Fault-tolerant k-Supplier with Outliers

Deeparnab Chakrabarty
Department of Computer Science, Dartmouth College, Hanover, NH, USA

Luc Cote
Thayer School of Engineering, Dartmouth College, Hanover, NH, USA

Ankita Sarkar
Department of Computer Science, Dartmouth College, Hanover, NH, USA

Abstract

We present approximation algorithms for the Fault-tolerant k-Supplier with Outliers (FkSO) problem. This is a common generalization of two known problems – k-Supplier with Outliers, and Fault-tolerant k-Supplier – each of which generalize the well-known k-Supplier problem. In the k-Supplier problem the goal is to serve n clients C, by opening k facilities from a set of possible facilities F; the objective function is the farthest that any client must travel to access an open facility. In FkSO, each client v has a *fault-tolerance* ℓ_v, and now desires ℓ_v facilities to serve it; so each client v's contribution to the objective function is now its distance to the ${\ell_v}^{\text{th}}$ closest open facility. Furthermore, we are allowed to choose m clients that we will serve, and only those clients contribute to the objective function, while the remaining $n - m$ are considered outliers.

Our main result is a $(4t - 1)$-approximation for the FkSO problem, where t is the number of distinct values of ℓ_v that appear in the instance. At $t = 1$, i.e. in the case where the ℓ_v's are uniformly some ℓ, this yields a 3-approximation, improving upon the 11-approximation given for the uniform case by Inamdar and Varadarajan [2020], who also introduced the problem. Our result for the uniform case matches tight 3-approximations that exist for k-Supplier, k-Supplier with Outliers, and Fault-tolerant k-Supplier.

Our key technical contribution is an application of the *round-or-cut* schema to FkSO. Guided by an LP relaxation, we reduce to a simpler optimization problem, which we can solve to obtain distance bounds for the "round" step, and valid inequalities for the "cut" step. By varying how we reduce to the simpler problem, we get varying distance bounds – we include a variant that gives a $(2^t + 1)$-approximation, which is better for $t \in \{2, 3\}$. In addition, for $t = 1$, we give a more straightforward application of round-or-cut, yielding a 3-approximation that is much simpler than our general algorithm.

2012 ACM Subject Classification Theory of computation → Facility location and clustering

Keywords and phrases Clustering, approximation algorithms, round-or-cut

Digital Object Identifier 10.4230/LIPIcs.STACS.2024.23

Funding *Deeparnab Chakrabarty*: Partially supported by NSF grant CCF-2041920.
Ankita Sarkar: Partially supported by NSF grant CCF-2041920.

1 Introduction

Clustering problems form a class of discrete optimization problems that appear in many application areas ranging from operations research [28, 32, 30] to machine learning [21, 1, 31, 24]. They also have formed a sandbox where numerous algorithmic ideas, especially ideas in approximation algorithms, have arisen and developed over the years. One of the first clustering problems to have been studied is the k-Supplier problem [19]: in this problem, one is given a set of points in a metric space $(C \cup F, d)$, where C is the set of "clients" and F is the set of "facilities", and a number k. The objective is to "open" a collection $S \subseteq F$ of k centers so as to minimize the maximum distance between a client $v \in V$ to its nearest

41st International Symposium on Theoretical Aspects of Computer Science (STACS 2024).
Editors: Olaf Beyersdorff, Mamadou Moustapha Kanté, Orna Kupferman, and Daniel Lokshtanov; Article No. 23; pp. 23:1–23:19
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

open center in S, that is, minimize $\max_{v \in C} \min_{f \in S} d(f,v)$. It has been known since the mid-80's, due to an influential paper of Hochbaum and Shmoys [19], that this problem has a 3-approximation and no better approximation is possible[1].

One motivation behind the objective function above is that $d(v,S) := \min_{f \in S} d(f,v)$ indicates how (un)desirable the client v perceives the the set of open facilities, and the k-Supplier objective tries to take the egalitarian view of trying to minimize the unhappiest client. However, in certain applications, a client v would perhaps be interested not only in having one open facility in a small neighborhood but a larger number. For instance, the client may be worried about some open facilities closing down. This leads to the *fault-tolerant* versions of clustering problems. In this setting, each client v has an integer ℓ_v associated with it, and the desirability of a subset S for v is not determined by the nearest facility in S, but rather the ${\ell_v}^{\text{th}}$ nearest facility. That is, we sort the facilities in S in increasing order of $d(f,v)$ and let $d_{\ell_v}(v,S)$ denote the ${\ell_v}^{\text{th}}$ distance in this order (so $d(v,S) = d_1(v,S)$). The Fault-tolerant k-Supplier (FkS) problem is to find $S \subseteq F$ with $|S| = k$ so as to minimize $\max_{v \in C} d_{\ell_v}(v,S)$. As far as we know, the Fault-tolerant k-Supplier problem has not been *explicitly* studied in the literature[2], however, as we show in Section 2, there is a simple 3-approximation based on the same scheme developed by Hochbaum and Shmoys [19].

One drawback of the k-Supplier objective is that it is extremely sensitive to outliers; since one is trying to minimize the maximum, a single far-away client makes the optimal value large. To allay this, people have considered the "outlier version" of the problem, k-Supplier with Outliers (kSO). In kSO, one is given an additional integer parameter m, and the goal of the algorithm is to open a subset S of k facilities *and* recognize a subset $T \subseteq C$ of m-clients, so as to minimize $\max_{v \in T} d(v,S)$. That is, all clients outside T are deemed outliers and one doesn't consider their distance to the solution. The outlier version is algorithmically interesting and is not immediately captured by the Hochbaum-Shmoys technique. Nevertheless, in 2001, Charikar, Khuller, Mount, and Narsimhan [10] described a combinatorial, greedy-like 3-approximation for kSO[3]. Since then, outlier versions of many clustering problems have been considered, and it has been a curious feature that the approximability of the outlier version has been of the same order as the approximability of the original version without outliers.

In this paper, as suggested by the title, we study the *Fault-tolerant k-Supplier with Outliers* (FkSO) problem which generalizes FkS and kSO. This problem was explicitly studied only recently by Inamdar and Varadarajan [20]; but that work only studies the *uniformly fault-tolerant* case where all ℓ_v's are the same (say, ℓ). The main result of [20] was a "reduction" to the "non-fault-tolerant" version of the clustering problem with outliers, and their result is that an α-approximation for the kSO problem translates[4] to a $(3\alpha+2)$-approximation for the FkSO problem with uniform fault-tolerance. Setting $\alpha = 3$ from the aforementioned work [10] on kSO, one gets an 11-approximation for the uniform case of FkSO.

---

[1] Howard Karloff is attributed the hardness result in [19].

[2] although the fault-tolerant facility location and k-median have been extensively studied [22, 17, 36, 18]; more on this in Section 1.1.

[3] A different LP-based approach was taken by Chakrabarty, Goyal, and Krishnaswamy [6] and vastly generalized by Chakrabarty and Negahbani [7]; more on this in Sections 1.1 and 2.

[4] Actually, they [20] only study the "k-Center" case when $F = C$, and in that case the result is $(2\alpha+2)$; their proofs do reveal that for the Supplier version, one obtains $(3\alpha+2)$.

Our Contributions

We begin by providing a simple LP-based 3-approximation for the FkSO problem when the fault-tolerances are uniform, that is, $\ell_v = \ell$ for all $v \in C$. This improves the known 11-approximation [20]. Even this special case is interesting in that when the uniform fault-tolerance ℓ divides k, then the "natural LP" suffices to obtain a 3-approximation using a rounding scheme similar to a prior rounding algorithm for kSO [6]. However, when ℓ doesn't divide k, then we need to add in valid inequalities akin to Chvátal-Gomory cuts [13, 16] in integer programming. Nevertheless, the rounding algorithm is simple and is described in Section 3.

Our main contribution is to the general FkSO problem, when ℓ_v's can be different for different clients. This problem becomes much more complex for the simple reason that if two clients v and v' are located very close together, but $\ell_v < \ell_{v'}$, then opening ℓ_v facilities around v would still render v' unhappy – this does not happen in the uniform case. Therefore, the Hochbaum-Shmoys procedure [19], or more precisely the LP-guided Hochbaum-Shmoys rounding that is known for kSO [6], simply doesn't apply under non-uniform fault-tolerance. Indeed, the natural LP relaxation and its natural strengthening, which give us the 3-approximation for the uniform case, has large integrality gaps even when the ℓ_v's take only two values; we show this in Section 3.1.

Our main result is a $(4t-1)$-approximation for the FkSO problem when there are t distinct[5] ℓ_v's (that is, $|\{\ell_v : v \in C\}| = t$). When $t = 1$, we recover the 3-approximation mentioned above. This is not the most desirable result (one would hope a $O(1)$-approximation for any t), but as the above integrality gap example illustrates, even when $t = 2$, strong LPs have bad integrality gaps. We also use the same schema to give a $(2^t + 1)$-approximation, which gives better approximation factors for $t \in \{2, 3\}$.

Our main technical contribution is to apply the *round-or-cut* schema introduced for clustering problems by Chakrabarty and Negahbani [7] to FkSO. In particular, this schema uses a fractional solution $\{\mathsf{cov}_v\}_{v\in C}$ which indicates the extent to which each client v is an "inlier" (that is, in the final set T of at least m clients). Earlier works [6, 7] use this fractional solution to guide the Hochbaum-Shmoys-style [19] rounding algorithm, creating a partition on the set of clients and solving a simpler optimization problem on this partition. We also use the same schemata, except that our partitioning scheme is a more general one warranted by the non-uniform fault-tolerances; nevertheless, we show that we either obtain the desired approximation factor (the "round" step), or we can prove that the cov_v's cannot arise as a combination of integral solutions (the "cut" step). Once we do this, the round-or-cut schema implies a polynomial time approximation factor. We also show that t is the limiting factor in our approach; more precisely, the diameter of the parts of the desired partition dictates the upper bound on the approximation factor, and in Appendix B we construct an instance such that the diameter needs to be $\Omega(t)$. We leave the possibility of obtaining $O(1)$-approximations for FkSO, or alternately proving a super-constant hardness, as an intriguing open problem.

[5] We should point to the reader that one can't simply solve t different uniform FkSO versions and "stick them together" to get such a result; although it is a natural idea, note that a priori we do not know how many outliers one will obtain from each fault-tolerant class, and enumerating is infeasible.

1.1 Related Work

The Hochbaum-Shmoys algorithm [19] gives a 3-approximation for the k-Supplier problem, and has been extended to give approximation algorithms for multiple related problems. Plesník [33] gave one such extension, obtaining a 3-approximation[6] when each client v has weight $w(v)$, and this scales the client's "unhappiness", so that the objective function becomes $\max_{v \in C} (w(v) \cdot \min_{f \in S} d(v, f))$. In another direction, Chakrabarty, Goyal, and Krishnaswamy [6] gave an extension to k-Supplier with Outliers, using an LP relaxation to indicate which clients are outliers, and obtaining a Hochbaum-Shmoys-like 3-approximation. This was vastly extended by Chakrabarty and Negahbani [7], implying a 3-approximation for multiple problems, including kSO with knapsack constraints on the facilities. Bajpai, Chekuri, Chakrabarty, and Negahbani [4] generalized the aforementioned weighted version [33] to handle outliers, matroid constraints, and knapsack constraints, obtaining constant approximation ratios for each.

In the early 2000s, Jain and Vazirani [22] introduced the notion of fault-tolerance for the Uncapacitated Facility Location (UFL) problem. The notion has thereafter been studied for various related problems: UFL [17, 36, 5]; UFL with multiset solutions, often called facility *placement* or *allocation* [37, 38, 34]; k-Median [36, 27, 18]; matroid and knapsack Median [15]; and k-Center [26, 11, 25, 27]. In particular relevance to this paper, the FkSO problem was studied by Inamdar and Varadarajan [20]. In addition, prior work also addresses alternate notions of fault-tolerance and outlier-type constraints. In a 2020 preprint, Deng [14] combines fault-tolerance with an outlier-type constraint requiring that the number of *client-facility connections*, rather than the weight of satisfied clients, be at least some m. An altogether different notion of fault-tolerance has also been studied [12, 29, 35], where clients each want just one facility, but an adversary secretly causes some $k' \leq k$ of the chosen facilities to fail.

The round-or-cut schema that this paper applies, has found widespread usage in clustering problems, including in problems related to k-Supplier. For example, the weighted version of k-Supplier [33, 4] can be extended to impose different budgets to different weight classes – i.e. there is no longer one k, but one k_i per distinct weight w_i. This version admits a constant-factor approximation for certain special cases [8, 23] via the round-or-cut schema. Round-or-cut has also been used for k-Supplier with covering constraints [3], and for the Capacitated Facility Location problem [2]. In the continuous clustering realm, where facilities can be picked from a potentially infinite-sized ambient metric space, round-or-cut has been used to circumvent the infinitude of the instance [9].

2 Preliminaries

Before we formally define our main problem, let us set up some important notation.

▶ **Definition 1.** *Given a subset $S \subseteq F$, a client $v \in C$, and $a \in [k]$, let $d_a(v, S)$ be the distance of v to its a^{th} closest neighbor in S (breaking ties arbitrarily and consistently). So $d_1(v, S) = d(v, S)$. Also let $N_a(v, S) \subseteq S$ denote the a facilities in S that are closest to v.*

▶ **Definition 2** (Fault-tolerant k-Supplier and Fault-tolerant k-Supplier with Outliers)**.** *In the* Fault-tolerant k-Supplier (FkS) *problem, we are given a finite metric space $(C \cup F, d)$, where C is a set of n clients and F is a set of* $\text{poly}(n)$ *facilities. We are also given a parameter $k \in \mathbb{N}$, and* fault-tolerances $\{\ell_v \in [k]\}_{v \in C}$. *The goal is to open k facilities, i.e. pick $S \subseteq F : |S| \leq k$, minimizing* $\max_{v \in C} d_{\ell_v}(v, S)$.

[6] The work [33] studies the k-Center case, where $F = C$, and gives a 2-approximation; but the proofs imply a 3-approximation for the Supplier version.

In the Fault-tolerant k-Supplier with Outliers (FkSO) *problem, we are given an* FkS *instance along with an additional parameter* $m \in [n]$*. The goal is to pick* S *of size* k *as before, along with* inliers $T \subseteq C : |T| \geq m$*, minimizing* $\max_{v \in T} d_{\ell_v}(v, S)$*.*

In the absence of fault-tolerances and outliers, i.e. in the k-Supplier problem, the Hochbaum-Shmoys algorithm [19] achieves a 3-approximation as follows. It starts with a guess r of the optimum value, large enough that every client j has a facility within distance r of itself, but otherwise arbitrary. Then it picks an arbitrary client j, opens a facility within distance r of j, and deletes the set of "children" of j, which is $\mathsf{child}(j) := B(j, 2r) \cap C = \{v \in C : d(v, j) \leq 2r\}$. Then it repeats this with the remaining clients, until there are no clients left. Observe that the j's picked over the iterations – call them the set R – has the following *well-separated* property.

▶ **Definition 3** (r-well-separated set)**.** *A set* $X \subseteq C$ *is* r-well-separated *if for distinct* $x, y \in X$*, we have* $d(x, y) > 2r$*. Where* r *is clear from context, we simply say that* X *is* well-separated*.*

Since R is well-separated, it takes $|R|$ clients to provide every $j \in R$ with a facility in $B(j, r)$. So if $|R| > k$, then the guess of r is too small – we can double r and retry the algorithm. On the other hand, if $|R| \leq k$, then the guess is either correct or too large, so we halve r and retry. This binary search yields the correct r, and the following guarantee: $\{\mathsf{child}(j)\}_{j \in R}$ partitions C, and for a $v \in \mathsf{child}(j)$, since $d(v, j) \leq 2r$ and we opened a facility in $B(j, r)$, there is a facility within distance $3r$ of v. This means that we have a 3-approximation.

The Hochbaum-Shmoys algorithm described above, generalizes to give a 3-approximation for FkS via the following modifications: instead of picking j's into R in arbitrary order, we pick them in decreasing order of ℓ_v's; we also open ℓ_j facilities in each $B(j, r) : j \in R$, instead of just one. This guarantees that, if $v \in \mathsf{child}(j)$, $\ell_v \leq \ell_j$, allowing us to extend the Hochbaum-Shmoys [19] guarantee to FkS. We now formally state this algorithm.

Algorithm 1 Hochbaum-Shmoys [19] modified for FkS.

Input: C
1: $U \leftarrow C$
2: $R \leftarrow \emptyset$
3: $S \leftarrow \emptyset$
4: **while** $U \neq \emptyset$ **do**
5: $\quad j \leftarrow \operatorname{argmax}_{v \in U} \ell_v$
6: $\quad R \leftarrow R \cup \{j\}$
7: $\quad i_1, i_2, \ldots, i_{\ell_j} \leftarrow \ell_j$ arbitrary facilities in $B(j, r) \cap F$ ▷ they exist by choice of r
8: $\quad S \leftarrow S \cup \{i_1, i_2, \ldots, i_{\ell_j}\}$
9: $\quad \mathsf{child}(j) \leftarrow \{v \in U : d(v, j) \leq 2r\}$
10: $\quad U \leftarrow U \setminus \mathsf{child}(j)$
Output: $S \subseteq F$

To show that this algorithm yields a 3-approximation, we need to argue that $\forall v \in C$, $d_{\ell_v}(v, S) \leq 3r$. To see this, consider $j \in R : v \in \mathsf{child}(j)$. Algorithm 1 guarantees that $\ell_v \leq \ell_j$, so $d_{\ell_v}(v, S) \leq d_{\ell_j}(v, S)$. By triangle inequalities, this is at most $d(v, j) + d_{\ell_j}(j, S)$. By construction of $\mathsf{child}(j)$, $d(v, j) \leq 2r$; and since we open ℓ_j facilities in Line 7, $d_{\ell_j}(j, S) \leq r$. We have just shown that

▶ **Theorem 4.** *The* FkS *problem admits a 3-approximation.*

One way to achieve a 3-approximation for the k-Supplier with Outliers problem, described in [6], is as follows: under a guess of r as before, a linear program relaxation is used to assign variables $\mathsf{cov}_v \in [0,1]$ to each client v, representing whether v is "covered", i.e. whether there is an open facility in $B(v,r)$. The LP-guided Hochbaum-Shmoys algorithm considers clients in decreasing order of these cov_v's, and we wait to pick facilities until the loop terminates. Then, facilities are opened near those $j \in R$ that have the k largest $|\mathsf{child}(j)|$. The LP relaxation is used to ensure that $\geq m$ clients are served in this way. This *does not* generalize directly to FkSO because the decreasing order of ℓ_v's that we employed above for FkS can conflict with the decreasing order of cov_v's (indeed, one may just expect clients v with large ℓ_v's would be more likely to be outliers, that is, have low cov_v's). So in our algorithm for FkSO, we elect to follow the cov_v order, and explicitly force $\ell_v \leq \ell_j$ for v's that we pick into $\mathsf{child}(j)$. This choice breaks the well-separated property of R, so our techniques are devoted to obtaining other well-separated sets that can guide our rounding; details of this can be found in Section 4. When all the ℓ_v's are the same, though, we can indeed use the natural LP relaxation (with a slight strengthening to take care of divisibility issues), and we show this in the next section.

3 3-approximation for UFkSO

In this section, we address the *uniform* case, where all fault-tolerances in the instance are the same, i.e.

▶ **Definition 5** (Uniformly Fault-tolerant k-Supplier with Outliers (UFkSO))**.** *The Uniformly Fault-tolerant k-Supplier with Outliers problem is a special case of the* FkSO *problem where, for an $\ell \in \mathbb{N}$, $\forall v \in C$, $\ell_v = \ell$.*

We prove that

▶ **Theorem 6.** *The* UFkSO *problem admits a 3-approximation.*

Our algorithm begins by rounding a solution to the following LP relaxation, closely mimicking the 3-approximation for kSO [6] described in the last paragraph of the previous section. This rounding suffices when $\ell \mid k$. When $\ell \nmid k$, we identify a valid inequality for the round-or-cut framework. In the LP, the variables $\{\mathsf{cov}_v\}_{v\in C}$ denote whether or not a client $v \in C$ is *covered* i.e. served within distance r; and variables $\{x_i\}_{i\in F}$ denote whether or not a facility $i \in F$ is open. $B(v,r)$ is the *ball* of radius r around v, containing all points within distance r of v, i.e. $B(v,r) := \{x \in C \cup F : d(v,x) \leq r\}$.

$$\sum_{v\in C} \mathsf{cov}_v \geq m \tag{WL1}$$

$$\sum_{i\in F} x_i \leq k \tag{WL2}$$

$$\forall v \in C, \quad \sum_{i\in F\cap B(v,r)} x_i \geq \ell \mathsf{cov}_v \tag{WL3}$$

$$\forall v \in C : d_\ell(v,F) > r, \quad \mathsf{cov}_v = 0 \tag{WL4}$$

$$\forall v \in C, i \in F, \quad 0 \leq \mathsf{cov}_v, x_i \leq 1 \tag{WL5}$$

Here, (WL1) enforces that at least m clients must be covered, and (WL2) enforces that at most k facilities can be opened. (WL3) and (WL4) connect the cov_v variables with the x_i variables, ensuring that a client cannot be covered unless there are sufficient facilities opened within distance r of it. Finally, (WL5) enforces that a client can be covered only once, and a facility can be opened only once. Claim 7 shows that this LP is a valid relaxation of our problem. We defer its proof to Appendix A.

▷ Claim 7. An instance of UFkSO is feasible iff it admits an integral solution satisfying (WL1)-(WL5).

Given a solution $\left(\{\mathsf{cov}_v\}_{v\in C}, \{x_i\}_{i\in F}\right)$ satisfying (WL1)-(WL5), we round as per Algorithm 2. This algorithm constructs a well-separated set of *representatives* $R_{\mathsf{cov}} \subseteq C$. Each client v that has $\mathsf{cov}_v > 0$ becomes the *child* of some representative, yielding a partition $\{\mathsf{child}(j)\}_{j\in R_{\mathsf{cov}}}$ of these clients. Then, facilities $S_{\mathsf{cov}} \subseteq F$ are opened in a manner that serves the $\left\lfloor \frac{k}{\ell} \right\rfloor$ largest $\mathsf{child}(j)$ sets within distance $3r$.

Algorithm 2 3-approximation for UFkSO.

Input: $\left(\{\mathsf{cov}_v\}_{v\in C}, \{x_i\}_{i\in F}\right)$ satisfying (WL1)-(WL5)

1: $R_{\mathsf{cov}} \leftarrow \emptyset$
2: $U \leftarrow \{v \in C : \mathsf{cov}_v > 0\}$
3: **while** $U \neq \emptyset$ **do** ▷ filtering
4: $\quad j \leftarrow \mathrm{argmax}_{v\in U}\, \mathsf{cov}_v$
5: $\quad R_{\mathsf{cov}} \leftarrow R_{\mathsf{cov}} \cup \{v\}$
6: $\quad \mathsf{child}(j) \leftarrow B(j, 2r) \cap U$
7: $\quad U \leftarrow U \setminus \mathsf{child}(j)$
8: $S_{\mathsf{cov}} \leftarrow \emptyset$
9: $R' \leftarrow R_{\mathsf{cov}}$
10: **while** $|S_{\mathsf{cov}}| < \left\lfloor \frac{k}{\ell} \right\rfloor \cdot \ell$ **do** ▷ picking facilities to open
11: $\quad j \leftarrow \mathrm{argmax}_{j'\in R'}\, |\mathsf{child}(j')|$
12: $\quad R' \leftarrow R' \setminus \{j\}$
13: $\quad S_{\mathsf{cov}} \leftarrow S_{\mathsf{cov}} \cup N_\ell(j, F)$
14: **return** S_{cov}

Output: $S_{\mathsf{cov}} \subseteq F$ ▷ open facilities

We argue that Algorithm 2 opens at most k facilities, and that if $N_\ell(j, F)$ is opened, then $\mathsf{child}(j)$ is served within distance $3r$. Formally,

▶ **Lemma 8.** *Given* $\left(\{\mathsf{cov}_v\}_{v\in C}, \{x_i\}_{i\in F}\right)$ *satisfying* (WL1)-(WL5),

- $|S_{\mathsf{cov}}| \leq k$, *and*
- *Let* R'_{cov} *be the clients* j *for which* $N_\ell(j, F)$ *was added to* S_{cov} *in Line 13. Then* $\forall j \in R'_{\mathsf{cov}}$, $\forall v \in \mathsf{child}(j)$, $d_\ell(v, S_{\mathsf{cov}}) \leq 3r$.

Proof.

- Line 13 adds ℓ facilities to S_{cov} in each iteration. So Line 10 ensures that $|S_{\mathsf{cov}}| \leq k$.
- Consider $v \in \mathsf{child}(j)$, $j \in R'_{\mathsf{cov}}$. By triangle inequalities, $d_\ell(v, S_{\mathsf{cov}}) \leq d(v, j) + d_\ell(j, S_{\mathsf{cov}})$. By Line 6, $d(v, j) \leq 2r$. Since $N_\ell(j, F) \subseteq S_{\mathsf{cov}}$, $d_\ell(j, S_{\mathsf{cov}}) \leq d_\ell(j, F)$; and by Line 2, $\mathsf{cov}_j > 0$, i.e. by (WL4), $d_\ell(j, F) \leq r$. ◀

It remains to show that $\sum_{j\in R'_{\mathsf{cov}}} |\mathsf{child}(j)| \geq m$. We have

$$\sum_{j\in R_{\mathsf{cov}}} |\mathsf{child}(j)|\, \mathsf{cov}_j \geq \sum_{v\in C:\mathsf{cov}_v>0} \mathsf{cov}_v = \sum_{v\in C} \mathsf{cov}_v \geq m\,, \tag{1}$$

where the first inequality is by Line 4 and the last inequality is by (WL1). We also have

$$\sum_{j\in R_{\mathsf{cov}}} \mathsf{cov}_j \le \sum_{j\in R_{\mathsf{cov}}} \sum_{i\in F\cap B(j,r)} \frac{x_i}{\ell} \le \sum_{i\in F} \frac{x_i}{\ell} \le \frac{k}{\ell}\,, \tag{2}$$

where the first inequality is by (WL3); the second inequality is because R_{cov} is well-separated; and the last inequality is by (WL2). So we can view the LHS in (1) as a weighted sum of $|\mathsf{child}(j)|$ values, the weights being cov_j's. Since this weighted sum is $\ge m$ and the weights sum to $\le k/\ell$, the $\lfloor \frac{k}{\ell} \rfloor$ largest child-sets must contain at least $\frac{m}{k/\ell} \cdot \lfloor \frac{k}{\ell} \rfloor$ elements. Hence, if $\ell \mid k$, we are done.

In fact, we observe the following even when $\ell \nmid k$: if we can replace the RHS in (2) with $\lfloor \frac{k}{\ell} \rfloor$, then the weighted-sum argument would yield $\frac{m}{\lfloor k/\ell \rfloor} \cdot \lfloor \frac{k}{\ell} \rfloor = m$. To achieve this, observe that the argument in (2) applies to any well-separated set $R \subseteq C$, yielding $\sum_{j\in R} \mathsf{cov}_j \le k/\ell$. Also, for any integral solution, the RHS can be replaced by its floor. Thus the following are valid inequalities:

$$\forall R \subseteq C : R \text{ is well-separated}, \quad \sum_{j\in R} \mathsf{cov}_j \le \left\lfloor \frac{k}{\ell} \right\rfloor . \tag{WLCut}$$

We have showed that if (WLCut) holds for $R = R_{\mathsf{cov}}$ then we are done, i.e.

▶ **Lemma 9.** *Given* $\left(\{\mathsf{cov}_v\}_{v\in C}, \{x_i\}_{i\in F}\right)$ *satisfying* (WL1)-(WL5)*, if* (WLCut) *holds for* $R = R_{\mathsf{cov}}$ *where* R_{cov} *is constructed as per Lines 1-7, then* S_{cov} *is a* 3*-approximation.*

Using this, we now present our overall algorithm via a round-or-cut schema.

Proof of Theorem 6. Given $\left(\{\mathsf{cov}_v\}_{v\in C}, \{x_i\}_{i\in F}\right)$ satisfying (WL1)-(WL5), we round as per Lines 1-7 to obtain $R_{\mathsf{cov}} \subseteq C$. If (WLCut) holds for $R = R_{\mathsf{cov}}$, then we continue Algorithm 2 to obtain S_{cov} that satisfies the desired guarantees via Lemmas 8 and 9. Otherwise, we know that the valid inequality (WLCut) for $R = R_{\mathsf{cov}}$ is violated. So we pass it to the ellipsoid algorithm as a separating hyperplane, obtaining fresh cov_v's with which we restart Algorithm 2. By the guarantees of the ellipsoid algorithm, in polynomial time, we either round to get S_{cov}, or detect that the guess of r is too small. ◀

We conclude this section by exhibiting that the above algorithm fails for the general problem. In particular, we exhibit an infinite integrality gap when there are just two different fault-tolerances in the instance.

3.1 Gap example for FkSO

Consider (WL3) generalized to FkSO:

$$\forall v \in C, \qquad \sum_{i\in F\cap B(v,r)} x_{iv} \ge \ell_v \mathsf{cov}_v\,; \tag{WL3$'$}$$

and a similar generalization of (WLCut):

$$\forall R \subseteq C : R \text{ is well-separated}, \quad \left\lceil \sum_{v\in R} \ell_v \mathsf{cov}_v \right\rceil \le k\,. \tag{WLCut$'$}$$

These, along with (WL1)-(WL2) and (WL4)-(WL5), generalize the earlier LP to FkSO. We now show an infinite integrality gap w.r.t. this LP.

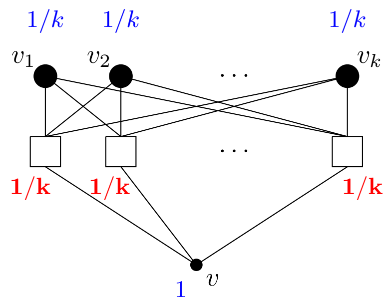

Figure 1 One of the k identical gadgets in the gap example, showing LP values in **red** (x values) and blue (cov values). The "edges" represent distance 1, and all other distances are determined by making triangle inequalities tight. The fault-tolerances are $\ell_{v_1} = \ell_{v_2} = \cdots = \ell_{v_k} = k$, and $\ell_v = 1$.

Consider k identical gadgets, each like in Figure 1, infinitely apart from each other. Let $m = 2k$. The small client in each gadget (v in Figure 1) has fault-tolerance 1. The big clients in each gadget ($v_1, v_2, \ldots, v_k$ in Figure 1) have fault-tolerance k. Within a gadget, an integral solution only benefits from either picking one facility to serve just the small client, or picking all facilities to serve all $(k+1)$ clients. So over all gadgets, an integral solution can either pick one facility per gadget, or pick all facilities in exactly one gadget, either way serving $k < m$ clients. Since all facilities are within distance 1 of the clients in their gadget, the above is true for an integral solution with any radius dilation $\alpha \geq 1$.

But the LP can assign $x_i = 1/k$ to each of the k^2 facilities in the instance. This allows it to assign $\mathsf{cov}_v = 1$ to all the small clients, and $\mathsf{cov}_{v_1} = \mathsf{cov}_{v_2} = \cdots \mathsf{cov}_{v_k} = 1/k$ to all the big clients, thus serving $k \cdot 1 + k^2 \cdot \frac{1}{k} = 2k = m$ clients.

4 Fault-tolerant k-Supplier with Outliers

In this section, we address FkSO in its full generality. We use t to denote the number of distinct fault-tolerances in the instance, i.e. $|\{\ell_v : v \in C\}| = t$. We prove that

▶ **Theorem 10.** *The* FkSO *problem admits a* $(\min\{4t-1, 2^t+1\})$*-approximation.*

4.1 Strong LP Relaxation and the Round-or-Cut Schema

To circumvent the gap example in Section 3, we adapt the following stronger linear program idea from Chakrabarty and Negahbani [7]. As before, r is the guess of the optimal solution, and we have the same fractional variables cov_v indicating coverage. However, we assert that these cov_v's arise as a convex combination of integral solutions. More precisely, we have *exponentially many* auxilary variables $\{z_S\}_{S \subseteq F : |S| \leq k}$ indicating possible locations of open facilities and the fractional amount to which they are open. When such a solution is opened, a client v is "covered" if there are ℓ_v facilities in an r-neighborhood. To this end, for a client v, we define the collection $\mathcal{F}_v := \{S \subseteq F : |S| \leq k \wedge |S \cap B(v,r)| \geq \ell_v\}$ of solutions which can serve v. Therefore, the coverage cov_v is simply the total fractional weight of sets in $\mathcal{F}_v$. Formally, if r is a correct guess, then the following (huge) LP has a feasible solution.

$$\sum_{v \in C} \mathsf{cov}_v \geq m \tag{L1}$$

$$\forall v \in C,\ \mathsf{cov}_v = \sum_{S \in \mathcal{F}_v} z_S \tag{L2}$$

$$\sum_{S \subseteq F : |S| \leq k} z_S \leq 1 \tag{L3}$$

$$\forall S \subseteq F, \forall v \in C,\ 0 \leq z_S, \mathsf{cov}_v \leq 1 \tag{L4}$$

(L1) enforces that at least m clients must be covered. (L2) connects the cov_v and z_S variables, ensuring that a client v can only be covered via solutions in $\mathcal{F}_v$. (L3)-(L4) enforce convexity. (L4) also enforces that each client can be covered at most once.

▶ **Observation 11.** *All* cov_v*'s that satisfy* (L1)-(L4) *also satisfy* (WL1)-(WL5)*.*

Also observe that we cannot efficiently figure out whether the above system is feasible or not; indeed, if so we would solve the Fault-tolerant k-Supplier with Outliers problem optimally. Nevertheless, one can use the round-or-cut schema to obtain an *approximation* algorithm. In order to do so, the first step is to use the *dual* of the above system to obtain the collection of all valid inequalities on the cov_v's. Recall, a valid inequality is one that every feasible cov_v must satisfy; the lemma below from the literature [6], in some sense, eliminates all the z_S variables from the above program.

▶ **Lemma 12** ([7, Lemma 10])**.** *Given real numbers* $\{\lambda_v\}_{v \in C}$ *such that*

$$\forall S \subseteq F, \quad \sum_{v \in C : S \in \mathcal{F}_v} \lambda_v < m\,, \tag{λ1}$$

the following is a valid inequality for (L1)-(L4)*:*

$$\sum_{v \in C} \lambda_v \mathsf{cov}_v < m\,. \tag{λ2}$$

Given $\{\lambda_v\}_{v \in C}$, one cannot easily check (λ1), and thus, a priori, one cannot see the usefulness of the above lemma. We now briefly describe its usefulness to the round-or-cut schema. The algorithm begins with values of $\{0 \leq \mathsf{cov}_v \leq 1\}_{v \in C}$ that satisfy (L1) – such cov is straightforward to find. We then try to use these cov_v's to "round" and obtain a solution where clients are covered within distance $\alpha \cdot r$ for desired factor α, and if we fail, then we find a valid inequality that "cuts" cov_v away from the above system. If we can do so, then we can feed this separating hyperplane to the ellipsoid algorithm which would give us new cov_v's. Repeating the above procedure a polynomial number of times, we would either obtain an α-approximation, or prove that the above system is empty implying our guess r was too small. For $\mathsf{F}k\mathsf{SO}$, the "round" step is via the abstract concept of a "good partition" where the "radius" of the partition dictates the approximation factor; this definition and resulting rounding algorithm is described in Section 4.2. For the "cut" step, we show that if our rounding algorithm fails, then we can use this failure to generate $\{\lambda_v\}_{v \in C}$'s that satisfy (λ1) but not (λ2), leveraging our definition of "good partitions". This gives our separating hyperplane using Lemma 12, and we succeed in cutting, and thus we can run the round-or-cut schema. Subsequently, we construct good partitions. In Section 4.3, we describe two methods to do this: one with "radius" $(4t-1)$ and the other with radius (2^t+1). In Appendix B, we show a limitation of our approach, via an example where this "radius" can be $\Omega(t)$.

Before proceeding, we make one simplification: at the beginning of every rounding step, we discard any clients that have $\mathsf{cov}_v = 0$, and hereafter assume, without loss of generality, that $\forall v \in C$, $\mathsf{cov}_v > 0$.

4.2 Good Partitions and Implementing Round-or-cut

Given cov_v's for every $v \in C$, we define a notion of a "good partition". Before formally defining it, we explain this operationally, hopefully giving intuition for the definition. We start with a finer partition, and the good partition $\mathcal{P}$ coarsens it. As in previous algorithms discussed so far, we have $R \subseteq C$, a set of *representatives*. The finer partition is $\{\mathsf{child}(j)\}_{j\in R}$, as motivated by our algorithms for FkS in Section 2 and UFkSO in Section 3. This time, however, we want favorable properties from both of those algorithms to coincide – we want, for $j \in R$ and $v \in \mathsf{child}(j)$, $\mathsf{cov}_v \le \mathsf{cov}_j$ as well as $\ell_v \le \ell_j$. These desired properties of the finer partition are formalized as Property 1.

The property above breaks the "well-separated" property of R, which was crucial in our other algorithms in Sections 2 and 3. Therefore, instead of requiring R to be well-separated, we coalesce the child-sets of certain representatives, to get a *coarsening* $\mathcal{P}$ of $\{\mathsf{child}(j)\}_{j\in R}$ such that representatives *across* different parts of $\mathcal{P}$ are indeed well-separated. This is Property 2.

Our approximation ratio is then determined by the diameter of the parts P's in the good partition; so we impose a radius bound on each $P \in \mathcal{P}$, requiring that the highest-fault-tolerance client in each P be not too far from the rest of P. This is Property 3. We are now ready to present the formal definition.

▶ **Definition 13** ((ρ, cov)-good partition). *Given a parameter $\rho \in \mathbb{R}$, and $\{0 \le \mathsf{cov}_v \le 1\}_{v\in C}$ satisfying* (L1)*, a partition $\mathcal{P}$ of C is (ρ, cov)-good if there exists $R \subseteq C$ such that the following hold.*

1. *Every $v \in C$ is assigned to be the* child *of a $j \in R$, forming a partition $\{\mathsf{child}(j)\}_{j\in R}$ of C that refines $\mathcal{P}$. Also, $\forall j \in R, \forall v \in \mathsf{child}(j)$, $\mathsf{cov}_j \ge \mathsf{cov}_v$ and $\ell_j \ge \ell_v$.*
2. *For any two $j, j' \in R$ that lie in different parts of $\mathcal{P}$, $d(j, j') > 2r$.*
3. *For each $P \in \mathcal{P}$, let $j_P := \operatorname{argmax}_{v\in P} \ell_v$ (breaking ties arbitrarily). Then $\forall v \in P$, $d(j_P, v) \le \rho r$.*

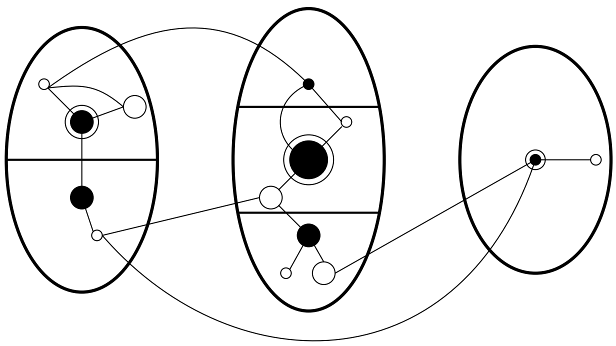

Figure 2 An example of a $(6, \mathsf{cov})$-good partition $\mathcal{P}$ (Definition 13). The ellipses represent $\mathcal{P}$, and their subdivisions represent the child sets. All the circles are clients, with the filled-in circles being R, and among those, the double borders indicate the j_P's. cov values are 1 on R and 1/2 elsewhere. ℓ_v values are indicated by the sizes of the circles. The "edges" represent distance $2r$, and all other distances are obtained by making triangle inequalities tight.

We observe here that the child-sets constructed in Section 3 are themselves a good partition, so for UFkSO, we did not need to coarsen it. This will not necessarily be the case for child-sets that we construct in Section 4.3. We also observe that

▶ **Observation 14.** *In a* (ρ, cov)*-good partition* $\mathcal{P}$*, by Property 1, the* j_P*'s in Property 3 can be chosen such that* $\forall P \in \mathcal{P}$*,* $j_P \in R$*. So we can assume, without loss of generality, that all* j_P*'s are in* R*.*

We prove that a good partition suffices to achieve our desired approximation. That is,

▶ **Theorem 15.** *If we have a feasible instance with a* (ρ, cov)*-good partition, then in polynomial time, we can either obtain a* $(\rho+1)$*-approximation, or identify a valid inequality for* (L1)-(L4) *that is violated by* cov*.*

To prove Theorem 15, we solve a budgeting problem on the (ρ, cov)-good partition. We want to distribute our budget of k facilities among the $P \in \mathcal{P}$, assigning each $P \in \mathcal{P}$ with k_P facilities that are within distance $(\rho+1)r$ of the clients in P. Here k_P must be at most $\ell_P := \ell_{j_P}$, because at most ℓ_P facilities are guaranteed to exist within a bounded distance of clients in P. The payoff from assigning k_P facilities to P in this way is that the clients $\{v \in P : \ell_v \leq k_P\}$ are served within distance $(\rho+1)r$. So if $\sum_{P\in\mathcal{P}} |\{v \in P : \ell_v \leq k_P\}| \geq m$, we have our $(\rho+1)$-approximation. Therefore, we want our choice of k_P's to maximize $\sum_{P\in\mathcal{P}} |\{v \in P : \ell_v \leq k_P\}|$, and this maximum to be $\geq m$. However, our analysis can only handle clients from well-separated sets; so instead, we maximize the following lower-bound on our desired quantity: $\sum_{P\in\mathcal{P}} \sum_{j\in R\cap P:\ell_j\leq k_P} |\mathsf{child}(j)|$, where we under-count by only considering $v \in \mathsf{child}(j)$ served if j is served. Formally, our budgeting problem is the following.

▶ **Definition 16** (Budgeting over a (ρ, cov)-good partition)**.** *Given a* (ρ, cov)*-good partition* $\mathcal{P}$*, let* $\ell_P := \max_{v\in P} \ell_v$*. Find* $\{k_P \leq \ell_P\}_{P\in\mathcal{P}}$ *such that* $\sum_{P\in\mathcal{P}} k_P \leq k$*, maximizing* $\sum_{P\in\mathcal{P}} \sum_{j\in R\cap P:\ell_j\leq k_P} |\mathsf{child}(j)|$*. Let* $\mathtt{opt}_B(\mathcal{P})$ *denote this maximum.*

In Lemma 17, we show that if $\mathtt{opt}_B(\mathcal{P}) \geq m$, then we can round. Then in Lemma 18, we see that if $\mathtt{opt}_B(\mathcal{P}) < m$, then we can cut. Lemma 19 shows that $\mathtt{opt}_B(\mathcal{P})$ can be found efficiently. Together, these three lemmas yield the proof of Theorem 15.

▶ **Lemma 17.** *Given a* (ρ, cov)*-good partition* $\mathcal{P}$*, if* $\mathtt{opt}_B(\mathcal{P}) \geq m$*, then we have a* $(\rho+1)$*-approximation.*

Proof. Let $\{k_P\}_{P\in\mathcal{P}}$ be an optimal solution to the budgeting problem (Definition 16). Define $S := \cup_{P\in\mathcal{P}} N_{k_P}(j_P, F)$. So $|S| \leq k$. We show that S serves $\geq m$ clients within distance $(\rho+1)r$.

Define $T := \uplus_{P\in\mathcal{P}} \uplus_{j\in R\cap P:\ell_j\leq k_P} \mathsf{child}(j)$. Then $|T| = \sum_{P\in\mathcal{P}} \sum_{j\in R\cap P:\ell_j\leq k_P} |\mathsf{child}(j)| = \mathtt{opt}_B(\mathcal{P}) \geq m$. We complete this proof by showing that $\forall v \in T$, $d_{\ell_v}(v, S) \leq (\rho+1)r$. For this, fix $v \in T$. By triangle inequalities, we have that $d_{\ell_v}(v, S) \leq d(v, j_P) + d_{\ell_v}(j_P, S)$. By Property 3, $d(v, j_P) \leq \rho r$, so it remains to show that $d_{\ell_v}(j_P, S) \leq r$.

By definition of T, $d_{\ell_v}(j_P, S) \leq d_{k_P}(j_P, S)$. Since $N_{k_P}(j_P, F) \subseteq S$, $d_{k_P}(j_P, S) \leq d_{k_P}(j_P, F)$. By definitions of k_P and ℓ_P, $d_{k_P}(j_P, F) \leq d_{\ell_P}(j_P, F) = d_{\ell_{j_P}}(j_P, F)$. But $\mathsf{cov}_{j_P} > 0$; so by Observation 11 and (WL4), $d_{\ell_{j_P}}(j_P, F) \leq r$. ◀

▶ **Lemma 18.** *Given a* (ρ, cov)*-good partition* $\mathcal{P}$*, if* $\mathtt{opt}_B(\mathcal{P}) < m$*, then we find a valid inequality for* (L1)-(L4) *that is violated by* cov*.*

Proof. We appeal to Lemma 12 mentioned in Section 4.1. $\forall v \in C$, define

$$\lambda_v := \begin{cases} |\mathsf{child}(v)| & \text{if } v \in R, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

Note that

$$
\begin{aligned}
\sum_{v\in C}\lambda_v \mathsf{cov}_v = \sum_{j\in R}\lambda_j \mathsf{cov}_j = \sum_{j\in R}|\mathsf{child}(j)|\,\mathsf{cov}_j &= \sum_{j\in R}\sum_{v\in\mathsf{child}(j)} \mathsf{cov}_j \\
&\geq \sum_{j\in R}\sum_{v\in\mathsf{child}(j)} \mathsf{cov}_v && \text{... by Property 1} \\
&= \sum_{v\in C}\mathsf{cov}_v && \text{... by Definition 13} \\
&\geq m\,, && \text{... by (L1)}
\end{aligned}
$$

i.e. these λ_v's violate (λ2). So by Lemma 12, it suffices to show that (λ1) holds for these λ_v's.

Suppose not, i.e. $\exists S_0 \subseteq F : |S_0| \leq k$ and $\sum_{v\in C:S_0\in\mathcal{F}_v}\lambda_v \geq m$. Then, devise a candidate solution $\{k'_P\}_{P\in\mathcal{P}}$ for the budgeting problem in Definition 16, as follows. For each $P\in\mathcal{P}$, if $\exists j\in R\cap P$ such that $S_0\in\mathcal{F}_j$, then set k'_P to be the largest fault-tolerance among such j's; that is, where $j'_P := \operatorname{argmax}_{j\in R\cap P:S_0\in\mathcal{F}_j}\ell_j$, set $k'_P := \ell_{j'_P}$. Otherwise, i.e. when there is no such j and j'_P is not well-defined, set $k'_P := 0$. By definitions, $\forall P\in\mathcal{P}$, $k'_P \leq \ell_P$.

Also, by Property 2, $\{B(j'_P, r)\}_{P\in\mathcal{P}}$ is pairwise disjoint. Since $S_0\in\mathcal{F}_{j'_P}$ for each $P\in\mathcal{P}$, we then have $\sum_{P\in\mathcal{P}} k'_P \leq \sum_{P\in\mathcal{P}}|S_0\cap B(j'_P,r)| \leq |S_0| \leq k$. So $\{k'_P\}_{P\in\mathcal{P}}$ is indeed a candidate solution for the budgeting problem. We evaluate the objective function of the budgeting problem (see Definition 16) on $\{k'_P\}_{P\in\mathcal{P}}$:

$$
\begin{aligned}
\sum_{P\in\mathcal{P}}\sum_{j\in R\cap P:\ell_j\leq k'_P}|\mathsf{child}(j)| &= \sum_{P\in\mathcal{P}}\sum_{j\in R\cap P:\ell_j\leq k'_P}\lambda_j \\
&\geq \sum_{P\in\mathcal{P}}\sum_{j\in R\cap P:S_0\in\mathcal{F}_j}\lambda_j && \text{... by choice of } k'_P\text{'s} \\
&= \sum_{j\in R:S_0\in\mathcal{F}_j}\lambda_j && \text{... by Definition 13} \\
&= \sum_{v\in C:S_0\in\mathcal{F}_v}\lambda_v && \text{... by choice of } \lambda_v\text{'s} \\
&\geq m && \text{... by supposition.}
\end{aligned}
$$

So $\{k'_P\}_{P\in\mathcal{P}}$ is a candidate solution to the budgeting problem, for which the objective function evaluates to $\geq m$, contradicting $\mathtt{opt}_B(\mathcal{P}) < m$. Hence ($\lambda$1) holds for our chosen λ_v's, and (λ2) is the desired valid inequality that is violated by cov. ◀

▶ **Lemma 19.** *The budgeting problem in Definition 16 can be solved in polynomial time.*

Proof. We proceed via dynamic programming. Let $N := |\mathcal{P}|$. Without loss of generality, say $\mathcal{P} =: \{P_1, P_2, \ldots, P_N\}$. For brevity, $\forall a\in[N]$, we say $L_a := \ell_{P_a}$. To handle base cases in our DP, we set the convention that $P_0 := \emptyset$. Now define the entries in our DP table: $\forall\nu\in[N]\cup\{0\}$ and $\forall b\in[k]\cup\{0\}$,

$$
M[\nu,b] := \max_{\{k_a\leq L_a\}_{a=1}^{\nu}:\sum_{a=1}^{\nu}k_a\leq b}\ \sum_{a=1}^{\nu}\sum_{j\in R\cap P_a:\ell_j\leq k_a}|\mathsf{child}(j)|\,. \tag{DP-defn}
$$

The desired entry is $M[N,k]$, as the corresponding $\{k_a\}_{a=1}^{N}$ becomes, upon renaming as $\{k_{P_a} = k_a\}_{a=1}^{N}$, the k_P's that we want.

The base cases are: $M[0,0]=0$; $\forall \nu \in [N]$, $M[\nu,0]=0$; and $\forall b \in [k]$, $M[0,b]=0$. The DP table has $O(Nk)=O(nk)$ entries; so in polynomial time, we can fil it via the following recurrence.

$$M[\nu,b] := \max_{\ell=0}^{\min(b,L_\nu)} \left(M[\nu-1,b-\ell] + \sum_{j\in R\cap P_\nu:\ell_j\le \ell} |\mathsf{child}(j)| \right). \tag{DP-rec}$$

We also remember, for each entry $M[\nu,b]$, the ℓ that maximizes the RHS of (DP-rec). Note, in (DP-defn), that the RHS for $M[N,k]$ corresponds, up to renaming, with the RHS in the objective function (see Definition 16). Thus it remains to show that (DP-rec) is correct wrt (DP-defn).

- To show that LHS $\le$ RHS, consider the solution $\{k_a^*\}_{a=1}^{\nu}$ corresponding to $M[\nu,b]$. By (DP-defn), $k_\nu^* \le \min(b,L_\nu)$. So $\{k_a^*\}_{a=1}^{\nu-1}$ is a candidate solution for $M\left[\nu-1,b-k_a^*\right]$, i.e. $\sum_{a=1}^{\nu-1}\sum_{j\in R\cap P_a:\ell_j\le k_a^*}|\mathsf{child}(j)| \le M\left[\nu-1,b-k_a^*\right]$, so

$$\begin{aligned}\text{LHS} = M[\nu,b] &= \sum_{a=1}^{\nu}\sum_{j\in R\cap P_a:\ell_j\le k^*a}|\mathsf{child}(j)| \\ &\le M\left[\nu-1,b-k_a^*\right] + \sum_{j\in R\cap P_\nu:\ell_j\le k_\nu^*}|\mathsf{child}(j)| \le \text{RHS}\end{aligned}$$

  since the RHS is a maximum.
- To show that RHS $\le$ LHS, fix an $\ell \in \{0,\dots,\min(b,L_\nu)\}$, and let $\{k'_a\}_{a=1}^{\nu-1}$ be the solution corresponding to $M[\nu-1,b-\ell]$. Setting $k'_\nu=\ell$ yields $\{k'_a\}_{a=1}^{\nu}$, a candidate solution for $M[\nu,b]$. So

$$M[\nu-1,b-\ell] + \sum_{j\in R\cap P_\nu:\ell_j\le\ell}|\mathsf{child}(j)| = \sum_{a=1}^{\nu}\sum_{j\in R\cap P_a:\ell_j\le k'_a}|\mathsf{child}(j)| \le M[\nu,b] = \text{LHS}$$

  since $M[\nu,b]$ is a maximum by (DP-defn).

  As the RHS maximizes over $\ell\in\{0,\dots,\min(b,L_\nu)\}$, we are done. ◀

Proof of Theorem 15. Given a (ρ,cov)-good partition $\mathcal{P}$, we solve the budgeting problem (Definition 16), which we can do efficiently due to Lemma 19, and obtain $\mathtt{opt}_B(\mathcal{P})$. If $\mathtt{opt}_B(\mathcal{P}) \ge m$, Lemma 17 guarantees a $(\rho+1)$-approximation; otherwise, Lemma 18 gives a valid inequality that is violated by cov. We pass the valid inequality as a separating hyperplane to the ellipsoid algorithm, and restart our rounding process with fresh cov_v's. By the guarantees of ellipsoid, in polynomial time, we either round to obtain a $(\rho+1)$-approximation, or detect that the guess of r is too small. ◀

4.3 Obtaining a good partition

▶ **Theorem 20.** *Given $\{0\le \mathsf{cov}_v \le 1\}_{v\in C}$, in polynomial time, we can obtain the following:*
1. *a $(4t-2,\mathsf{cov})$-good partition, and*
2. *a $(2^t,\mathsf{cov})$-good partition.*

Theorem 15 follows from Lemmas 21 and 22.

▶ **Lemma 21.** *Algorithm 3 yields a $(4t-2,\mathsf{cov})$-good partition.*

Algorithm 3 Finding a $(4t-2, \mathsf{cov})$-good partition.

Input: $\{0 \le \mathsf{cov}_v \le 1\}_{v \in C}$
1: $U \leftarrow C$
2: $R \leftarrow \emptyset$
3: **while** $U \neq \emptyset$ **do**
4: $\quad j \leftarrow \text{argmax}_{v \in U} \mathsf{cov}_v$
5: $\quad R \leftarrow R \cup \{j\}$
6: $\quad \mathsf{child}(j) \leftarrow \{v \in U : d(v, j) \le 2tr \wedge \ell_v \le \ell_j\}$
7: $\quad U \leftarrow U \setminus \mathsf{child}(j)$
8: $\mathcal{P} \leftarrow \emptyset$
9: $G \leftarrow (R, E := \{\{j, j'\} : d(j, j') \le 2r\})$ ▷ undirected graph
10: $\mathcal{C} \leftarrow$ connected components of G
11: $\mathcal{P} \leftarrow \{\cup_{j \in V} \mathsf{child}(j)\}_{V \in \mathcal{C}}$
Output: A partition $\mathcal{P}$ of C.

Proof. Consider $\mathcal{P}$, the output of Algorithm 3, and the child and R constructed alongside. Line 7 ensures that $\{\mathsf{child}(j)\}_{j \in R}$ is a partition of C. Line 11 ensures that this partition is a refinement of $\mathcal{P}$. Lines 4 and 6 construct child as desired, ensuring that $\forall j \in R$, $\forall v \in \mathsf{child}(j)$, $\mathsf{cov}_j \ge \mathsf{cov}_v$ and $\ell_j \ge \ell_v$. So Property 1 holds.

Now consider $P_1, P_2 \in \mathcal{P}$, $x_1 \in R \cap P_1, x_2 \in R \cap P_2 : P_1 \neq P_2$. By Lines 9-10, $R \cap P_1$ and $R \cap P_2$ are distinct connected components in $\mathcal{C}$, so $\{x_1, x_2\} \notin E$, i.e. $d(x_1, x_2) > 2r$. This shows that Property 2 holds.

Finally, consider $P \in \mathcal{P}$, and $v \in P$ s.t. $v \in \mathsf{child}(j_1)$ for $j_1 \in R$. By Line 11, $j_1 \in R \cap P$. Also consider a different $j_2 \in R \cap P$. By Lines 9-10, $R \cap P \in \mathcal{C}$. In G, consider π, the shortest j_1-j_2 path passing entirely through $R \cap P$. We claim that

▷ Claim. π contains at most t vertices.

Proof. Suppose not. Then, by the pigeonhole principle, π contains vertices $u, v \in R \cap P$ s.t. $u \neq v$ and $\ell_u = \ell_v$. Choose such u, v minimizing $d(u, v)$, and consider the u-v subpath π' of π. If π' contains $> t$ vertices, then we can replace j_1, j_2 with u, v and repeat our argument to obtain a smaller $d(u, v)$ – contradicting our choice of u, v. So π' contains $\le t$ vertices, i.e. $d(u, v) \le 2(t-1)r$; but since $u, v \in R$, this contradicts Line 6. ◁

So $d(j_1, j_2) \le 2(t-1)r$, i.e. by Line 6, $d(v, j_2) \le d(v, j_1) + d(j_1, j_2) \le 2tr + 2(t-1)r = (4t-2)r$. We have just showed that, $\forall v \in P, j \in R \cap P$, $d(v, j) \le (4t-2)r$. By Observation 14, this implies Property 3 for $\rho = (4t-2)$. ◀

▶ **Lemma 22.** *Algorithm 4 yields a* $(2^t, \mathsf{cov})$*-good partition.*

Proof. Consider $\mathcal{P}$, the output of Algorithm 4, and the child and R constructed alongside. Note that, since Line 8 only creates edges to `Roots`, and Line 9 updates `Roots` accordingly, (R, E) is indeed a forest.

Line 12 ensures that $\{\mathsf{child}(j)\}_{j \in R}$ is a partition of C. Line 14 ensures that this partition is a refinement of $\mathcal{P}$. Lines 6 and 11 construct child as desired, ensuring that $\forall j \in R$, $\forall v \in \mathsf{child}(j)$, $\mathsf{cov}_j \ge \mathsf{cov}_v$ and $\ell_j \ge \ell_v$. So Property 1 holds.

Now consider $P_1, P_2 \in \mathcal{P}$, $x_1 \in R \cap P_1$, $x_2 \in R \cap P_2$. Without loss of generality, suppose x_2 was added to R after x_1; if $d(x_1, x_2) \le 2r$, then by Lines 8 and 10, we would have $d(x_2, x_1) \in E$, i.e. x_1, x_2 would lie in the same connected component in $\mathcal{T}$. So by Lines 13-14, $P_1 = P_2$. This shows that Property 2 holds.

Algorithm 4 Finding a $(2^t, \mathsf{cov})$-good partition.

1: $U \leftarrow C$
2: $(R, E) \leftarrow (\emptyset, \emptyset)$ ▷ initializing an empty directed forest
3: $\forall v \in U$, $\text{height}(v) \leftarrow 0$ ▷ height in the forest; $\text{height}(v) = 0 \implies v \notin R$
4: $\texttt{Roots} \leftarrow \emptyset$ ▷ tracking roots in the forest
5: **while** $U \neq \emptyset$ **do**
6: $\quad j \leftarrow \text{argmax}_{v \in U} \mathsf{cov}_v$
7: $\quad R \leftarrow R \cup \{j\}$
8: $\quad E \leftarrow E \cup \left\{(j, j') : j' \in \texttt{Roots} \wedge d(j, j') \leq 2^{\text{height}(j')} r\right\}$
9: $\quad \texttt{Roots} \leftarrow (\texttt{Roots} \setminus \{j' : (j, j') \in E\}) \cup \{j\}$
10: $\quad \text{height}(j) \leftarrow 1 + \max_{(j,j') \in E} \text{height}(j')$ ▷ convention: max over $\emptyset$ is 0
11: $\quad \mathsf{child}(j) \leftarrow \left\{v \in U : d(v, j) \leq 2^{\text{height}(j)} r \wedge \ell_v \leq \ell_j\right\}$
12: $\quad U \leftarrow U \setminus \mathsf{child}(j)$
13: $\mathcal{T} \leftarrow$ connected components in the forest (R, E) ▷ each component induces a tree
14: $\mathcal{P} \leftarrow \{\cup_{j \in V} \mathsf{child}(j)\}_{V \in \mathcal{T}}$

Finally, note that

▷ Claim 23. $(j, j') \in E \implies \ell_j > \ell_{j'}$.

Proof. Since $(j, j') \in E$, we know that j' was added to R before j, and $d(j, j') \leq 2^{\text{height}(j')} r$. So if $\ell_j \leq \ell_{j'}$, then by Line 11, we would have $j \in \mathsf{child}(j')$, contradicting the fact that $j \in R$. ◁

Now fix $P \in \mathcal{P}$, and consider j_P which, by Observation 14, lies in $R \cap P$, and hence by Lines 13-14, $R \cap P$ induces a tree in (R, E). Claim 23 tells us that j_P is the root in this tree, and that $\text{height}(j_P) \leq t$. So by Line 8, for any $j \in R \cap P$, $d(j_P, j) \leq \left(2^{\text{height}(j_P)} - 2^{\text{height}(j)}\right) r \leq \left(2^t - 2^{\text{height}(j)}\right) r$. Now consider $v \in P : v \in \mathsf{child}(j)$ for a $j \in R \cap P$. Then $d(v, j) \leq 2^{\text{height}(j)}$, so $d(v, j_P) \leq d(v, j) + d(j, j_P) \leq \left(2^t - 2^{\text{height}(j)} + 2^{\text{height}(j)}\right) r = 2^t r$. Thus Property 3 holds for $\rho = 2^t$. ◀

5 Conclusion

In this paper, we have studied the Fault-tolerant k-Supplier with Outliers problem and presented a $(4t - 1)$-approximation when there are t distinct fault tolerances. While this gives the optimal 3-approximation for the uniform version of the problem (improving upon the recent result [20]), the parameter t could be as large as k. To obtain our result, we needed to resort to the powerful hammer of the round-or-cut schema, and indeed used a very strong LP relaxation. This was necessary since, as we saw in Section 3.1, natural LP relaxations and their strengthenings have unbounded integrality gaps. We also show a $\Omega(t)$-bottleneck to our approach (Appendix B), and this raises the intriguing question: are there $O(1)$-approximations for the FkSO problem? As noted in Section 1, the authors are not aware of clustering problems where the version without outliers has a constant approximation (as we saw in Section 2, FkS does), but the outlier version doesn't. Perhaps FkSO is such a candidate example. This also raises the question of designing inapproximability results for metric clustering problems, which has not been explored much. We leave all these as interesting avenues of further study.

References

1 Ravinder Ahuja, Aakarsha Chug, Shaurya Gupta, Pratyush Ahuja, and Shruti Kohli. Classification and clustering algorithms of machine learning with their applications. *Nature-inspired computation in data mining and machine learning*, pages 225–248, 2020. doi:10.1007/978-3-030-28553-1_11.

2 Hyung-Chan An, Mohit Singh, and Ola Svensson. LP-Based Algorithms for Capacitated Facility Location. In *Proc., IEEE Symposium on Foundations of Computer Science (FOCS)*, 2014. doi:10.1137/151002320.

3 Georg Anegg, Haris Angelidakis, Adam Kurpisz, and Rico Zenklusen. A technique for obtaining true approximations for k-center with covering constraints. *Math. Programming*, pages 1–25, 2022. doi:10.1007/s10107-021-01645-y.

4 Tanvi Bajpai, Deeparnab Chakrabarty, Chandra Chekuri, and Maryam Negahbani. Revisiting Priority k-Center: Fairness and Outliers. In *Proc., International Colloquium on Automata, Languages and Programming (ICALP)*, pages 21:1–21:20, 2021. doi:10.4230/LIPIcs.ICALP.2021.21.

5 Jaroslaw Byrka, Aravind Srinivasan, and Chaitanya Swamy. Fault-tolerant facility location: a randomized dependent LP-rounding algorithm. In *Proc., MPS Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 244–257, 2010. doi:10.1007/978-3-642-13036-6.

6 Deeparnab Chakrabarty, Prachi Goyal, and Ravishankar Krishnaswamy. The Non-Uniform k-Center Problem. *ACM Trans. on Algorithms (TALG)*, 2020. Preliminary version in ICALP 2016. doi:10.1145/3392720.

7 Deeparnab Chakrabarty and Maryam Negahbani. Generalized Center Problems with Outliers. *ACM Trans. on Algorithms (TALG)*, 2019. Prelim. version in ICALP 2018. doi:10.1145/3338513.

8 Deeparnab Chakrabarty and Maryam Negahbani. Robust k-center with two types of radii. *Math. Programming*, 197(2):991–1007, 2023. Special Issue for Proc. IPCO 2021. doi:10.1007/s10107-022-01799-3.

9 Deeparnab Chakrabarty, Maryam Negahbani, and Ankita Sarkar. Approximation Algorithms for Continuous Clustering and Facility Location Problems. In *Proc., European Symposium on Algorithms*, pages 33:1–33:15, 2022. doi:10.4230/LIPIcs.ESA.2022.33.

10 Moses Charikar, Samir Khuller, David M. Mount, and Giri Narasimhan. Algorithms for Facility Location Problems with Outliers. In *Proc., ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 642–651, 2001. URL: https://dl.acm.org/doi/abs/10.5555/365411.365555.

11 Shiva Chaudhuri, Naveen Garg, and Ramamoorthi Ravi. The p-neighbor k-center problem. *Inform. Process. Lett.*, 65(3):131–134, 1998. doi:10.1016/S0020-0190(97)00224-X.

12 Shiri Chechik and David Peleg. Robust fault tolerant uncapacitated facility location. *Theoretical Computer Science*, 543:9–23, 2014. Preliminary version appeared in STACS 2010. doi:10.1016/j.tcs.2014.05.013.

13 Vasek Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics*, 4(4):305–337, 1973. doi:10.1016/0012-365X(73)90167-2.

14 Shichuan Deng. Fault-Tolerant Center Problems with Robustness and Fairness. *CoRR*, abs/2011.00817v2, 2020. Citation specific to version 2; latest version is a significantly different work. arXiv:2011.00817v2.

15 Shichuan Deng. Constant approximation for fault-tolerant median problems via iterative rounding. *Operations Research Letters*, 50(4):384–390, 2022. doi:10.1016/j.orl.2022.05.002.

16 Ralph E Gomory. Some polyhedra related to combinatorial problems. *Linear Algebra Appl.*, 2(4):451–558, 1969. doi:10.1016/0024-3795(69)90017-2.

17 Sudipto Guha, Adam Meyerson, and Kamesh Munagala. A constant factor approximation algorithm for the fault-tolerant facility location problem. *Journal of Algorithms*, 48(2):429–440, 2003. Preliminary version appeared in SODA 2001. doi:10.1016/S0196-6774(03)00056-7.

18 MohammadTaghi Hajiaghayi, Wei Hu, Jian Li, Shi Li, and Barna Saha. A constant factor approximation algorithm for fault-tolerant k-median. *ACM Trans. on Algorithms (TALG)*, 12(3):1–19, 2016. doi:10.1145/2854153.

19 Dorit S. Hochbaum and David B. Shmoys. A unified approach to approximation algorithms for bottleneck problems. *Journal of the ACM*, 33(3):533–550, 1986. doi:10.1145/5925.5933.

20 Tanmay Inamdar and Kasturi Varadarajan. Fault tolerant clustering with outliers. In *Proc., Workshop on Approximation and Online Algorithms (WAOA)*, pages 188–201. Springer, 2020. doi:10.1007/978-3-030-39479-0_13.

21 Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999. doi:10.1145/331499.331504.

22 Kamal Jain and Vijay V Vazirani. An approximation algorithm for the fault tolerant metric facility location problem. *Algorithmica*, 38:433–439, 2004. Preliminary version appeared in APPROX 2000. doi:10.1007/s00453-003-1070-1.

23 Xinrui Jia, Lars Rohwedder, Kshiteej Sheth, and Ola Svensson. Towards Non-Uniform k-Center with Constant Types of Radii. In *Symposium on Simplicity in Algorithms (SOSA)*, pages 228–237. SIAM, 2022. doi:10.1137/1.9781611977066.16.

24 Christopher Jung, Sampath Kannan, and Neil Lutz. Service in Your Neighborhood: Fairness in Center Location. In *Proceedings, Foundations of Responsible Computing, FORC 2020*, volume 156, pages 5:1–5:15, 2020. doi:10.4230/LIPIcs.FORC.2020.5.

25 Samir Khuller, Robert Pless, and Yoram J Sussmann. Fault tolerant k-center problems. *Theoretical Computer Science*, 242(1-2):237–245, 2000. doi:10.1016/S0304-3975(98)00222-9.

26 Sven Oliver Krumke. On a generalization of the p-center problem. *Inform. Process. Lett.*, 56(2):67–71, 1995. doi:10.1016/0020-0190(95)00141-X.

27 Nirman Kumar and Benjamin Raichel. Fault Tolerant Clustering Revisited. In *Canadian Conference on Computational Geometry (CCCG)*, 2013. doi:10.48550/arXiv.1307.2520.

28 Richard CT Lee. Clustering analysis and its applications. In *Advances in Information Systems Science: Volume 8*, pages 169–292. Springer, 1981. doi:10.1007/978-1-4613-9883-7_4.

29 Yu Li, Dachuan Xu, Donglei Du, and Naihua Xiu. Improved approximation algorithms for the robust fault-tolerant facility location problem. *Inform. Process. Lett.*, 112(10):361–364, 2012. doi:10.1016/j.ipl.2012.02.004.

30 Chiun-Ming Liu. Clustering techniques for stock location and order-picking in a distribution center. *Comput. Oper. Res.*, 26(10-11):989–1002, 1999. doi:10.1016/S0305-0548(99)00026-X.

31 Sepideh Mahabadi and Ali Vakilian. (Individual) Fairness for k-Clustering. In *Proc., International Conference on Machine Learning (ICML)*, pages 7925–7935, 2020. URL: https://dl.acm.org/doi/abs/10.5555/3524938.3525549.

32 Boris Mirkin. *Mathematical classification and clustering*, volume 11. Springer Science & Business Media, 1996. doi:10.1007/978-1-4613-0457-9.

33 Ján Plesník. A heuristic for the p-center problems in graphs. *Discrete Applied Mathematics*, 17(3):263–268, 1987. doi:10.1016/0166-218X(87)90029-1.

34 Bartosz Rybicki and Jaroslaw Byrka. Improved approximation algorithm for fault-tolerant facility placement. In *Proc., Workshop on Approximation and Online Algorithms (WAOA)*, pages 59–70. Springer, 2015. doi:10.1007/978-3-319-18263-6_6.

35 Chinmay Sonar, Subhash Suri, and Jie Xue. Fault Tolerance in Euclidean Committee Selection. In Inge Li Gørtz, Martin Farach-Colton, Simon J. Puglisi, and Grzegorz Herman, editors, *31st Annual European Symposium on Algorithms (ESA 2023)*, volume 274 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 95:1–95:14, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ESA.2023.95.

36 Chaitanya Swamy and David B. Shmoys. Fault-Tolerant Facility Location. *ACM Trans. on Algorithms (TALG)*, 4(4), August 2008. doi:10.1145/1383369.1383382.

37 Shihong Xu and Hong Shen. The fault-tolerant facility allocation problem. In *Proc., Int. Symposium on Algorithms and Computation (ISAAC)*, pages 689–698. Springer, 2009. doi:10.1007/978-3-642-10631-6_70.

38 Li Yan and Marek Chrobak. Approximation algorithms for the fault-tolerant facility placement problem. *Inform. Process. Lett.*, 111(11):545–549, 2011. doi:10.1016/j.ipl.2011.03.005.

A Proof of Claim 7

Consider a feasible solution S^* that serves inliers T^*. Set

- $\forall v \in C$, $\mathsf{cov}_v = \mathbf{1}_{v \in T^*}$, and
- $\forall i \in F$, $y_i = \mathbf{1}_{i \in S^*}$.

These satisfy (WL1), (WL2), and (WL5) by construction. Now note that, for a $v \in T^*$, $N_{\ell_v}(v, F) \subseteq S$; so (WL3) is satisfied. Furthermore, for a $v \in C$, if $d_{\ell_v}(v, F) > r$ then $v \notin T^*$, satisfying (WL4).

Conversely, given an integral solution satisfying (WL1)-(WL5), we can construct $S^* = \{i \in F : y_i = 1\}$, and $T^* = \{v \in C : \mathsf{cov}_v = 1\}$. (WL2) implies $|S^*| \leq k$, and (WL1) implies $w(T^*) \geq W$. For any $v \in T^*$,

$$|S^* \cap B(v,r)| = \sum_{i \in F \cap B(v,r)} y_i \qquad \ldots \text{by construction of } S^*$$

$$\geq \ell \mathsf{cov}_v = \ell\,, \qquad \ldots \text{by (WL3) and construction of } T^*$$

so $d_{\ell_v}(v, S^*) \leq r$.

B Limiting Example for Good-Partition Rounding

In order to achieve a better approximation factor than $\Omega(t)$, we will need to move beyond the overall schema of using a good partition (Definition 13) to round solutions to (L1)-(L4). This can be seen via the following example, illustrated in Figure 3. Here $r = 1, n = t, m = 1$. C is the set $\{v_1, \cdots, v_t\}$, with each client v_a having fault-tolerance $\ell_{v_a} = a$. F is the union of t sets $\{F_a\}_{a=1}^t$, where $F_a = \{i_{a1}, i_{a2}, \ldots, i_{ak}\}$, for a total of tk facilities in F. Each client v_a has distance 2 to v_{a+1} and v_{a-1}, and distance 1 to each facility in F_a. Remaining distances are determined by making triangle inequalities tight.

Consider the following (cov, z) satisfying (L1)-(L4). We set $z_{F_a} = \frac{1}{aH_t}$ for each $a \in [t]$, where H_t is the t^{th} Harmonic number; and set all other z_S's to zero. This allows us to set $\mathsf{cov}_{v_a} = \frac{1}{aH_t}$ for each $a \in [t]$. Under this (cov, z), observe that $\forall v_a, v_b \in C$, $v_a \neq v_a \wedge \mathsf{cov}_a \geq \mathsf{cov}_b \implies \ell_a < \ell_b$; so Property 1 can only hold if all clients are in the same piece of the partition, i.e. $\mathcal{P} = \{C\}$. This means that a (ρ, cov)-good partition can only be attained for $\rho \geq 2(t-1)$, so upon applying Theorem 15, this approach attains a $(2t-1)$-approximation at best.

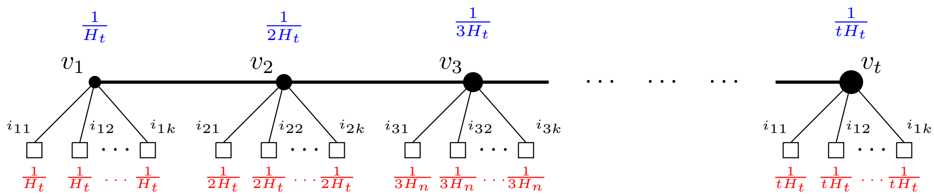

Figure 3 An example showing the limitations of good partitions, with a solution to (L1)-(L4) shown in **red** (z values) and blue (cov values). The thin "edges" represent distance 1, the thick "edges" represent distance 2, and all other distances are determined by making triangle inequalities tight. The fault-tolerances are $\ell_{v_1} = 1, \ell_{v_2} = 2, \ldots, \ell_{v_t} = t$.