\max}(j')$. Then, we have that

$$\begin{aligned} d(j,j') + \frac{d_{\max}(j)}{15} &\le \frac{6d_{\max}(j')}{15} + \frac{d_{\max}(j')}{10} + \frac{6d_{\max}(j')}{15} \\ &= 0.9d_{\max}(j') \end{aligned}$$

and $B_j \subseteq \mathsf{Ball}\left(j', d(j,j') + \frac{d_{\max}(j)}{15}\right)$. Thus, we have $B_j \subseteq \mathsf{Ball}\left(j', 0.9d_{\max}(j')\right)$. Since $y(B_j) \ge r - 1/3 > r - 1 \ge r'$, we have $y(\mathsf{Ball}(j', 0.9d_{\max}(j'))) \ge r'$, contradicting the definition of $d_{\max}$.

In fact, if j and j' satisfy the condition of Lemma 2.4, we can see that the distance from every point in $B_{j'}$ to j is at most

$$\begin{aligned} d(j,j') &+ \frac{1}{15} d_{\max}(j') \\ &\le \frac{1}{15} d_{\max}(j) + \frac{1}{10} d_{\max}(j') + \frac{1}{15} d_{\max}(j') \\ &\le (\frac{1}{15} + \frac{1}{36}) d_{\max}(j). \end{aligned}$$

Intuitively, this suggests that $B_{j'}$ is *almost* contained in B_j. If the condition of Lemma 2.4 does not hold, j and j' are obviously disjoint. Therefore, we can see the family $\{B_j\}_{j\in D'}$ is *almost laminar*. In fact, by slightly modifying the sets B_j, we can form a laminar family.

Now, we present the algorithm for creating the laminar family $\mathcal{B}$. For any client $j \in D'$, we now construct a new set $B'_j \supseteq B_j$, which is B_j plus a small volume set of facilities. Algorithm 3 describes the process. See Figure 1 for an illustration of our algorithm. We prove that $\{B'_j\}_{j\in D'}$ forms a laminar family.

LEMMA 2.5. *The following properties hold for $\mathcal{B} = \{B'_j\}_{j\in D'}$:*

1. *$B'_j \subseteq \mathsf{Ball}(j, d_{\max}(j)/10)$ for every $j \in D'$;*

Algorithm 3 building a laminar family $\mathcal{B} = \{B'_j : j \in D'\}$ of sets

1: **For** $r = 1$ to R **do**
2: **For** each client $j \in D'$ such that $r_j = r$ **do**
3: Let D'' be the set of clients j' such that $r_{j'} < r$ and $B'_{j'} \cap B_j \neq \emptyset$;
4: $B'_j \leftarrow B_j \cup \bigcup_{j' \in D''} B'_{j'}$;

2. *$\mathcal{B} = \{B'_j\}_{j \in D'}$ forms a laminar family.*

Proof. We prove both the statements together by induction or r. We prove $B'_j \subseteq \mathsf{Ball}(j, d_{\max}(j)/10)$ for any client j such that $r_j \leq r$; also, the family $\mathcal{B}_r = \{B'_j\}_{j \in D' : r_j \leq r}$ form a laminar family. If $r = 1$, we have $B'_j = B_j = \mathsf{Ball}(j, d_{\max}(j)/15)$ for every $j \in D'$ with $r_j = 1$. Also, by Lemma 2.3, B'_j and $B'_{j'}$ are disjoint for two distinct clients j and j' in D' with $r_j = r_{j'} = 1$. Thus the statements are true for $r = 1$.

Suppose the statement is true for $r - 1$. Consider two clients j and j' in D' such that $r_j = r, r_{j'} < r$ and $B_j \cap B'_{j'} \neq \emptyset$. By the induction hypothesis, $B'_{j'} \subseteq \mathsf{Ball}(j', d_{\max}(j')/10)$, implying $d(j, j') \leq d_{\max}(j)/15 + d_{\max}(j')/10$. By Lemma 2.4, $d_{\max}(j') \leq \frac{1}{6} d_{\max}(j)$. Then, $d(j, j') + d_{\max}(j')/10 \leq d_{\max}(j)/15 + d_{\max}(j)/60 + d_{\max}(j')/60 = d_{\max}(j)/10$. Thus,

$$B'_{j'} \subseteq \mathsf{Ball}\Big(j, d(j, j') + \frac{d_{\max}(j')}{10}\Big) \subseteq \mathsf{Ball}\Big(j, \frac{d_{\max}(j)}{10}\Big).$$

This is true for any such client j'. By the definition of B'_j at Line 4, we have that

$$B'_j \subseteq \mathsf{Ball}(j, d_{\max}(j)/10).$$

Consider two distinct clients $j, j' \in D'$ such that $r_j = r_{j'} = r$. We claim that there is no j'' such that $r_{j''} < r$ and $B'_{j''}$ intersect both B_j and $B_{j'}$. Assume there is such a client j''. Then, we have that

$$d(j, j'') \leq \frac{d_{\max}(j)}{15} + \frac{d_{\max}(j'')}{10} \leq \frac{d_{\max}(j)}{12}.$$

Similarly $d(j', j'') \leq d_{\max}(j')/12$. Thus, $d(j, j') \leq d_{\max}(j)/12 + d_{\max}(j')/12$. Contradicting Lemma 2.3.

Notice that in order to construct B'_j at Line 4, it is enough to consider the sets in $\mathcal{B}_{r-1} = \{B'_{j''} \mid j'' \in D', r_{j''} \leq r - 1\}$ that are inclusively maximal (those that are not properly contained by other set in $\mathcal{B}_{r-1}$). By the induction hypothesis, these inclusively maximal sets are disjoint. Thus, for any clients $j, j' \in D'$ with $r_j = r_{j'} = r$, B'_j and $B'_{j'}$ are disjoint. Moreover, for any $j'' \in D'$ with $r_{j''} < r$, either $B'_{j''} \subseteq B'_j$ or $B'_{j''} \cap B'_j = \emptyset$. Thus, the family $\mathcal{B}_r = \{B'_j : j \in D', r_j \leq r\}$ is laminar.

2.3 Rounding After obtaining a LP solution $(\{y_i : i \in F\}, \{F_j : j \in C\})$, we run the algorithm of [43] as described in Section 2.1 to obtain a family $\mathcal{U}$ of disjoint bundles and the sets $\{U_{j,t} : j \in C, t \in [r_j]\}$. We then create the laminar family $\mathcal{B} = \{B'_j : j \in D'\}$ of sets. Notice that by Lemma 2.5, we have $\mathsf{Ball}(j, d_{\max}(j)/15) = B_j \subseteq B'_j \subseteq \mathsf{Ball}(j, d_{\max}(j)/10)$. Since j is dangerous, $y(\mathsf{Ball}(j, d_{\max}(j)/45)) \geq r - 1$. Thus, $r_j - 1 \leq y(B'_j) \leq r_j$. Consider the polytope defined by the following set of constraints. The set of variables is $\{z_i : i \in F\}$:

1. $\sum_{i \in U} z_i = 1 \quad \forall U \in \mathcal{U}$
2. $r_j - 1 \leq \sum_{i \in B'_j} z_i \leq r_j \quad \forall j \in D'$
3. $\sum_{i' \in g^{-1}(i)} z_{i'} \leq 1 \quad \forall i \in F$
4. $\sum_{i \in F} z_i = k$

From the construction of B'_j, it is easy to see that either $g^{-1}(i) \subseteq B'_j$ or $g^{-1}(i) \cap B'_j = \emptyset$ for any $i \in F$ and $j \in D'$. Thus, $\mathcal{B} \cup \{F\} \cup \{g^{-1}(i) : i \in F\}$ forms a laminar family. The constraints of the above polytope is defined by two laminar families of sets : $\mathcal{U}$ and $\mathcal{B} \cup \{F\} \cup \{g^{-1}(i) : i \in F\}$. It is well known that such a polytope defined by two laminar families is integral (the corresponding matrix is unimodular) (see e.g., [18]). Also, notice that the $z_i = y_i$ for every $i \in F$ is a feasible solution. Thus, we can express our vector y as a convex combination of vertices of the above polytope. Such a convex combination can be computed in polynomial time. Treating the coefficients in the convex combination as probabilities (note that the coefficients sum up to 1), we sample a random vertex. Due to the last constraint, the vertex contains exact k open facilities. Let S be the set of k facilities defined by the vertex. We summarize the useful properties of our rounding step as follows.

1. The probability that each facility $i \in F$ is open is exactly y_i;
2. For any $i \in F$, we open at most one facility inside $g^{-1}(i)$;
3. We open exactly 1 facility inside each $U \in \mathcal{U}$;
4. For each $j \in D'$, we open either $r_j - 1$ or r_j facilities in B'_j. Moreover, we have that

$$\Pr[r_j \text{ facilities are open in } B'_j] = y(B'_j) - (r_j - 1),$$
$$\Pr[r_j - 1 \text{ facilities are open in } B'_j] = r_j - y(B'_j).$$

2.4 Analysis We now have every piece ready to prove a constant factor approximation for FTMed. Each of the following lemmas deals with one type of clients. First, we consider safe clients.

LEMMA 2.6. *For any client $j \in C \setminus D$ with $r_j = r$, the expected connection cost of j is at most $93rd_{\mathsf{av}}(j)$.*

Proof. Notice that we always open 1 facility inside $U_{j,t}$ for every $t \in [r]$. We connect j to the r facilities in $\bigcup_{t\in[r]} U_{j,t}$. Connecting j to the facility in $U_{j,t}$ costs at most $2d^t_{\max}(j) + d^t_{\mathsf{av}}(j)$ in expectation, by Lemma 2.1. Thus, the expected connection cost of j is at most

$$\begin{aligned}
&\sum_{t=1}^{r} \left(2d^t_{\max}(j) + d^t_{\mathsf{av}}(j)\right) \\
&\qquad \le 2\sum_{t=1}^{r-1} d^{t+1}_{\mathsf{av}}(j) + 2d_{\max}(j) + \sum_{t=1}^{r} d^t_{\mathsf{av}}(j) \\
&\qquad \le 3rd_{\mathsf{av}}(j) + 2d_{\max}(j) \\
&\qquad \le 3rd_{\mathsf{av}}(j) + 2 \times 45 d^r_{\mathsf{av}}(j) \le 93rd_{\mathsf{av}}(j),
\end{aligned}$$

where the first inequality used the fact that $d^t_{\max}(j) \le d^{t+1}_{\mathsf{av}}(j)$ and the third inequality holds because j is a safe client.

LEMMA 2.7. *For any client $j \in D'$ with $r_j = r$, the expected connection cost of j is at most $46rd_{\mathsf{av}}(j)$.*

Proof. Notice that by Lemma 2.1, the distance from j to its r-th closest open facility is always at most $3d_{\max}(j)$. We can bound the expected connection cost of j as follows. If there are r_j open facilities inside B'_j, we connect j to the r open facilities; otherwise (they are $r-1$ open facilities), we connect j to the $r-1$ open facilities in B'_j and a r-th open facility outside B'_j whose distance to j can be bounded by $3d_{\max}(j)$. Thus, the expected connection cost of j is at most

$$\begin{aligned}
&\sum_{i\in B'_j} d(j,i)y_i + \Pr[r_j - 1 \text{ facilities are open in } B'_j] \cdot 3d_{\max}(j) \\
&\le rd_{\mathsf{av}}(j) + 3(r - y(B_j))d_{\max}(j) \\
&\le rd_{\mathsf{av}}(j) + 3 \times 15 d^r_{\mathsf{av}}(j) \le 46rd_{\mathsf{av}}(j),
\end{aligned}$$

where the second inequality follows from Lemma 2.2.

LEMMA 2.8. *For any client $j \in D \setminus D'$ with $r_j = r$, the expected connection cost of j is at most $52rd_{\mathsf{av}}(j)$.*

Proof. There is a $j' \in D'$ such that $r_j = r_{j'} = r, d_{\mathsf{av}}(j') \le d_{\mathsf{av}}(j)$ and $d(j,j') \le 6d_{\mathsf{av}}(j)$. By Lemma 2.7, the expected connection cost of j' is at most $46rd_{\mathsf{av}}(j')$. By triangle inequality, the expected connection cost of j is at most $46rd_{\mathsf{av}}(j') + rd(j,j') \le 46rd_{\mathsf{av}}(j) + 6rd_{\mathsf{av}}(j) = 52rd_{\mathsf{av}}(j)$.

Combining Lemma 2.6, 2.7 and 2.8, the expected connection cost of any client $j \in C$ is at most $93rd_{\mathsf{av}}(j)$, leading to a 93-approximation for FTMed.

3 FTMed on Paths and HSTs

We first consider the case where all the facilities and clients are on a line.

THEOREM 3.1. *For the non-uniform* FTMed *on a line metric, the problem can be solved exactly in polynomial time.*

In fact, all we need is to show the linear program (2.1) has an integral optimal solution. Unlike in the usual case, we can not show that the polytope defined by the LP constraints is integral. In fact, the polytope is the same as that for the general NP-hard k-median problem, thus not integral. The integral optimum is due to the specialty of the cost coefficients, i.e., $\mathrm{d}(i,j)$.

LEMMA 3.1. *If $\mathrm{d}(i,j)$s are defined by a line metric, the linear program* (2.1) *always has an integer optimal solution.*

Proof. We show for any fractional optimal solution $(x_{i,j}, y_i)$, we can construct an integral solution with the same cost. By the splitting trick [3], we can assume that $x_{i,j} = \{0, y_i\}$. Each client (fractionally) connects to a consecutive segment of facilities. Suppose i is needed by demands set J.

Now we can write another linear program without $x_{i,j}$ variables as follows. We use i' for indexing the facilities after the split and i for original facility. We write $i' \in \mathsf{sp}(i)$ to indicate that the new facility i' is derived from the original facility i. Let F_j be the set of facilities serving j (after the splitting process). The facilities in F_j form a consecutive segment in the path.

$$\begin{aligned}
(3.2) \qquad \text{minimize} \qquad & \sum_j \sum_{i'\in F_j} \mathrm{d}(i',j) y_{i'} \\
\text{subject to} \qquad & \sum_{i'\in F_j} y_{i'} \ge r, \quad \forall j \in C \\
& \sum_{i'\in \mathsf{sp}(i)} y_{i'} \le 1, \quad \forall i \in F \\
& \sum_{i'\in F} y_{i'} \le k, \quad \forall i \in F
\end{aligned}$$

It is easy to see that the optimal solution for the new LP is no more than that for the original LP. The constraint matrix of the new LP has the consecutive "one"s property: in each row of the constraint matrix, the "1"s appear in consecutive positions. Such matrices are known to be totally unimodular and the corresponding linear program has an integral optimal

[3] Consider facility i. Let J_l be the set of clients on the left side of i and J_r the set of clients on the right side. Consider the numbers $\{x_{i,j}\}_{j\in J_1} \cup \{y_i - x_{i,j}\}_{j\in J_2}$. These numbers split the interval $[0, y_i]$ into several pieces, and for each piece, we create a facility with fractional value equal to the length of that piece.

solution. (See e.g.,[38]). Furthermore, it is easy to see any integral feasible solution of (3.2) corresponds to a feasible solution for FTMed with the same cost. Therefore, the optimal integral solution of (3.2) has to be the same as that of (2.1). The above argument also gives us an algorithm to construct an integral solution of (2.1) of the optimal cost.

Using the same idea, we can get a polynomial time algorithm on an HST metric where all facilities and clients are located at leaves. We recall an HST (hierarchically well separated tree) is a tree where on any root to leaf path, the edge lengths decrease by some fixed factor in each step.

LEMMA 3.2. *The general* FTMed *problem can be solved exactly in polynomial time on an HST metric where all facilities and clients are located at leaves.*

Proof. We use $LCA(j_1, j_2)$ to denote the least common ancestor of leaves j_1 and j_2. Suppose the leaves of the HST are ordered according the preorder traversal. Consider a client j and suppose the path from j to the root is $\{j, p_1, p_2, \ldots, r\}$. In a fractional optimal solution $(x_{i,j}, y_i)$ of (2.1), client j chooses to connect all the facilities in the subtree rooted at p_1, then those at p_2, and so on. For any leaves j_1, j_2, j_3, if $LCA(j_1, j_2) = LCA(j_1, j_3)$, we can easily see that $\mathrm{d}_T(j_1, j_2) = \mathrm{d}_T(j_1, j_3)$. Therefore, we can assume j connects to a consecutive segment of facilities (in the preorder sequence of the facilities). Using almost the same argument as in Lemma 3.1, we can show that the LP has an integral solution with the optimal value.

Note that combining this result with classic tree embedding result [6, 17], we can easily get a simple $O(\log n)$-approximation for general FTMed on any metric. Since we have already shown a constant approximation for general FTMed, we omit the details.

4 Fault Tolerant Facility Location

For FTFL problem with arbitrary weights, we have a set F of n facilities and a set C of m clients. In the following sections, the terms "demand" and "client" are used interchangeably. For each client j, there is a nonnegative weight vector $\mathbf{w}_j = \{w_j^{(1)}, \ldots, w_j^{(r_j)}\}$ for some $r_j \le n$. Assume that the set of open facilities are $i_1, i_2, \ldots, i_h$ for some $1 \le h \le n$, sorted according to the nondecreasing order of their distance to j. The service cost of client j is $\sum_{t=1}^{r_j} w_j^{(t)} \mathrm{d}(i_t, j)$. If $h < r_j$, the service cost of j is infinity.

We focus on a special case of the above problem where only one entry of the vector $\mathbf{w}_j$ is nonzero. For ease of notation, we use r_j to denote the index of the nonzero coordinate in $\mathbf{w}_j$ and w_j to denote $w_j^{(r_j)}$, i.e., $w_j^{(r_j)} > 0$ and $w_j^{(t)} = 0$ for any $t \neq r_j$. Indeed, considering this special case is without loss of generality since we can create multiple copies for each demand node j, with the 1st copy associated with the weight vector $\{w_j^{(1)}, 0, \ldots, 0\}$, the 2nd copy $\{0, w_j^{(2)}, \ldots, 0\}$ and so on. It is straightforward to establish the equivalence and we omit the proof here. From now on, we use FTFL to denote this special case of the fault tolerant facility location problem. Our main result is a constant factor approximation algorithm for FTFL.

First, we note that the most natural linear integer programming formulation that was used for nonincreasing weight vectors in previous work does not work any more.

Hence, we use a different linear integer programming formulation as follows. We use boolean variable y_i to denote whether facility i is open, x_{ij} to denote whether demand j is assigned to facility i. We use $\pi(j,t)$ to denote the tth facility closest to j. Let $N(j,t) = \{\pi(j,1), \pi(j,2), \ldots, \pi(j,t)\}$ and $c_{jt} = \mathrm{d}(j, \pi(j,t))$. Let $c_{j0} = 0$ for all j. We use indicator variable z_{jt} to denote the event whether demand j is satisfied by $N(j,t)$ (i.e., at least r_j facilities among $N(j,t)$ are opened).

$$\text{(4.3)} \quad \min. \sum_i f_i y_i + \sum_j w_j \sum_{t \ge 0} (1 - z_{jt})(c_{j(t+1)} - c_{jt})$$

$$\text{(4.4)} \quad \text{s.t.} \quad \sum_i x_{ij} \ge r_j, \qquad \forall j \in C$$

$$\text{(4.5)} \quad y_i \ge x_{ij}, \qquad \forall i, j \in C$$

$$\text{(4.6)} \quad \sum_{i \in N(j,t)} x_{ij} \ge r_j z_{jt} \qquad \forall j \in C, \forall t \in [n]$$

$$\text{(4.7)} \quad y_i, x_{ij}, z_{jt} \in \{0,1\}, \forall i \in F, j \in C, t \in [n] \cup \{0\}$$

First, we need to explain our objective function since it is not the most frequently used objective for facility location. It is easy to see that a feasible solution of FTFL satisfies the IP formulation. For any optimal solution of the IP, if $N(j,t)$ satisfies j, $N(j,t')$ also satisfies j for $t' \ge t$. Therefore, $z_{jt} \ge z_{j(t-1)}$ for all t. If t' is the smallest t such that $z_{jt} = 1$, we can see that $w_j \sum_{t \ge 0}(1 - z_{jt})(c_{j(t+1)} - c_{jt})$ is equal to $w_j c_{jt'}$, which is exactly the service cost of j. We set $c_{j(n+1)} = \infty$. Constraints 4.4 specify that client j must be connected to r_j facilities. Constraints 4.5 ensure that a client is connected only to open facilities and constraints 4.6 imply that if $z_{jt} = 1$ then at least r_j facilities must be open in $N(j,t)$. The LP relaxation is obtained by replacing last constraints by $y_i, x_{ij}, z_{jt} \in [0,1]$.

However, we can not use the above LP directly to get a constant factor approximation algorithm since its integrality gap is large and can be as large as $\Omega(n)$. Consider the following FTFL instance in a line metric. There are n facilities and only one client. All facilities have cost zero and the client have demand n (i.e., $r_1 = n$). The x-coordinate of the client is 0. The x-coordinate of the ith facility is 0 for all $1 \le i \le n-1$ and the x-coordinate of the nth facility is n. The optimal integral solution opens all facilities and the service cost is n. A feasible fractional solution opens all facilities too. However, z_{jt} can take fractional values $\frac{1}{n}\sum_{i\in N(j,t)} x_{ij} = \frac{t}{n}$. The fractional service cost of the client is $\frac{n-1}{n}\cdot 0 + \ldots + \frac{2}{n}\cdot 0 + \frac{1}{n}\cdot n = 1$. Therefore, we obtain an integrality gap of $\Omega(n)$.

To strengthen the LP relaxation, we use the following *knapsack cover constraints* to replace constraints (4.6):

$$\sum_{i\in N(j,t)\setminus A} x_{ij} \geq (r_j - |A|)z_{jt}, \forall j\in C, t\in[n], A\subseteq N(j,t) \tag{4.8}$$

The constraints require that if $z_{jt} = 1$, then for every subset A, at least $r_j - |A|$ facilities from the set $N(j,t)\setminus A$ must be chosen to serve j. We can also see that there is a polynomial time separation oracle for (4.8): Suppose (x_{ij}, z_{jt}) is a solution. For fixed t and j, we can test the feasibility of (4.8) for all A with $|A| = k$ by checking whether the sum of the smallest $|N(j,t)|-k$ terms in $N(j,t)$ is at least $(r_j-k)z_{jt}$. Therefore, the relaxation can be solved optimally in polynomial time by the ellipsoid algorithm. Let (x^*, y^*, z^*) be the optimal fractional solution of the linear program and OPT be the optimal value.

Now, we round the fractional solution (x^*, y^*, z^*) to an integral solution $(\widehat{x}, \widehat{y}, \widehat{z})$ as follows. Let us consider a particular demand j. Let $\alpha < 1$ be a constant fixed later. Let t_j^* be the smallest integer t such that $z_{jt}^* \geq \alpha$.

LEMMA 4.1. *For every j, it holds that*

$$c_{jt_j^*} \leq \frac{1}{1-\alpha}\sum_{t=0}^{n-1}(1-z_{jt}^*)(c_{j(t+1)} - c_{jt}).$$

Proof. We can easily see that

$$\begin{aligned}&\frac{1}{1-\alpha}\sum_{t=0}^{n-1}(1-z_{jt}^*)(c_{j(t+1)} - c_{jt})\\ &\geq \frac{1}{1-\alpha}\sum_{t=0}^{t_j^*-1}(1-\alpha)(c_{j(t+1)} - c_{jt}) = c_{jt_j^*}.\end{aligned}$$

The first inequality follows because $z_{jt}^* \geq z_{j(t-1)}^*$ for all t. This is true because if we set $z_{j,t}^* \leftarrow \max\{z_{j,1}^*, ..., z_{j,t}^*\}$, it yields a feasible solution of no greater cost.

Now, we create a set of $\widetilde{y}_i$ values that we will round, based on the y_i^* values, as follows.

1. For all facility i with $y_i^* \geq \alpha$, we round it up to 1, i.e., $\widetilde{y}_i = 1$.
2. For all facility i with $y_i^* < \alpha$, we let $\widetilde{y}_i = \frac{1}{\alpha}y_i^*$.

LEMMA 4.2. *For each client j, $\sum_{i\in N(j,t_j^*)} \widetilde{y}_i \geq r_j$.*

Proof. Consider a particular client j. Let A be the set of facility i such that $x_{ij}^* \geq \alpha$ and $i \in N(j, t_j^*)$. From (4.8), we know that

$$\begin{aligned}\sum_{i\in N(j,t_j^*)\setminus A} y_i^* \;&\geq \sum_{i\in N(j,t_j^*)\setminus A} x_{ij}^*\\ &\geq\; z_{jt_j^*}^*(r_j - |A|) \;\geq\; \alpha(r_j - |A|).\end{aligned}$$

Therefore, we can see that

$$\sum_{i\in N(j,t_j^*)\setminus A} \widetilde{y}_i \;\geq \sum_{i\in N(j,t_j^*)\setminus A}\frac{1}{\alpha}y_i^* \;\geq\; r_j - |A|.$$

For each facility $i \in A$, we have $\widetilde{y}_i = 1$. Hence, $\sum_{i\in N(j,t_j^*)} \widetilde{y}_i \geq r_j$, which completes the proof.

Now, we round the $\widetilde{y}$ values to integers. Our rounding scheme is a slight variant of the one in [41]. Let $F_j = N(j, t_j^*)$. Let r_j' be the residual requirement of j, which is initially set to be r_j. We iterate the following steps until no client remains in the graph.

S1. We pick the client j with the minimum $c_{jt_j^*}$.

S2. Let $M \subseteq F_j$ be the set of the cheapest facilities in F_j (w.r.t. facility opening costs) such that $\sum_{i\in M}\widetilde{y}_i \geq r_j'$. If $\sum_{i\in M}\widetilde{y}_i$ is strictly larger than r_j', we replace the last facility, say facility i, by two "clones" i_1 and i_2. Set $\widetilde{y}_{i_1} = r_j' - \sum_{i\in M\setminus\{i\}}\widetilde{y}_i$ and $\widetilde{y}_{i_2} = \widetilde{y}_i - \widetilde{y}_{i_1}$. Include i_1 in M. Hence, $\sum_{i=M} y_i = r_j'$.

S3. Open the r_j' cheapest facilities in M. For each client k with $F_k \cap M \neq \emptyset$, we use any $\min(r_k', r_j')$ of the facilities we just opened to serve k and let $r_k' = r_k' - \min(r_k', r_j')$. Delete facilities in M and all clients with zero residual requirement from the input.

LEMMA 4.3. *The above rounding scheme returns a feasible solution. Moreover, the following properties hold.*

1. *The facility opening cost is at most $\sum_i f_i\widetilde{y}_i$.*
2. *For each client j, at least r_j facilities in $B(j, 3c_{jt_j^*})$ are open.*

Proof. The proof is almost the same as the one in [41]. For completeness, we include it here. Consider a particular iteration. It is easy to see the invariant $\sum_{i\in F_j}\widetilde{y}_i \geq r_j'$ is maintained throughout the three steps. So it is always possible to choose the set M. We also need to argue that no facility is opened twice since we have made some clones. We argue that whenever a facility i is replaced by two clones, the first clone never gets opened: This is simply because i is the most expensive facility in M and there are at least r_j' facilities cheaper than i (otherwise, we do not have to make clones).

To bound the facility cost, just notice that the cost of open facilities in M is less than $\sum_{i \in M} f_i \widetilde{y}_i$. This proves (1). To bound the connection cost, consider a particular client j. Any opened facility in F_j is at most $c_{jt_j^*}$ distance away from j. Notice that j may be served by some facilities in F_k for some other client k. This only happens if $F_j \cap F_k \neq \emptyset$ and $c_{kt_k^*} \leq c_{jt_j^*}$ (we process client k first). A facility in F_k is at most $2c_{kt_k^*} + c_{jt_j^*} \leq 3c_{jt_j^*}$ away from j.

From Lemma 4.3, we know that the first r_j copies of client j are assigned within a distance of $3c_{jt_j^*}$. Therefore, we have that the total cost of this integral solution

$$\mathsf{SOL} \leq \frac{1}{\alpha}\sum_i f_i y_i^* + 3\sum_j w_j c_{j(t_j^*)} \leq \frac{1}{\alpha}\sum_i f_i y_i^* + \frac{3}{1-\alpha}\sum_j w_j \sum_t (1 - z_{jt}^*)(c_{j(t+1)} - c_{jt}),$$

where the second inequality holds because of Lemma 4.1.

Setting $\alpha = \frac{1}{4}$ gives us an approximation ratio of 4. We can choose a random α to improve the approximation ratio as in [39, 20]. Let $L_j(\alpha)$ be c_{jt} for the minimal t such that $z_{jt} > \alpha$. It is easy to see the following.

LEMMA 4.4.

$$\int_0^1 L_j(\alpha)\mathsf{d}\alpha = \sum_t (1 - z_{jt}^*)(c_{j(t+1)} - c_{jt}).$$

Choose a random α uniformly distributed over $[h, 1]$. Then, the expected cost is

$$\begin{aligned}\mathbb{E}[\mathsf{SOL}] \;&\leq\; \int_h^1 \frac{1}{1-h}\Big(\frac{1}{\alpha}\sum_i f_i y_i^* + 3\sum_j w_j L_j(\alpha)\Big)\mathsf{d}\alpha \\ &\leq \frac{1}{1-h}\ln\frac{1}{h}\sum_i f_i y_i^* \\ &\quad + \frac{3}{1-h}\sum_j w_j \sum_t (1 - z_{jt}^*)(c_{j(t+1)} - c_{jt}).\end{aligned}$$

The above expression is minimized at $h = e^{-3}$, which gives an approximation ratio 3.16.

THEOREM 4.1. *There is a polynomial time approximation approximation with an approximation factor* 3.16 *for* FTFL.

Acknowledgement: We would like to thank anonymous reviewers for many useful comments which helped to improve the presentation of the paper.

References

[1] 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(5-6):161–167, 1999.

[2] B. M. Anthony, V. Goyal, A. Gupta, and V. Nagarajan. A plant location guide for the unsure. In *Proceedings of the 19th annual ACM-SIAM symposium on Discrete algorithms*, pages 1164–1173. SIAM, 2008.

[3] A. Archer, R. Rajagopalan, and D. B. Shmoys. Lagrangian relaxation for the k-median problem: new insights and continuity properties. In *In Proceedings of the 11th Annual European Symposium on Algorithms*, pages 31–42. Springer, 2003.

[4] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit. Local search heuristic for k-median and facility location problems. In *Proceedings of the 33rd annual ACM symposium on Theory of computing*, pages 21–29. ACM, 2001.

[5] N. Bansal, A. Gupta, and R. Krishnaswamy. A constant factor approximation algorithm for generalized min-sum set cover. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1539–1545. SIAM, 2010.

[6] Y. Bartal. On approximating arbitrary metrices by tree metrics. In *Proceedings of the 30th annual ACM symposium on Theory of computing*, pages 161–168. ACM, 1998.

[7] J. Byrka, A. Srinivasan, and C. Swamy. Fault-tolerant facility location: a randomized dependent lp-rounding algorithm. *Proceedings of the 14th Conference on Integer Programming and Combinatorial Optimization*, pages 244–257, 2010.

[8] R. D. Carr, L. K. Fleischer, V. J. Leung, and C. A. Phillips. Strengthening integrality gaps for capacitated network design and covering problems. In *Proceedings of the 11th annual ACM-SIAM symposium on Discrete algorithms*, pages 106–115. SIAM, 2000.

[9] M. Charikar, C. Chekuri, A. Goel, and S. Guha. Rounding via trees: deterministic approximation algorithms for group steiner trees and k-median. In *Proceedings of the 30th annual ACM symposium on Theory of computing*, pages 114–123. ACM, 1998.

[10] M. Charikar and S. Guha. Improved combinatorial algorithms for facility location problems. *SIAM Journal on Computing*, 34(4):803–824, 2005.

[11] M. Charikar, S. Guha, É. Tardos, and D.B. Shmoys. A constant-factor approximation algorithm for the k-median problem. *Journal of Computer and System Sciences*, 65(1):129–149, 2002.

[12] M. Charkar and S. Li. A dependent LP-rounding approach for the k-median problem. *The 39th International Colloquium on Automata, Languages and Programming*, 2012.

[13] S. Chaudhuri, N. Garg, and R. Ravi. The p-neighbor k-center problem. *Information Processing Letters*, 65(3):131–134, 1998.

[14] F. A. Chudak. Improved approximation algorithms for uncapacitated facility location. In *Proceedings of the 6th Conference on Integer Programming and Combinatorial Optimization*, pages 180–194. Springer, 1998.

[15] F. A. Chudak and D. B. Shmoys. Improved approximation algorithms for the uncapacitated facility location problem. *SIAM Journal on Computing*, 33(1):1–25, 2004.

[16] J. Chuzhoy and Y. Rabani. Approximating k-median with non-uniform capacities. In *Proceedings of the 16th annual ACM-SIAM symposium on Discrete algorithms*, pages 952–958. SIAM, 2005.

[17] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *Proceedings of the 35th annual ACM symposium on Theory of computing*, pages 448–455. ACM, 2003.

[18] Naveen Garg, Telikepalli Kavitha, Amit Kumar, Kurt Mehlhorn, and Julián Mestre. Assigning papers to referees. *Algorithmica*, 58(1):119–136, 2010.

[19] S. Guha and S. Khuller. Greedy strikes back: improved facility location algorithms. In *Proceedings of the 9th Annual ACM-SIAM symposium on Discrete algorithms*, pages 649–657. SIAM, 1998.

[20] S. Guha, A. Meyerson, and K. Munagala. A constant factor approximation algorithm for the fault-tolerant facility location problem. *Journal of Algorithms*, 48(2):429–440, 2003.

[21] M. T. Hajiaghayi, R. Khandekar, and G. Kortsarz. Budgeted red-blue median and its generalizations. *Proceedings of the 18th Annual European Symposium on Algorithms*, pages 314–325, 2010.

[22] K. Jain, M. Mahdian, E. Markakis, A. Saberi, and V. V. Vazirani. Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP. *Journal of the ACM*, 50(6):795–824, 2003.

[23] K. Jain and V. V. Vazirani. Primal-dual approximation algorithms for metric facility location and k-median problems. In *Proceedings of 40th Annual Symposium on Foundations of Computer Science*, pages 2–13. IEEE, 1999.

[24] K. Jain and V. V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *Journal of the ACM*, 48(2):274–296, 2001.

[25] J. Kamal and V. V. Vazirani. An approximation algorithm for the fault tolerant metric facility location problem. *Algorithmica*, 38:433–439, 2003.

[26] S. Khuller, R. Pless, and Y. Sussmann. Fault tolerant k-center problems. *Theoretical Computer Science*, 242(1-2):237–245, 2000.

[27] A. Kolen and A. Tamir. Covering problems. *Discrete location theory*, 1995:263–304, 1990.

[28] R. Krishnaswamy, A. Kumar, V. Nagarajan, Y. Sabharwal, and B. Saha. The matroid median problem. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1117–1130. SIAM, 2011.

[29] A. Kumar. Constant factor approximation algorithm for the knapsack median problem. In *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 824–832. SIAM, 2012.

[30] R. Levi, D. Shmoys, and C. Swamy. LP-based approximation algorithms for capacitated facility location. *Proceedings of the 10th Conference on Integer Programming and Combinatorial Optimization*, pages 21–27, 2004.

[31] J. Li and S. Khuller. Generalized machine activation problems. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 80–94. SIAM, 2011.

[32] S. Li. A 1.488 approximation algorithm for the uncapacitated facility location problem. In *Proceedings of the 38th international conference on Automata, languages and programming - Volume Part II*, pages 77–88. Springer, 2011.

[33] Shi Li and Ola Svensson. Approximating k-median via pseudo-approximation. In *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing*, STOC '13, pages 901–910, 2013.

[34] J. H. Lin and J. S. Vitter. e-approximations with minimum packing constraint violation (extended abstract). In *Proceedings of the 24th annual ACM symposium on Theory of computing*, pages 771–782. ACM, 1992.

[35] J.H. Lin and J.S. Vitter. Approximation algorithms for geometric median problems. *Information Processing Letters*, 44(5):245–249, 1992.

[36] M. Mahdian and M. Pál. Universal facility location. *Proceedings of the 11th Annual European Symposium on Algorithms*, pages 409–421, 2003.

[37] M. Pal, T. Tardos, and T. Wexler. Facility location with nonuniform hard capacities. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science*, pages 329–338. IEEE, 2001.

[38] A. Schrijver. *Combinatorial Optimization : Polyhedra and Efficiency*. Springer, 2003.

[39] D. B. Shmoys, É. Tardos, and K. Aardal. Approximation algorithms for facility location problems (extended abstract). In *Proceedings of the 29th annual ACM symposium on Theory of computing*, pages 265–274. ACM, 1997.

[40] Z. Svitkina. Lower-bounded facility location. *ACM Transactions on Algorithms*, 6(4):69, 2010.

[41] C. Swamy and D. B. Shmoys. Fault-tolerant facility location. *ACM Transactions on Algorithms*, 4(4):1–27, 2008.

[42] A. Tamir. An o (pn^2) algorithm for the p-median and related problems on tree graphs. *Operations Research Letters*, 19:59–64, 1996.

[43] Li Yan and Marek Chrobak. Lp-rounding algorithms for the fault-tolerant facility placement problem. In *8th International Conference on Algorithms and Complexity*, pages 370–381. Springer, 2013.