Journal of Machine Learning Research 20 (2019) 1-33 Submitted 10/18; Revised 6/19; Published 8/19

Approximation algorithms for stochastic clustering

David G. Harris DAVIDGHARRIS29@GMAIL.COM
Department of Computer Science, University of Maryland
College Park, MD 20742

Shi Li SHIL@BUFFALO.EDU
University at Buffalo, Buffalo, NY

Thomas Pensyl TPENSYL@BANDWIDTH.COM
Bandwidth, Inc. Raleigh, NC

Aravind Srinivasan SRIN@CS.UMD.EDU
Department of Computer Science and Institute for Advanced Computer Studies
University of Maryland, College Park, MD 20742

Khoa Trinh KHOATRINH@GOOGLE.COM
Google, Mountain View, CA 94043

Editor: Maya Gupta

Abstract

We consider stochastic settings for clustering, and develop provably-good approximation algorithms for a number of these notions. These algorithms yield better approximation ratios compared to the usual deterministic clustering setting. Additionally, they offer a number of advantages including clustering which is fairer and has better long-term behavior for each user. In particular, they ensure that *every user* is guaranteed to get good service (on average). We also complement some of these with impossibility results.

Keywords: clustering, k-center, k-median, lottery, approximation algorithms

This is an extended version of a paper that appeared in the *Conference on Neural Information Processing Systems* (NeurIPS) 2018. Research supported in part by NSF Awards CNS-1010789, CCF-1422569, CCF-1749864, CCF-1566356, CCF-1717134, a gift from Google, Inc., and by research awards from Adobe, Inc. and Amazon, Inc.

1. Introduction

Clustering is a fundamental problem in machine learning and data science. A general clustering task is to partition the given datapoints such that points inside the same cluster are "similar" to each other. More formally, consider a set of datapoints $\mathcal{C}$ and a set of "potential cluster centers" $\mathcal{F}$, with a metric d on $\mathcal{C} \cup \mathcal{F}$. We define $n := |\mathcal{C} \cup \mathcal{F}|$. Given any set $\mathcal{S} \subseteq \mathcal{F}$, each $j \in \mathcal{C}$ is associated with the key statistic $d(j, \mathcal{S}) = \min_{i \in \mathcal{S}} d(i, j)$. The typical clustering task is to select a set $\mathcal{S} \subseteq \mathcal{F}$ which has a small size and which minimizes the values of $d(j, \mathcal{S})$.

The size of the set $\mathcal{S}$ is often fixed to a value k, and we typically "boil down" the large collection of values $d(j, \mathcal{S})$ into a single overall objective function. A variety of objective functions and assumptions on sets $\mathcal{C}$ and $\mathcal{F}$ are used. The most popular problems include [1]

- the k-center problem: minimize the value $\max_{j \in \mathcal{C}} d(j, \mathcal{S})$ given that $\mathcal{F} = \mathcal{C}$.
- the k-supplier problem: minimize the value $\max_{j \in \mathcal{C}} d(j, \mathcal{S})$ (where $\mathcal{F}$ and $\mathcal{C}$ may be unrelated);
- the k-median problem: minimize the summed value $\sum_{j \in \mathcal{C}} d(j, \mathcal{S})$; and
- the k-means problem: minimize the summed square value $\sum_{j \in \mathcal{C}} d(j, \mathcal{S})^2$.

An important special case is when $\mathcal{C} = \mathcal{F}$ (e.g. the k-center problem); since this often occurs in the context of data clustering, we refer to this as the **self-contained clustering (SCC)** setting.

These classic NP-hard problems have been studied intensively for the past few decades. There is an alternative interpretation from the viewpoint of operations research: the sets $\mathcal{F}$ and $\mathcal{C}$ can be thought of as "facilities" and "clients", respectively. We say that $i \in \mathcal{F}$ is *open* if i is placed into the solution set $\mathcal{S}$. For a set $\mathcal{S} \subseteq \mathcal{F}$ of open facilities, $d(j, \mathcal{S})$ can then be interpreted as the connection cost of client j. This terminology has historically been used for clustering problems, and we adopt it throughout for consistency. However, our focus is on the case in which $\mathcal{C}$ and $\mathcal{F}$ are arbitrary abstract sets in the data-clustering setting.

Since these problems are NP-hard, much effort has been paid on algorithms with "small" provable *approximation ratios/guarantees*: i.e., polynomial-time algorithms that produce solutions of cost at most α times the optimal. The current best-known approximation ratio for k-median is 2.675 by Byrka et al. (2017) and it is NP-hard to approximate this problem to within a factor of $1 + 2/e \approx 1.735$ (Jain et al., 2002). The recent breakthrough by Ahmadian et al. (2017) gives a 6.357-approximation algorithm for k-means, improving on the previous approximation guarantee of $9 + \epsilon$ based on local search (Kanungo et al., 2004). Finally, the k-supplier problem is "easier" than both k-median and k-means in the sense that a simple 3-approximation algorithm (Hochbaum and Shmoys, 1986) is known, as is a 2-approximation for k-center problem: we cannot do better than these approximation ratios unless P = NP (Hochbaum and Shmoys, 1986).

While optimal approximation algorithms for the center-type problems are well-known, one can easily demonstrate instances where such algorithms return a worst-possible solution: (i) all clusters have the same worst-possible radius ($2T$ for k-center and $3T$ for k-supplier where T is the optimal radius) and (ii) almost all data points are on the circumference of the resulting clusters. Although it is NP-hard to improve these approximation ratios, our new randomized algorithms provide significantly better "per-point" guarantees. For example, we achieve a new "per-point" guarantee $\mathbf{E}[d(j, \mathcal{S})] \leq (1 + 2/e)T \approx 1.736T$, while respecting the usual guarantee $d(j, \mathcal{S}) \leq 3T$ with probability one. *Thus, while maintaining good global quality with probability one, we also provide superior stochastic guarantees for each user.*

1. In the original version of the k-means problem, $\mathcal{C}$ is a subset of $\mathbb{R}^\ell$ and $\mathcal{F} = \mathbb{R}^\ell$ and d is the Euclidean metric. By standard discretization techniques (see, e.g., Matoušek (2000); Feldman et al. (2007)), the set $\mathcal{F}$ can be reduced to a polynomially-bounded set with only a small loss in the value of the objective function.

The general problem we study in this paper is to develop approximation algorithms for center-type problems where $\mathcal{S}$ is drawn from a probability distribution over k-element subsets of $\mathcal{F}$; we refer to these as *k-lotteries.* We aim to construct a k-lottery Ω achieving certain guarantees on the distributional properties of $d(j, \mathcal{S})$. The classical k-center problem can be viewed as the special case where the distribution Ω is deterministic, that is, it is supported on a single point. Our goal is to find an *approximating distribution* $\tilde{\Omega}$ which matches the *target distribution* Ω as closely as possible for each client j.

Stochastic solutions can circumvent the approximation hardness of a number of classical center-type problems. There are a number of additional applications where stochasticity can be beneficial. We summarize three here: smoothing the integrality constraints of clustering, solving repeated problem instances, and achieving fair solutions.

Stochasticity as interpolation. In practice, *robustness* of the solution is often more important than achieving the absolute optimal value for the objective function. One potential problem with the (deterministic) center measure is that it can be highly non-robust. As an extreme example, consider k-center with k points, each at distance 1 from each other. This clearly has value 0 (choosing $\mathcal{S} = \mathcal{C}$). However, if a single new point at distance 1 to all other points is added, then the solution jumps to 1. Stochasticity alleviates this discontinuity: by choosing k facilities uniformly at random among the full set of $k + 1$, we can ensure that $\mathbf{E}[d(j, \mathcal{S})] = \frac{1}{k+1}$ for every point j, a much smoother transition.

Repeated clustering problems. Consider clustering problems where the choice of $\mathcal{S}$ can be changed periodically: e.g., $\mathcal{S}$ could be the set of k locations in the cloud chosen by a service-provider. This set $\mathcal{S}$ can be shuffled periodically in a manner transparent to end-users. For any user $j \in \mathcal{C}$, the statistic $d(j, \mathcal{S})$ represents the latency of the service j receives (from its closest service-point in $\mathcal{S}$). If we aim for a fair or minmax service allocation, then our k-center stochastic approximation results ensure that, with high probability, *every client j* gets long-term average service of at most around $1.736T$. The average here is taken over the periodic re-provisioning of $\mathcal{S}$. (Furthermore, we have the risk-avoidance guarantee that in no individual provisioning of $\mathcal{S}$ will any client have service greater than $3T$.)

Fairness in clustering. The classical clustering problems combine the needs of many different points (elements of $\mathcal{C}$) into one metric. However, clustering (and indeed many other ML problems) are increasingly driven by inputs from parties with diverse interests. Fairness in these contexts has taken on greater importance in the current environment where decisions are increasingly made by algorithms and machine learning. Some examples of recent concerns include the accusations of, and fixes for, possible racial bias in Airbnb rentals (Badger, 2016) and the finding that setting the gender to "female" in Google's *Ad Settings* resulted in getting fewer ads for high-paying jobs (Datta et al., 2015). Starting with older work such as Schulman et al. (1999), there have been highly-publicized works on bias in allocating scarce resources – e.g., racial discrimination in hiring applicants who have very similar resumés (Bertrand and Mullainathan, 2004). Additional work discusses the possibility of bias in electronic marketplaces, whether human-mediated or not (Ayres et al., 2015; Badger, 2016).

A fair allocation should provide good service guarantees *to each user individually.* In data clustering settings where a user corresponds to a datapoint, this means that every point $j \in \mathcal{C}$ should be guaranteed a good value of $d(j, \mathcal{S})$. This is essentially the goal of

k-center type problems, but the stochastic setting broadens the meaning of good per-user service.

Consider the following scenarios. Each user, either explicitly or implicitly, submits their data (corresponding to a point in $\mathcal{C}$) to an aggregator such as an e-commerce site. A small number k of users are then chosen as "influencer" nodes; for instance, the aggregator may give them a free product sample to influence the whole population in aggregate, as in Kempe et al. (2015), or the aggregator may use them as a sparse "sketch", so that each user gets relevant recommendations from a influencer which is similar to them. Each point j would like to be in a cluster that is "high quality" *from its perspective*, with $d(j, \mathcal{S})$ being a good proxy for such quality. Indeed, there is increasing emphasis on the fact that organizations monetize their user data, and that users need to be compensated for this (see, e.g., Lanier (2014); Ibarra et al. (2018)). This is a transition from viewing data as capital to viewing *data as labor*. A concrete way for users (i.e., the data points $j \in \mathcal{C}$) to be compensated in our context is for each user to get a guarantee on their solution quality: i.e., bounds on $d(j, \mathcal{S})$.

1.1. Our contributions and overview

In Section 2, we encounter the first clustering problem which we refer to as *chance k-coverage*: namely, every client j has a distance demand r_j and probability demand p_j, and we wish to find a distribution satisfying $\Pr[d(j, \mathcal{S}) \leq r_j] \geq p_j$. We show how to obtain an approximation algorithm to find an approximating distribution $\tilde{\Omega}$ with[2]

$$\Pr_{\mathcal{S} \sim \tilde{\Omega}}[d(j, \mathcal{S}) \leq 9r_j] \geq p_j.$$

In a number of special cases, such as when all the values of p_j or r_j are the same, the distance factor 9 can be improved to 3, *which is optimal*; it is an interesting question to determine whether this factor can also be improved in the general case.

In Section 3, we consider a special case of chance k-coverage, in which $p_j = 1$ for all clients j. This is equivalent to the classical (deterministic) k-supplier problem. Allowing the approximating distribution $\tilde{\Omega}$ to be stochastic yields significantly better distance guarantees than are possible for k-supplier or k-center. For instance, we find an approximating distribution $\tilde{\Omega}$ with

$$\forall j \in \mathcal{C} \qquad \mathbf{E}_{\mathcal{S} \sim \tilde{\Omega}}[d(j, \mathcal{S})] \leq 1.592T \text{ and } \Pr[d(j, \mathcal{S}) \leq 3T] = 1$$

where T is the optimal solution to the (deterministic) k-center problem. By contrast, deterministic polynomial-time algorithms cannot guarantee $d(j, \mathcal{S}) < 2T$ for all j, unless P = NP (Hochbaum and Shmoys, 1986).

In Section 4, we show a variety of lower bounds on the approximation factors achievable by efficient algorithms (assuming P $\neq$ NP). For instance, we show that our approximation algorithm for chance k-coverage with equal p_j or r_j has the optimal distance approximation factor 3, that our approximation algorithm for k-supplier has optimal approximation factor $1 + 2/e$, and that the approximation factor 1.592 for k-center cannot be improved below $1 + 1/e$.

2. Notation such as "$S \sim \tilde{\Omega}$" indicates that the random set S is drawn from the distribution $\tilde{\Omega}$.

In Section 5, we consider a different type of stochastic approximation problem based on expected distances: namely, every client has a demand t_j, and we seek a k-lottery Ω with $\mathbf{E}[d(j,\mathcal{S})] \leq t_j$. We show that we can leverage *any given* α-approximation algorithm for k-median to produce a k-lottery $\tilde{\Omega}$ with $\mathbf{E}[d(j,\mathcal{S})] \leq \alpha t_j$. (Recall that the current-best α here is 2.675 as shown in Byrka et al. (2017).)

In Section 6, we consider the converse problem to Section 3: if we are given a k-lottery Ω with $\mathbf{E}[d(j,\mathcal{S})] \leq t_j$, can we produce a single deterministic set $\mathcal{S}$ so that $d(j,\mathcal{S}) \approx t_j$ and $|\mathcal{S}| \approx k$? We refer to this as a *determinization* of Ω. We show a variety of determinization algorithms. For instance, we are able to find a set $\mathcal{S}$ with $|\mathcal{S}| \leq 3k$ and $d(j,\mathcal{S}) \leq 3t_j$. We also show a number of nearly-matching lower bounds.

1.2. Related Work

With algorithms increasingly running our world, there has been substantial recent interest on incorporating fairness systematically into algorithms and machine learning. One important notion is *disparate impact*: in addition to requiring that *protected attributes* such as gender or race not be used (explicitly) in decisions, this asks that decisions not be disproportionately different for diverse protected classes (Feldman et al., 2015). This is developed further in the context of clustering in the work of Chierichetti et al. (2017). Such notions of *group fairness* are considered along with *individual fairness* – treating similar individuals similarly – in Zemel et al. (2013). See Dwork et al. (2012) for earlier work that developed foundations and connections for several such notions of fairness.

In the context of the location and sizing of services, there have been several studies indicating that proactive on-site provision of healthcare improves health outcomes significantly: e.g., mobile mammography for older women (Reuben et al., 2002), mobile healthcare for reproductive health in immigrant women (Guruge et al., 2010), and the use of a community mobile-health van for increased access to prenatal care (Edgerley et al., 2007). Studies also indicate the impact of distance to the closest facility on health outcomes: see, e.g., McCarthy et al. (2007); Mooney et al. (2000); Schmitt et al. (2003). Such works naturally suggest tradeoffs between resource allocation (provision of such services, including sizing – e.g., the number k of centers) and expected health outcomes.

While much analysis for facility-location problems has focused on the static case, other works have examined a similar lottery model for center-type problems. Harris et al. (2019, 2017) analyzed models similar to chance k-coverage and minimization of $\mathbf{E}[d(j,\mathcal{S})]$, but applied to knapsack center and matroid center problems; they also considered robust versions (in which a small subset of clients may be denied service). While the overall model was similar to the ones we explore here, the techniques are somewhat different. Furthermore, these works focus only on the case where the target distribution is itself deterministic.

Similar stochastic approximation guarantees have appeared in the context of approximation algorithms for static problems, particularly k-median problems. Charikar and Li (2012) discussed a randomized procedure for converting a linear-programming relaxation in which a client has *fractional* distance t_j, into a distribution Ω satisfying $\mathbf{E}_{\mathcal{S}\sim\Omega}[d(j,\mathcal{S})] \leq 3.25t_j$. This property can be used, among other things, to achieve a 3.25-approximation for k-median. However, many other randomized rounding algorithms for k-median only seek to preserve the *aggregate* value $\sum_j \mathbf{E}[d(j,\mathcal{S})]$, without our type of per-point guarantee.

We also contrast our approach with a different stochastic k-center problem considered in works such as Huang and Li (2017); Alipour and Jafari (2018). These consider a model with a fixed, deterministic set $\mathcal{S}$ of open facilities, while the client set is determined stochastically; this model is almost precisely opposite to ours.

1.3. Publicly Verifying the Distributions

Our approximation algorithms will have the following structure: given some target distribution Ω, we construct a randomized procedure $\mathcal{A}$ which returns some random set $\mathcal{S}$ with good probabilistic guarantees matching Ω. Thus the algorithm $\mathcal{A}$ is *itself* the approximating distribution $\tilde{\Omega}$.

In a number of cases, we can convert the randomized algorithm $\mathcal{A}$ into a distribution $\tilde{\Omega}$ which has a sparse support (set of points to which it assigns nonzero probability), and which can be enumerated directly. This may cause a small loss in approximation ratio. The distribution $\tilde{\Omega}$ can be publicly verified, and the users can then draw from $\tilde{\Omega}$ as desired.

Recall that one of our main motivations is fairness in clustering; the ability for the users to verify that they are being treated fairly in a stochastic sense (although not necessarily in any one particular run of the algorithm) is particularly important.

1.4. Notation

We define $\binom{\mathcal{F}}{k}$ to be the collection of k-element subsets of $\mathcal{F}$. We assume throughout that $\mathcal{F}$ can be made arbitrarily large by duplicating its elements; thus, whenever we have an expression like $\binom{\mathcal{F}}{k}$, we assume without loss of generality that $|\mathcal{F}| \geq k$.

We will let $[t]$ denote the set $\{1, 2, \ldots, t\}$. For any vector $a = (a_1, \ldots, a_t)$ and a subset $X \subseteq [t]$, we write $a(X)$ as shorthand for $\sum_{i \in X} a_i$.

We use the Iverson notation throughout, so that for any Boolean predicate $\mathcal{P}$ we let $[[\mathcal{P}]]$ be equal to one if $\mathcal{P}$ is true and zero otherwise.

For a real number $q \in [0, 1]$, we define $\overline{q} = 1 - q$.

Given any $j \in \mathcal{C}$ and any real number $r \geq 0$, we define the ball $B(j, r) = \{i \in \mathcal{F} \mid d(i, j) \leq r\}$. We let $\theta(j)$ be the distance from j to the nearest facility, and V_j be the facility closest to j, i.e. $d(j, V_j) = d(j, \mathcal{F}) = \theta(j)$. Note that in the SCC setting we have $V_j = j$ and $\theta(j) = 0$.

For a solution set $\mathcal{S}$, we say that $j \in \mathcal{C}$ is *matched* to $i \in \mathcal{S}$ if i is the closest facility of $\mathcal{S}$ to j; if there are multiple closest facilities, we take i to be one with least index.

1.5. Some Useful Subroutines

We will use two basic subroutines repeatedly: *dependent rounding* and *greedy clustering*.

In dependent rounding, we aim to preserve certain marginal distributions and negative correlation properties while satisfying some constraints with probability one. Our algorithms use a dependent-rounding algorithm from Srinivasan (2001), which we summarize as follows:

Proposition 1 *There exists a randomized polynomial-time algorithm* DepRound(y) *which takes as input a vector $y \in [0, 1]^n$, and outputs a random set $Y \subseteq [n]$ with the following properties:*

(P1) $\Pr[i \in Y] = y_i$, *for all* $i \in [n]$,

(P2) $\lfloor \sum_{i=1}^n y_i \rfloor \le |Y| \le \lceil \sum_{i=1}^n y_i \rceil$ *with probability one,*

(P3) $\Pr[Y \cap S = \emptyset] \le \prod_{i \in S}(1 - y_i)$ *for all* $S \subseteq [n]$.

We adopt the following additional convention: suppose $(y_1, \ldots, y_n) \in [0,1]^n$ and $S \subseteq [n]$; we then define $\textsc{DepRound}(y, S) \subseteq S$ to be $\textsc{DepRound}(x)$, for the vector x defined by $x_i = y_i[[i \in S]]$.

The greedy clustering procedure takes an input a set of weights w_j and sets $F_j \subseteq \mathcal{F}$ for every client $j \in \mathcal{C}$, and executes the following procedure:

Algorithm 1 $\textsc{GreedyCluster}(F, w)$

Sort $\mathcal{C}$ as $\mathcal{C} = \{j_1, j_2, \ldots, j_\ell\}$ where $w_{j_1} \le w_{j_2} \le \cdots \le w_{j_\ell}$.
Initialize $C' = \emptyset$
for $t = 1, \ldots, \ell$ **do**
    if $F_{j_t} \cap F_{j'} = \emptyset$ for all $j' \in C'$ **then** update $C' \leftarrow C' \cup \{j_t\}$
Return C'

Observation 2 *If* $C' = \textsc{GreedyCluster}(F, w)$ *then for any* $j \in \mathcal{C}$ *there is* $z \in C'$ *with* $w_z \le w_j$ *and* $F_z \cap F_j \neq \emptyset$.

2. The Chance k-coverage Problem

In this section, we consider a scenario we refer to as the *chance k-coverage problem*: every point $j \in \mathcal{C}$ has demand parameters p_j, r_j, and we wish to find a k-lottery Ω such that

$$\Pr_{\mathcal{S} \sim \Omega}[d(j, \mathcal{S}) \le r_j] \ge p_j. \tag{1}$$

If a k-lottery satisfying (1) exists, we say that parameters p_j, r_j are *feasible.* We refer to the special case wherein every client j has a common value $p_j = p$ and a common value $r_j = r$, as *homogeneous.* Homogeneous instances naturally correspond to fair allocations, for example, k-supplier is a special case of the homogeneous chance k-coverage problem, in which $p_j = 1$ and r_j is equal to the optimal k-supplier radius.

Our approximation algorithms for this problem will be based on a linear programming (LP) relaxation which we denote $\mathcal{P}_{\text{chance}}$. It has fractional variables b_i, where i ranges over $\mathcal{F}$ (b_i represents the probability of opening facility i), and is defined by the following constraints:

(B1) $\sum_{i \in B(j, r_j)} b_i \ge p_j$ for all $j \in \mathcal{C}$,

(B2) $b(\mathcal{F}) = k$,

(B3) $b_i \in [0,1]$ for all $i \in \mathcal{F}$.

Proposition 3 *If parameters* p, r *are feasible, then* $\mathcal{P}_{chance}$ *is nonempty.*

Proof Consider a distribution Ω satisfying (1). For each $i \in \mathcal{F}$, set $b_i = \Pr_{\mathcal{S}\sim\Omega}[i \in \mathcal{S}]$. For $j \in \mathcal{C}$ we have $p_j = \Pr\left[\bigvee_{i\in B(j,r_j)} i \in \mathcal{S}\right] \leq \sum_{i\in B(j,r_j)} \Pr[i \in S] = \sum_{i\in B(j,r_j)} b_i$ and thus (B1) is satisfied. We have $k = \mathbf{E}[|\mathcal{S}|] = \sum_{i\in\mathcal{F}} \Pr[i \in S] = b(\mathcal{F})$ and so (B2) is satisfied. (B3) is clear, so we have demonstrated a point in $\mathcal{P}_{\text{chance}}$. ■

For the remainder of this section, we assume we have a vector $b \in \mathcal{P}_{\text{chance}}$ and focus on how to round it to an integral solution. By a standard facility-splitting step, we also generate, for every $j \in \mathcal{C}$, a set $F_j \subseteq B(j, r_j)$ with $b(F_j) = p_j$. We refer to this set F_j as a *cluster*. In the SCC setting, it will also be convenient to ensure that $j \in F_j$ as long as $b_j \neq 0$.

As we show in Section 4, any approximation algorithm must either significantly give up a guarantee on the distance, or probability (or both). Our first result is an approximation algorithm which respects the distance guarantee exactly, with constant-factor loss to the probability guarantee:

Theorem 4 *If p, r is feasible then one may efficiently construct a k-lottery Ω satisfying*

$$\Pr_{\mathcal{S}\sim\Omega}[d(j, \mathcal{S}) \leq r_j] \geq (1 - 1/e)p_j.$$

Proof Let $b \in \mathcal{P}_{\text{chance}}$ and set $\mathcal{S} = \textsc{DepRound}(b)$. This satisfies $|\mathcal{S}| \leq \lceil \sum_{i=1}^{n} b_i \rceil \leq \lceil k \rceil = k$ as desired. Each $j \in \mathcal{C}$ has

$$\Pr[\mathcal{S} \cap F_j = \emptyset] \leq \prod_{i\in F_j} (1 - b_i) \leq \prod_{i\in F_j} e^{-b_i} = e^{-b(F_i)} = e^{-p_j}.$$

and then simple analysis shows that

$$\Pr[d(j, \mathcal{S}) \leq r_j] \geq \Pr[\mathcal{S} \cap F_j \neq \emptyset] \geq 1 - e^{-p_j} \geq (1 - 1/e)p_j.$$ ■

As we will later show in Theorem 20, this approximation constant $1 - 1/e$ is optimal.

We next turn to preserving the probability guarantee exactly with some loss to distance guarantee. As a warm-up exercise, let us consider the special case of "half-homogeneous" problem instances: all the values of p_j are the same, or all the values of r_j are the same. A similar algorithm works both these cases: we first select a set of clusters according to some greedy order, and then open a single item from each cluster. We summarize this as follows:

Algorithm 2 Rounding algorithm for half-homogeneous chance k-coverage

1: Set $C' = \textsc{GreedyCluster}(F_j, w_j)$
2: Set $Y = \textsc{DepRound}(p, C')$
3: Form solution set $\mathcal{S} = \{V_j \mid j \in Y\}$.

Algorithm 2 opens at most k facilities, as the dependent rounding step ensures that $|Y| \leq \lceil \sum_{j\in C'} p_j \rceil = \lceil \sum_{j\in C'} b(F_j) \rceil \leq \lceil \sum_{i\in\mathcal{F}} b_i \rceil \leq k$. The only difference between the two cases is the choice of weighting function w_j for the greedy cluster selection.

Proposition 5 *1. Suppose that p_j is the same for every $j \in \mathcal{C}$. Then using the weighting function $w_j = r_j$ ensures that every $j \in \mathcal{C}$ satisfies $\Pr[d(j, \mathcal{S}) \leq 3r_j] \geq p_j$. Furthermore, in the SCC setting, it satisfies $\Pr[d(j, \mathcal{S}) \leq 2r_j] \geq p_j$.*

2. *Suppose r_j is the same for every $j \in \mathcal{C}$. Then using the weighting function $w_j = 1 - p_j$ ensures that every $j \in \mathcal{C}$ satisfies $\Pr[d(j, \mathcal{S}) \leq 3r_j] \geq p_j$. Furthermore, in the SCC setting, it satisfies $\Pr[d(j, \mathcal{S}) \leq 2r_j] \geq p_j$.*

Proof Let $j \in \mathcal{C}$. By Observation 2 there is $z \in C'$ with $w_z \leq w_j$ and $F_j \cap F_z \neq \emptyset$. In either of the two cases, this implies that $p_z \geq p_j$ and $r_z \leq r_j$.

Letting $i \in F_j \cap F_z$ gives $d(j, z) \leq d(j, i) + d(z, i) \leq r_j + r_z \leq 2r_j$. This $z \in C'$ survives to Y with probability $p_z \geq p_j$, and in that case we have $d(z, \mathcal{S}) = \theta(z)$. In the SCC setting, this means that $d(z, \mathcal{S}) = 0$; in the general setting, we have $\theta(z) \leq r_z \leq r_j$. ■

2.1. Approximating the General Case

We next show how to satisfy the probability constraint exactly for the general case of chance k-coverage, with a constant-factor loss in the distance guarantee. Namely, we will find a probability distribution with

$$\Pr_{\mathcal{S} \sim \Omega}[d(j, \mathcal{S}) \leq 9r_j] \geq p_j.$$

The algorithm is based on iterated rounding, in which the entries of b go through an unbiased random walk until b becomes integral (and, thus corresponds to a solution set $\mathcal{S}$). Because the walk is unbiased, the probability of serving a client at the end is equal to the fractional probability of serving a client, which will be at least p_j. In order for this process to make progress, the number of active variables must be greater than the number of active constraints. We ensure this by periodically identifying and discarding clients which will be automatically served by serving other clients. This is similar to a method of Krishnaswamy et al. (2018), which also uses iterative rounding for (deterministic) approximations to k-median with outliers and k-means with outliers.

The sets F_j will remain fixed during this procedure. We will maintain a vector $b \in [0, 1]^{\mathcal{F}}$ and maintain two sets C_{tight} and C_{slack} with the following properties:

(C1) $C_{\text{tight}} \cap C_{\text{slack}} = \emptyset$.

(C2) For all $j, j' \in C_{\text{tight}}$, we have $F_j \cap F_{j'} = \emptyset$

(C3) Every $j \in C_{\text{tight}}$ has $b(F_j) = 1$,

(C4) Every $j \in C_{\text{slack}}$ has $b(F_j) \leq 1$.

(C5) We have $b(\bigcup_{j \in C_{\text{tight}} \cup C_{\text{slack}}} F_j) \leq k$

Given our initial solution b for $\mathcal{P}_{\text{chance}}$, setting $C_{\text{tight}} = \emptyset, C_{\text{slack}} = \mathcal{C}$ will satisfy criteria (C1)–(C5); note that (C4) holds as $b(F_j) = p_j \leq 1$ for all $j \in \mathcal{C}$.

Proposition 6 *Given any vector $b \in [0, 1]^{\mathcal{F}}$ satisfying constraints (C1)—(C5) with $C_{slack} \neq \emptyset$, it is possible to generate a random vector $b' \in [0, 1]^{\mathcal{F}}$ such that $\mathbf{E}[b'] = b$, and with probability one b' satisfies constraints (C1) — (C5) as well as having some $j \in C_{slack}$ with $b'(F_j) \in \{0, 1\}$.*

Proof We will show that any basic solution $b \in [0,1]^{\mathcal{F}}$ to the constraints (C1)—(C5) with $C_{\text{slack}} \neq \emptyset$ must satisfy the condition that $b(F_j) \in \{0,1\}$ for some $j \in C_{\text{slack}}$. To obtain the stated result, we simply modify b until it becomes basic by performing an unbiased walk in the nullspace of the tight constraints.

So consider a basic solution b. Define $A = \bigcup_{j \in C_{\text{tight}}} F_j$ and $B = \bigcup_{j \in C_{\text{slack}}} F_j$. We assume that $b(F_j) \in (0,1)$ for all $j \in C_{\text{slack}}$, as otherwise we are done.

First, suppose that $b(A \cap B) > 0$. So there must be some pair $j \in C_{\text{slack}}, j' \in C_{\text{tight}}$ with $i \in F_j \cap F_{j'}$ such that $b_i > 0$. Since $b(F_{j'}) = 1$, there must be some other $i' \in F_{j'}$ with $b_{i'} > 0$. Consider modifying b by incrementing b_i by $\pm\epsilon$ and decrementing $b_{i'}$ by $\pm\epsilon$, for some sufficiently small ϵ. Constraint (C5) is preserved. Since $F_{j'} \cap F_{j''} = \emptyset$ for all $j'' \in C_{\text{tight}}$, constraint (C3) is preserved. Since the (C4) constraints are slack, then for ϵ sufficiently small they are preserved as well. This contradicts that b is basic.

Next, suppose that $b(A \cap B) = 0$ and $b(A \cup B) < k$ strictly. Let $j \in C_{\text{slack}}$ and $i \in F_j$ with $b_i > 0$; if we change b_i by $\pm\epsilon$ for sufficiently small ϵ, this preserves (C4) and (C5); furthermore, since $i \notin A$, it preserves (C3) as well. So again b cannot be basic.

Finally, suppose that $b(A \cap B) = 0$ and $b(A \cup B) = k$. So $b(B) = k - |A|$ and $b(B) > 0$ as $C_{\text{slack}} \neq \emptyset$. Therefore, there must be at least two elements $i, i' \in B$ such that $b_i > 0, b_{i'} > 0$. If we increment b_i by $\pm\epsilon$ while decrementing $b_{i'}$ by $\pm\epsilon$, this again preserves all the constraints for ϵ sufficiently small, contradicting that b is basic. ■

We can now describe our iterative rounding algorithm, Algorithm 3.

Algorithm 3 Iterative rounding algorithm for chance k-coverage

Find a fractional solution b to $\mathcal{P}_{\text{chance}}$ and form the corresponding sets F_j.
Initialize $C_{\text{tight}} = \emptyset, C_{\text{slack}} = \mathcal{C}$
while $C_{\text{slack}} \neq \emptyset$ **do**
  Draw a fractional solution b' with $\mathbf{E}[b'] = b$ according to Proposition 6.
  Select some $v \in C_{\text{slack}}$ with $b'(F_v) \in \{0,1\}$.
  Update $C_{\text{slack}} \leftarrow C_{\text{slack}} - \{v\}$
  if $b'(F_v) = 1$ **then**
    Update $C_{\text{tight}} \leftarrow C_{\text{tight}} \cup \{v\}$
    if there is any $z \in C_{\text{tight}} \cup C_{\text{slack}}$ such that $r_z \geq r_v/2$ and $F_z \cap F_v \neq \emptyset$ **then**
      Update $C_{\text{tight}} \leftarrow C_{\text{tight}} - \{z\}, C_{\text{slack}} \leftarrow C_{\text{slack}} - \{z\}$
  Update $b \leftarrow b'$.
For each $j \in C_{\text{tight}}$, open an arbitrary center in F_j.

To analyze this algorithm, define $C^t_{\text{tight}}, C^t_{\text{slack}}, b^t$ to be the values of the relevant variables at iteration t. Since every step removes at least one point from C_{slack}, this process must terminate in $T \leq n$ iterations. We will write b^{t+1} to refer to the random value b' chosen at step (4) of iteration t, and v^t denote the choice of $v \in C_{\text{slack}}$ used step in step (5) of iteration t.

Proposition 7 *The vector b^t satisfies constraints (C1) — (C5) for all times $t = 1, \ldots, T$.*

Proof The vector b^0 does so since b satisfies $\mathcal{P}_{\text{chance}}$. Proposition 6 ensures that step (4) does not affect this. Also, removing points from C_{tight} or C_{slack} at step (6) or (1) will not violate these constraints.

Let us check that adding v^t to C_{tight} will not violate the constraints. This step only occurs if $b^{t+1}(v^t) = 1$, and so (C3) is preserved. Since we only move v^t from C_{slack} to C_{tight}, constraints (C1) and (C5) are preserved. Finally, to show that (C2) is preserved, suppose that $F_{v^t} \cap F_{v^s} \neq \emptyset$ for some other v^s which was added to C_{tight} at time $s < t$. If $r_{v^t} \geq r_{v^s}$, then step (10) would have removed v^t from C_{slack}, making it impossible to enter C^t_{tight}. Thus, $r_{v^t} \leq r_{v^s}$; this means that when we add v^t to C^t_{tight}, we also remove v^s from C^t_{tight}. ■

Corollary 8 *Algorithm 3 opens at most k facilities.*

Proof At the final step (12), the number of open facilities is equal to $|C_{\text{tight}}|$. By Proposition 7, the vector b^T satisfies constraints (C1) — (C5). So $b(F_j) = 1$ for $j \in C_{\text{tight}}$ and $F_j \cap F_{j'} = \emptyset$ for $j, j \in C_{\text{tight}}$; thus $|C_{\text{tight}}| = \sum_{j \in C_{\text{tight}}} b(F_j) \leq k$. ■

Proposition 9 *If $j \in C^t_{tight}$ for any time t, then $d(j, \mathcal{S}) \leq 3r_j$.*

Proof Let t be maximal such that $j \in C^t_{\text{tight}}$. We show the desired claim by induction on t. When $t = T$, then this certainly holds as step (12) will open some facility in F_j and thus $d(j, \mathcal{S}) \leq r_j$.

Suppose that j was added into C^s_{tight}, but was later removed from C^{t+1}_{tight} due to adding $z = v^t$. Thus there is some $i \in F_z \cap F_j$. When we added j in time s, we would have removed z from C^s_{tight} if $r_z \geq r_j/2$. Since this did not occur, it must hold that $r_z < r_j/2$.

Since z is present in C^{t+1}_{tight}, the induction hypothesis implies that $d(z, \mathcal{S}) \leq 3r_z$ and so

$$d(j, \mathcal{S}) \leq d(j, i) + d(i, z) + d(z, \mathcal{S}) \leq r_j + r_z + 3r_z \leq r_j + (r_j/2)(1 + 3) = 3r_j.$$ ■

Theorem 10 *Every $j \in \mathcal{C}$ has $\Pr[d(j, \mathcal{S}) \leq 9r_j] \geq p_j$.*

Proof We will prove by induction on t the following claim: suppose we condition on the full state of Algorithm 3 up to time t, and that $j \in C^t_{\text{tight}} \cup C^t_{\text{slack}}$. Then

$$\Pr[d(j, \mathcal{S}) \leq 9r_j] \geq b^t(F_j). \tag{2}$$

At $t = T$, this is clear; since $C^T_{\text{slack}} = \emptyset$, we must have $j \in C^T_{\text{tight}}$, and so $d(j, \mathcal{S}) \leq r_j$ with probability one. For the induction step at time t, note that as $\mathbf{E}[b^{t+1}(F_j)] = b(F_j)$, in order to prove (2) it suffices to show that if we also condition on the value of b^{t+1}, it holds that

$$\Pr[d(j, \mathcal{S}) \leq 9r_j \mid b^{t+1}] \geq b^{t+1}(F_j). \tag{3}$$

If j remains in $C^{t+1}_{\text{tight}} \cup C^{t+1}_{\text{slack}}$, then we immediately apply the induction hypothesis at time $t + 1$. So the only non-trivial thing to check is that (3) will hold even if j is removed from $C^{t+1}_{\text{tight}} \cup C^{t+1}_{\text{slack}}$.

If $j = v^t$ and $b^{t+1}(F_j) = 0$, then (3) holds vacuously. Otherwise, suppose that j is removed from C^t_{tight} at stage (10) due to adding $z = v^t$. Thus $r_j \geq r_z/2$ and there is some $i \in F_j \cap F_z$. By Proposition 9, this ensures that $d(z, \mathcal{S}) \leq 3r_z$. Thus with probability one we have

$$d(j, \mathcal{S}) \leq d(j, i) + d(i, z) + d(z, \mathcal{S}) \leq r_j + r_z + 3r_z \leq r_j + (2r_j)(1 + 3) = 9r_j.$$

This proves the induction. The claimed result follows since $b^0(F_j) = p_j$ and $C^0_{\text{slack}} = \mathcal{C}$. ■

3. Chance k-coverage: Approximating the Deterministic Case

An important special case of k-coverage is where $p_j = 1$ for all $j \in \mathcal{C}$. Here, the target distribution Ω is just a single set $\mathcal{S}$ satisfying $\forall j d(j, \mathcal{S}) \leq r_j$. In the homogeneous case, when all the r_j are equal to the same value, this is specifically the k-supplier problem. The usual approximation algorithm for this problem chooses a single approximating set $\mathcal{S}$, in which case the best guarantee available is $d(j, \mathcal{S}) \leq 3r_j$. We improve the distance guarantee by constructing a k-lottery $\tilde{\Omega}$ such that $d(j, \mathcal{S}) \leq 3r_j$ with probability one, and $\mathbf{E}_{\mathcal{S}\sim\tilde{\Omega}}[d(j, \mathcal{S})] \leq cr_j$, where the constant c satisfies the following bounds:

1. In the general case, $c = 1 + 2/e \approx 1.73576$;
2. In the SCC setting, $c = 1.60793$;
3. In the homogeneous SCC setting, $c = 1.592$.[3]

We show matching lower bounds in Section 4; the constant value $1 + 2/e$ is optimal for the general case (even for homogeneous instances), and for the third case the constant c cannot be made lower than $1 + 1/e \approx 1.367$.

We remark that this type of stochastic guarantee allows us to efficiently construct publicly-verifiable lotteries.

Proposition 11 *Let $\epsilon > 0$. In any of the above three cases, there is an expected polynomial time procedure to convert the given distribution Ω into an explicitly-enumerated k-lottery Ω', with support size $O(\frac{\log n}{\epsilon^2})$, such that $\Pr_{\mathcal{S}\sim\Omega'}[d(j, \mathcal{S}) \leq 3r_j] = 1$ and $\mathbf{E}_{\mathcal{S}\sim\Omega'}[d(j, \mathcal{S})] \leq c(1 + \epsilon)r_j$.*

Proof Take $X_1, \ldots, X_t$ as independent draws from Ω for $t = \frac{6 \log n}{c\epsilon^2}$ and set Ω' to be the uniform distribution on $\{X_1, \ldots, X_t\}$. To see that $\mathbf{E}_{\mathcal{S}\sim\Omega'}[d(j, \mathcal{S})] \leq c(1 + \epsilon)r_j$ holds with high probability, apply a Chernoff bound, noting that $d(j, X_1), \ldots, d(j, X_t)$ are independent random variables in the range $[0, 3r_j]$. ■

We use a similar algorithm to Algorithm 2 for this problem: we choose a covering set of clusters C', and open exactly one item from each cluster. The main difference is that

3. This value was calculated using some non-rigorous numerical analysis; we describe this further in what we call "Pseudo-Theorem" 17

instead of opening the nearest item V_j for each $j \in C'$, we instead open a cluster according to the solution b of $\mathcal{P}_{\text{chance}}$.

Algorithm 4 Rounding algorithm with clusters

1: Set $C' = \textsc{GreedyCluster}(F_j, r_j)$.
2: Set $F_0 = \mathcal{F} - \bigcup_{j \in C'} F_j$; this is the set of "unclustered" facilities
3: **for** $j \in C'$ **do**
4: Randomly select a point $W_j \in F_j$ according to the distribution $\Pr[W_j = i] = b_i$
// This is a valid probability distribution, as $b(F_j) = 1$
5: Let $\mathcal{S}_0 \leftarrow \textsc{DepRound}(b, F_0)$
6: Return $\mathcal{S} = \mathcal{S}_0 \cup \{W_j \mid j \in C'\}$

We will need the following technical result in order to analyze Algorithm 4.

Proposition 12 *For any set $U \subseteq \mathcal{F}$, we have*

$$\Pr[\mathcal{S} \cap U = \emptyset] \leq \prod_{i \in U \cap F_0} (1 - b_i) \prod_{j \in C'} (1 - b(U \cap F_j)) \leq e^{-b(U)}.$$

Proof The set U contains each W_j independently with probability $b(U \cap F_j)$. The set $\mathcal{S}_0$ is independent of them and by (P3) we have $\Pr[U \cap \mathcal{S}_0 = \emptyset] \leq \prod_{i \in U \cap F_0}(1 - b_i)$. So

$$\Pr[\mathcal{S} \cap U = \emptyset] \leq \prod_{i \in U \cap F_0} (1 - b_i) \prod_{j \in C'} (1 - b(U \cap F_j)) \leq \prod_{i \in U \cap F_0} e^{-b_i} \prod_{j \in C'} e^{-b(U \cap F_j)} = e^{-b(U)}$$ ■

At this point, we can show our claimed approximation ratio for the general (non-SCC) setting:

Theorem 13 *For any $j \in \mathcal{C}$, the solution set $\mathcal{S}$ of Algorithm 4 satisfies $d(j, \mathcal{S}) \leq 3r_j$ with probability one and $\mathbf{E}[d(j, \mathcal{S})] \leq (1 + 2/e)r_j$.*

Proof By Observation 2, there is some $v \in C'$ with $F_j \cap F_v \neq \emptyset$ and $r_v \leq r_j$. Letting $i \in F_j \cap F_v$, we have

$$d(j, \mathcal{S}) \leq d(j, i) + d(i, v) + d(v, \mathcal{S}) \leq r_j + r_v + r_v \leq 3r_j.$$

with probability one.

If $\mathcal{S} \cap F_j \neq \emptyset$, then $d(j, \mathcal{S}) \leq r_j$. Thus, a necessary condition for $d(j, \mathcal{S}) > r_j$ is that $\mathcal{S} \cap F_j = \emptyset$. Applying Proposition 12 with $U = F_j$ gives

$$\Pr[d(j, \mathcal{S}) > r_j] \leq \Pr[\mathcal{S} \cap F_j = \emptyset] \leq e^{-b(F_j)} = e^{-1}$$

and so $\mathbf{E}[d(j, \mathcal{S})] \leq r_j + 2r_j \Pr[d(j, \mathcal{S}) > r_j] \leq (1 + 2/e)r_j$. ■

3.1. The SCC Setting

To motivate the algorithm for the SCC setting (where $\mathcal{C} = \mathcal{F}$), note that if some client $j \in \mathcal{C}$ has some facility i opened in a nearby cluster F_v, then this guarantees that $d(j, \mathcal{S}) \leq d(j, v) + d(v, i) \leq 3r_j$. This is what we used to analyze the non-SCC setting. But, if instead of opening facility i, we opened v itself, then this would ensure that $d(j, \mathcal{S}) \leq 2r_j$. Thus, opening the centers of a cluster can lead to better distance guarantees compared to opening any other facility. We emphasize that this is only possible in the SCC setting, as in general we do not know that $v \in \mathcal{F}$.

We use the following Algorithm 5, which takes a parameter $q \in [0, 1]$ which we will discuss how to set shortly. We recall that we have assumed in this case that $j \in F_j$ for every $j \in \mathcal{C}$. (Also, note our notational convention that $\overline{q} = 1 - q$; this will be used extensively in this section to simplify the formulas.)

Algorithm 5 Rounding algorithm for k-center

1: Set $C' = \textsc{GreedyCluster}(F_j, r_j)$.
2: Set $F_0 = \mathcal{F} - \bigcup_{j \in C'} F_j$; this is the set of "unclustered" facilities
3: **for** $j \in C'$ **do**
4: Randomly select $W_j \in F_j$ according to the distribution $\Pr[W_j = i] = \overline{q} b_i + q[[i = j]]$
5: Let $\mathcal{S}_0 = \textsc{DepRound}(b, F_0)$
6: Return $\mathcal{S} = \mathcal{S}_0 \cup \{W_j \mid j \in C'\}$

This is the same as Algorithm 4, except that some of the values of b_i for $i \in F_j$ have been shifted to the cluster center j. In fact, we can think of Algorithm 5 as a two-part process: we first modify the fractional vector b to obtain a new fractional vector b' defined by

$$b'_i = \begin{cases} \overline{q} b_i + q & \text{if } i \in C' \\ \overline{q} b_i & \text{if } i \in F_j - \{j\} \text{ for } j \in C' \\ b_i & \text{if } i \in F_0 \end{cases}.$$

and we then execute Algorithm 4 on the resulting vector b'. In particular, Proposition 12 remains valid with respect to the modified vector b'.

Theorem 14 *Let $j \in \mathcal{C}$. After running Algorithm 5 with $q = 0.464587$ we have $d(j, \mathcal{S}) \leq 3r_j$ with probability one and $\mathbf{E}[d(j, \mathcal{S})] \leq 1.60793 r_j$.*

Proof Let $D = \{v \in C' \mid F_j \cap F_v \neq \emptyset, r_v \leq r_j\}$; note that $D \neq \emptyset$ by Observation 2. For each $v \in D \cup \{0\}$, set $a_v = b(F_j \cap F_v)$ and observe that $a_0 + \sum_{v \in D} a_v = b(F_j) = 1$.

As before, a necessary condition for $d(j, \mathcal{S}) > r_j$ is that $F_j \cap \mathcal{S} = \emptyset$. So by Proposition 12,

$$\begin{aligned} \Pr[d(j, \mathcal{S}) > r_j] \leq \Pr[F_j \cap \mathcal{S} = \emptyset] &\leq \prod_{i \in F_j \cap F_0} (1 - b'_i) \prod_{v \in C'} (1 - b'(F_v \cap F_j)) \\ &\leq \prod_{i \in F_j \cap F_0} (1 - b_i) \prod_{v \in D} (1 - \overline{q} b(F_v \cap F_j)) \leq e^{-b(F_j \cap F_0)} \prod_{v \in D} (1 - \overline{q} b(F_v \cap F_j)) \\ &= e^{-a_0} \prod_{v \in D} e^{a_v} (1 - \overline{q} a_v) = (1/e) \prod_{v \in D} e^{a_v} (1 - \overline{q} a_v). \end{aligned}$$

where the last equality comes from the fact that $a_0 + a(D) = 1$.

Similarly, if there is some $i \in \mathcal{S} \cap D$, then $d(j, \mathcal{S}) \le d(v, i) \le 2r_j$. Thus, a necessary condition for $d(j, \mathcal{S}) > 2r_j$ is that $\mathcal{S} \cap (D \cup F_j) = \emptyset$. Applying Proposition 12 with $U = D \cup F_j$ gives:

$$\begin{aligned}\Pr[d(j,\mathcal{S}) > 2r_j] &\le \prod_{v \in (D \cup F_j) \cap F_0} (1 - b_v) \prod_{v \in C'} (1 - b'((D \cup F_j) \cap F_v)) \\ &\le e^{-b(F_j \cap F_0)} \prod_{v \in D} \overline{q}(1 - a_v) = (1/e) \prod_{v \in D} e^{a_v} \overline{q}(1 - a_v)\end{aligned}$$

Putting these together gives:

$$\mathbf{E}[d(j,\mathcal{S})] \le r_j \Big(1 + 1/e \prod_{v \in D} e^{a_v}(1 - \overline{q}a_v) + 1/e \prod_{v \in D} e^{a_v}\overline{q}(1 - a_v)\Big) \tag{4}$$

Let us define $s = a(D)$ and $t = |D|$. By the AM-GM inequality we have:

$$\mathbf{E}[d(j,\mathcal{S})] \le r_j \Big(1 + e^{s-1}\big(1 - \overline{q}s/t\big)^t + e^{s-1}\overline{q}^t(1 - s/t)^t\Big)$$

This is a function of a single real parameter $s \in [0, 1]$ and a single integer parameter $t \ge 1$. Some simple analysis, which we omit here, shows that $\mathbf{E}[d(j, \mathcal{S})] \le 1.60793 r_j$. ■

3.2. The Homogeneous SCC Setting

From the point of view of the target distribution Ω, this setting is equivalent to the classical k-center problem. We may guess the optimal radius, and so we do not need to assume that the common value of r_j is "given" to us by some external process. By rescaling, we assume without loss of generality here that $r_j = 1$ for all j.

We will improve on Theorem 14 through a more complicated rounding process based on greedily-chosen partial clusters. Specifically, we select cluster centers $\pi(1), \ldots, \pi(n)$, wherein $\pi(i)$ is chosen to maximize $b(F_{\pi(i)} - F_{\pi(1)} - \cdots - F_{\pi(i-1)})$. By renumbering $\mathcal{C}$, we may assume without loss of generality that the resulting permutation π is the identity; therefore, we assume throughout this section that $\mathcal{C} = \mathcal{F} = [n]$ and for all pairs $i < j$ we have

$$b(F_i - F_1 - \cdots - F_{i-1}) \ge b(F_j - F_1 - \cdots - F_{i-1}) \tag{5}$$

For each $j \in [n]$, we let $G_j = F_j - F_1 - \cdots - F_{j-1}$ and we define $z_j = b(G_j)$. We say that G_j is a *full cluster* if $z_j = 1$ and a *partial cluster* otherwise. We note that the values of z appear in sorted order $1 = z_1 \ge z_2 \ge z_3 \ge \cdots \ge z_n \ge 0$.

We use the following randomized Algorithm 7 to select the centers. Here, the quantities $Q_{\mathrm{f}}, Q_{\mathrm{p}}$ (short for *full* and *partial*) are drawn from a joint probability distribution which we discuss later.

Algorithm 7 Partial-cluster based algorithm

1: Draw random variables $Q_{\mathrm{f}}, Q_{\mathrm{p}}$.
2: $Z \leftarrow \textsc{DepRound}(z)$
3: **for** $j \in Z$ **do**
4: Randomly select $W_j \in G_j$ according to the distribution $\Pr[W_j = i] = (\overline{q_j} y_i + q_j[[i = j]])/z_j$ where q_j is defined as

$$q_j = \begin{cases} Q_{\mathrm{f}} & \text{if } z_j = 1 \\ Q_{\mathrm{p}} & \text{if } z_j < 1 \end{cases}$$

5: Return $\mathcal{S} = \{W_j \mid j \in Z\}$

Before the technical analysis of Algorithm 7, let us provide some intuition. It may be beneficial to open the center of some cluster near a given client $j \in \mathcal{C}$ as this will ensure $d(j, \mathcal{S}) \leq 2$. However, there is no benefit to opening more than one such cluster center. So, we would like a significant negative correlation between opening the centers of distinct clusters near j. Unfortunately, the full clusters all "look alike," and so it seems impossible to enforce any significant negative correlation among them.

Partial clusters break the symmetry. There is at least one full cluster near j, and possibly some partial clusters as well. We will create a probability distribution with significant negative correlation between the event that partial clusters open their centers and the event that full clusters open their centers. This decreases the probability that j will see multiple neighboring clusters open their centers, which in turn gives an improved value of $\mathbf{E}[d(j, \mathcal{S})]$.

The dependent rounding in Algorithm 7 ensures that $|Z| \leq \lceil \sum_{j=1}^n z_j \rceil = \sum_{j=1}^n b(F_j - F_1 - \cdots - F_{j-1}) = b(\mathcal{F}) \leq k$, and so $|\mathcal{S}| \leq k$ as required. We also need the following technical result; the proof is essentially the same as Proposition 12 and is omitted.

Proposition 15 *For any $U \subseteq \mathcal{C}$, we have*

$$\Pr[\mathcal{S} \cap U = \emptyset \mid Q_f, Q_p] \leq \prod_{j=1}^{n} \big(1 - \overline{q}_j b(U \cap G_j) - q_j z_j [[j \in U]]\big).$$

Our main lemma to analyze Algorithm 7 is the following:

Lemma 16 *Let $i \in [n]$. Define $J_f, J_p \subseteq [n]$ as*

$$J_f = \{j \in [n] \mid F_i \cap G_j \neq \emptyset, z_j = 1\}$$
$$J_p = \{j \in [n] \mid F_i \cap G_j \neq \emptyset, z_j < 1\}$$

Let $m = |J_f|$ and $J_p = \{j_1, \ldots, j_t\}$ where $j_1 \leq j_2 \leq \cdots \leq j_t$. For each $\ell = 1, \ldots, t+1$ define

$$u_\ell = b(F_i \cap G_{j_\ell}) + b(F_i \cap G_{j_{\ell+1}}) + \cdots + b(F_i \cap G_{j_t})$$

Then $1 \geq u_1 \geq u_2 \geq \cdots \geq u_t \geq u_{t+1} = 0$ and $m \geq 1$. Furthermore, if we condition on a fixed value of (Q_f, Q_p) then we have

$$\mathbf{E}[d(i, \mathcal{S})] \leq 1 + \left(1 - \overline{Q_f}\frac{\overline{u_1}}{m}\right)^m \prod_{\ell=1}^{t} (1 - \overline{Q_p}(u_\ell - u_{\ell+1})) + \left(\overline{Q_f}(1 - \frac{\overline{u_1}}{m})\right)^m \prod_{\ell=1}^{t} (\overline{u_\ell} + \overline{Q_p} u_{\ell+1}).$$

Proof For $\ell = 1, \ldots, t$ we let $a_\ell = b(F_i \cap G_{j_\ell}) = u_\ell - u_{\ell+1}$. For $j \in J_{\rm f}$, we let $s_j = b(F_i \cap G_j)$.

First, we claim that $z_{j_\ell} \geq u_\ell$ for $\ell = 1, \ldots, t$. For, by (5), we have

$$\begin{aligned} z_{j_\ell} &\geq b(F_i - F_1 - \cdots - F_{j_\ell - 1}) \geq b(F_i) - \sum_{j \in J_{\rm f}} b(F_i \cap G_j) - \sum_{v < j_\ell} b(F_i \cap G_v) \\ &= b(F_i) - \sum_{j \in J_{\rm f}} b(F_i \cap G_j) - \sum_{v < \ell} b(F_i \cap G_{j_v}) \qquad \text{as } F_i \cap G_v = \emptyset \text{ for } v \notin J_{\rm p} \cup J_{\rm f} \\ &= 1 - \sum_{j \in J_{\rm f}} s_j - \sum_{v=1}^{\ell-1} a_v = u_\ell. \end{aligned}$$

Next, observe that if $m = 0$, we have $u_1 = \sum_{\ell=1}^{t} b(F_i \cap G_{j_\ell}) = \sum_{j=1}^{n} b(F_i \cap G_j) = b(F_i) = 1$. Applying the above bound at $\ell = 1$ gives $z_{j_1} \geq u_1 = 1$. But, by definition of $J_{\rm p}$ we have $z_{j_1} < 1$, which is a contradiction. This shows that $m \geq 1$.

We finish by showing the bound on $\mathbf{E}[d(i, \mathcal{S})]$. All the probabilities here should be interpreted as conditioned on fixed values for parameters $Q_{\rm f}, Q_{\rm p}$.

A necessary condition for $d(i, \mathcal{S}) > 1$ is that no point in F_i is open. Applying Proposition 15 with $U = F_i$ yields

$$\begin{aligned} \Pr[\mathcal{S} \cap F_i = \emptyset] &\leq \prod_{j \in J_{\rm f}} (1 - \overline{Q_{\rm f}} b(F_i \cap G_j) - Q_{\rm f}[[j \in F_i]]) \prod_{j \in J_{\rm p}} (1 - \overline{Q_{\rm p}} b(F_i \cap G_j)) - Q_{\rm p} z_j [[j \in F_i]]) \\ &\leq \prod_{j \in J_{\rm f}} (1 - \overline{Q_{\rm f}} b(F_i \cap G_j)) \prod_{j \in J_{\rm p}} (1 - \overline{Q_{\rm p}} b(F_i \cap G_j)) = \prod_{j \in J_{\rm f}} (1 - \overline{Q_{\rm f}} s_j) \prod_{\ell=1}^{t} (1 - \overline{Q_{\rm p}} a_\ell). \end{aligned}$$

A necessary condition for $d(i, \mathcal{S}) > 2$ is that we do not open any point in F_i, nor do we open center of any cluster intersecting with F_i. Applying Proposition 15 with $U = F_i \cup J_{\rm f} \cup J_{\rm p}$, and noting that $z_{j_\ell} \leq u_\ell$, we get:

$$\begin{aligned} \Pr[\mathcal{S} \cap U = \emptyset] &\leq \prod_{j \in J_{\rm f}} (1 - \overline{Q_{\rm f}} b(U \cap G_j) - Q_{\rm f}) \prod_{j \in J_{\rm p}} (1 - \overline{Q_{\rm p}} b(U \cap G_j)) - Q_{\rm p} z_j) \\ &\leq \prod_{j \in J_{\rm f}} (1 - \overline{Q_{\rm f}} b(F_i \cap G_j) - Q_{\rm f}) \prod_{\ell=1}^{t} (1 - \overline{Q_{\rm p}} b(F_i \cap G_{j_\ell}) - Q_{\rm p} z_{j_\ell}) \\ &= \prod_{j \in J_{\rm f}} \overline{Q_{\rm f}} (1 - s_j) \prod_{\ell=1}^{t} (1 - \overline{Q_{\rm p}} a_\ell - Q_{\rm p} z_{j_\ell}) \leq \prod_{j \in J_{\rm f}} \overline{Q_{\rm f}} (1 - s_j) \prod_{\ell=1}^{t} (1 - \overline{Q_{\rm p}} a_\ell - Q_{\rm p} u_\ell) \end{aligned}$$

Thus,

$$\mathbf{E}[d(i, \mathcal{S})] \leq 1 + \prod_{j \in J_{\rm f}} (1 - \overline{Q_{\rm f}} s_j) \prod_{\ell=1}^{t} (1 - \overline{Q_{\rm p}} a_\ell) + \prod_{j \in J_{\rm f}} \overline{Q_{\rm f}} (1 - s_j) \prod_{\ell=1}^{t} (1 - \overline{Q_{\rm p}} a_\ell - Q_{\rm p} u_\ell) \quad (6)$$

The sets G_j partition $[n]$, so $\sum_{j \in {\rm f}} s_j = 1 - \sum_{\ell=1}^{t} a_\ell = 1 - u_1$. So by the AM-GM inequality, we have

$$\mathbf{E}[d(i, \mathcal{S})] \leq 1 + \left(1 - \overline{Q_{\rm f}} \frac{\overline{u_1}}{m}\right)^m \prod_{\ell=1}^{t} (1 - \overline{Q_{\rm p}} a_\ell) + \left(\overline{Q_{\rm f}} (1 - \frac{\overline{u_1}}{m})\right)^m \prod_{\ell=1}^{t} (1 - \overline{Q_{\rm p}} a_\ell - Q_{\rm p} u_\ell). \quad (7)$$

The claim follows as $a_\ell = u_\ell - u_{\ell+1}$. ■

Lemma 16 gives an upper bound on $\mathbf{E}[d(i,\mathcal{S})]$ for a fixed distribution on $Q_\mathrm{f}, Q_\mathrm{p}$ and for fixed values of the parameters $m, u_1, \ldots, u_t$. By a computer search, along with a number of numerical tricks, we can obtain an upper bound over all values for m and all possible sequences $u_1, \ldots, u_t$ satisfying $1 = u_1 \geq u_2 \geq \cdots \geq u_t$. This gives the following result:

Pseudo-Theorem 17 *Suppose that Q_f, Q_p has the following distribution:*

$$(Q_f, Q_p) = \begin{cases} (0.4525, 0) & \text{with probability } p = 0.773436 \\ (0.0480, 0.3950) & \text{with probability } 1-p \end{cases}.$$

Then for all $j \in \mathcal{C}$ we have $d(j,\mathcal{S}) \leq 3r_j$ with probability one, and $\mathbf{E}[d(j,\mathcal{S})] \leq 1.592 r_j$.

We call this a "pseudo-theorem" because some of the computer calculations used double-precision floating point for convenience, without carefully tracking rounding errors. In principle, this could have been computed using rigorous numerical analysis, giving a true theorem. Since there are a number of technical complications in this calculation, we defer the details to Appendix A.

4. Lower Bounds on Approximating Chance k-coverage

We next show lower bounds for the chance k-coverage problems of Sections 2 and 3. These constructions are adapted from lower bounds for approximability of k-median (Guha and Khuller, 1999), which in turn are based on the hardness of set cover.

Formally, a set cover instance consists of a collection of sets $\mathcal{B} = \{S_1, \ldots, S_m\}$ over a ground set $[n]$. For any set $X \subseteq [m]$ we define $S_X = \bigcup_{i \in X} S_i$. The goal is to select a collection $X \subseteq [m]$ of minimum size such that $S_X = [n]$. The minimum value of $|X|$ thus obtained is denoted by OPT.

We quote a result of Moshkovitz (2015) on the inapproximability of set cover.

Theorem 18 (Moshkovitz (2015)) *Assuming $P \neq NP$, there is no polynomial-time algorithm which guarantees a set-cover solution X with $S_X = [n]$ and $|X| \leq (1-\epsilon)\ln n \times OPT$, where $\epsilon > 0$ is any constant.*

We will need a simple corollary of Theorem 18, which is a (slight reformulation) of the hardness of approximating max-coverage.

Corollary 19 *Assuming $P \neq NP$, there is no polynomial-time algorithm which guarantees a set-cover solution X with $|X| \leq OPT$ and $\left|S_X\right| \geq cn$ for any constant $c > 1 - 1/e$.*

Proof Suppose for contradiction that $\mathcal{A}$ is such an algorithm. We will repeatedly apply $\mathcal{A}$ to solve residual instances, obtaining a sequence of solutions $X_1, X_2, \ldots,$. Specifically, for iterations $i \geq 1$, we define $U_i = [n] - \bigcup_{j<i} S_{X_j}$ and $\mathcal{B}_i = \{S \cap U_i \mid S \in \mathcal{B}\}$, and we let X_i denote the output of $\mathcal{A}$ on instance $\mathcal{B}_i$.

Each $\mathcal{B}_i$ has a solution of cost at most OPT. Thus, $|X_i| \leq \mathrm{OPT}$ and $|U_i \cap S_{X_i}| \geq c|U_i|$. Thus $|U_{i+1}| = |U_i - S_{X_i}| \leq (1-c)|U_i|$. So, for $s = \lceil 1 + \frac{\ln n}{\ln(\frac{1}{1-c})} \rceil$ we have $U_s = \emptyset$, and so

the set $X = X_1 \cup \cdots \cup X_s$ solves the original set-cover instance $\mathcal{B}$, and $|X| \leq \sum_{i=1}^{s} |X_i| \leq (1 + \frac{\ln n}{\ln(\frac{1}{1-c})})\text{OPT}$.

Since $c > 1 - 1/e$, we have $\frac{1}{\ln(\frac{1}{1-c})} \leq (1 - \Omega(1))$, which contradicts Theorem 18. ■

Theorem 20 *Assuming $P \neq NP$, there is a family of homogeneous chance k-coverage instances, with a feasible demand of $p_j = r_j = 1$ for all clients j, such that no polynomial-time algorithm can guarantee a distribution Ω with either of the following:*

1. *$\forall j \mathbf{E}_{\mathcal{S}\sim\Omega}[d(j, \mathcal{S})] \leq cr_j$ for constant $c < 1 + 2/e$*
2. *$\forall j \Pr_{\mathcal{S}\sim\Omega}[d(j, \mathcal{S}) < 3r_j] \geq cp_j$ for constant $c > 1 - 1/e$*

In particular, approximation constants in Theorem 4, Proposition 5, and Theorem 13 cannot be improved.

Proof Consider a set cover instance $\mathcal{B} = \{S_1, \ldots, S_m\}$. We begin by guessing the value OPT (there are at most m possibilities, so this can be done in polynomial time). We define a k-center instance with $k = \text{OPT}$ and with disjoint client and facility sets, where $\mathcal{F}$ is identified with $[m]$ and $\mathcal{C}$ is identified with $[n]$. We define d by $d(i, j) = 1$ if $j \in S_i$ and $d(i, j) = 3$ otherwise.

If X is an optimal solution to $\mathcal{B}$ then $d(j, X) \leq 1$ for all points $j \in \mathcal{C}$. So there exists a (deterministic) distribution with feasible demand parameters $p_j = 1, r_j = 1$. Also, note that $d(j, \mathcal{S}) \in \{1, 3\}$ with probability one for any client j. Thus, if j satisfies the second property $\Pr_{\mathcal{S}\sim\Omega}[d(j, \mathcal{S}) < 3r_j] \geq cp_j$ for constant $c > 1 - 1/e$, then it satisfies $\mathbf{E}[d(j, \mathcal{S})] \leq 1 + 2(1 - cp_j) = 3 - 2c = (1 + 2/e - \Omega(1))r_j$. So it will satisfy the first property as well. So it suffices to show that no algorithm $\mathcal{A}$ can satisfy the first property.

Suppose that $\mathcal{A}$ does satisfy the first property. The resulting solution set $\mathcal{S} \subseteq \mathcal{F}$ can be regarded as a solution X to the set cover instance, where $d(j, \mathcal{S}) = 1 + 2[[j \notin S_X]]$. Thus

$$\sum_{j\in[n]} d(j, \mathcal{S}) = |S_X| + 3(n - |S_X|),$$

and so $|S_X| = \frac{3n - \sum_{j\in[n]} d(j,\mathcal{S})}{2}$. As $\mathbf{E}[d(j, \mathcal{S})] \leq cr_j = c$ for all j, this implies that $\mathbf{E}[|S_X|] \geq \frac{(3-c)n}{2}$.

After an expected constant number of repetitions of this process we can ensure that $|S_X| \geq c'n$ for some constant $c' > \frac{3-(1+2/e)}{2} = 1 - 1/e$. This contradicts Corollary 19. ■

A slightly more involved construction applies to the homogeneous SCC setting.

Theorem 21 *Assuming $P \neq NP$, there is a family of homogeneous SCC chance k-coverage instances, with a feasible demand of $p_j = r_j = 1$ for all j, such that no polynomial-time algorithm can guarantee a distribution Ω with either of the following:*

1. *$\forall j \mathbf{E}_{\mathcal{S}\sim\Omega}[d(j, \mathcal{S})] \leq cr_j$ for constant $c < 1 + 1/e$*
2. *$\forall j \Pr_{\mathcal{S}\sim\Omega}[d(j, \mathcal{S}) < 2r_j] \geq cp_j$ for constant $c > 1 - 1/e$*

In particular, the approximation constants in Theorem 4 and Proposition 5 cannot be improved for SCC instances, and the approximation factor 1.592 *in Pseudo-Theorem 17 cannot be improved below* $1 + 1/e$.

Proof Consider a set cover instance $\mathcal{B} = \{S_1, \ldots, S_m\}$, where we have guessed the value $\mathrm{OPT} = k$. We define a k-center instance as follows. For each $i \in [m]$, we create an item v_i and for each $j \in [n]$ we create $t = n^2$ distinct items $w_{j,1}, \ldots, w_{j,t}$. We define the distance by setting $d(v_i, w_{j,t}) = 1$ if $j \in S_i$ and $d(v_i, v_{i'}) = 1$ for all $i, i' \in [m]$, and $d(x, y) = 2$ for all other distances. This problem size is polynomial (in m, n), and so $\mathcal{A}$ runs in time $\mathrm{poly}(m, n)$.

If X is an optimal solution to the set cover instance, the corresponding set $\mathcal{S} = \{v_i \mid i \in X\}$ satisfies $d(j, \mathcal{S}) \leq 1$ for all $j \in \mathcal{C}$. So the demand vector $p_j = r_j = 1$ is feasible. Also, note that that $d(j, \mathcal{S}) \in \{0, 1, 2\}$ with probability one for any j. Thus, if j satisfies the second property $\Pr_{\mathcal{S}\sim\Omega}[d(j, \mathcal{S}) < 2r_j] \geq cp_j$ for constant $c > 1 - 1/e$, then it satisfies $\mathbf{E}[d(j, \mathcal{S})] \leq 1 + 1(1 - cp_j) = (1 + 1/e - \Omega(1))r_j$. So it will satisfy the first property as well. So it suffices to show that no algorithm $\mathcal{A}$ can satisfy the first property.

Suppose that algorithm $\mathcal{A}$ satisfies the first property. From the solution set $\mathcal{S}$, we construct a corresponding set-cover solution by $X = \{i \mid v_i \in \mathcal{S}\}$. For $w_{j,\ell} \notin \mathcal{S}$, we can observe that $d(w_{j,\ell}, \mathcal{S}) = 1 + [[j \notin S_X]]$. Therefore, we have

$$\begin{aligned}\sum_{j\in[n]} \sum_{\ell=1}^{t} d(w_{j,\ell}, \mathcal{S}) &\geq \sum_{j,\ell: w_{j,\ell}\notin\mathcal{S}} (1 + [[j \notin S_X]]) \geq \sum_{j,\ell} (1 + [[j \notin S_X]]) - 2|\mathcal{S}| \\ &\geq n^2(2n - |S_X|) - 2k,\end{aligned}$$

and so $|S_X| \geq 2n - \frac{\sum_{j,\ell} d(w_{j,\ell}, \mathcal{S})}{n^2} - 2k/n^2$.

Taking expectations and using our upper bound on $\mathbf{E}[d(j, \mathcal{S})]$, we have $\mathbf{E}[|S_X|] \geq 2n - cn - 2k/n^2 \geq (2 - c)n - 2/n$. Thus, for n sufficiently large, after an expected constant number of repetitions of this process we get $|S_X| \geq (2 - c - o(1))n \geq (1 - 1/e + \Omega(1))n$. This contradicts Corollary 19. ■

5. Approximation Algorithm for $\mathbf{E}[d(j, \mathcal{S})]$

In the chance k-coverage problem, our goal is to achieve certain fixed values of $d(j, \mathcal{S})$ with a certain probability. In this section, we consider another criterion for Ω; we wish to achieve certain values for the expectation $\mathbf{E}_{\mathcal{S}\sim\Omega}[d(j, \mathcal{S})]$. We suppose we are given values t_j for every $j \in C$, such that the target distribution Ω satisfies

$$\mathbf{E}_{\mathcal{S}\sim\Omega}[d(j, \mathcal{S})] \leq t_j. \tag{8}$$

In this case, we say that the vector t_j is *feasible.* As before, if all the values of t_j are equal to each other, we say that the instance is *homogeneous.* We show how to leverage any approximation algorithm for k-median with approximation ratio α, to ensure our target distribution $\tilde{\Omega}$ will satisfy

$$\mathbf{E}_{\mathcal{S}\sim\tilde{\Omega}}[d(j, \mathcal{S})] \leq (\alpha + \epsilon)t_j.$$

More specifically, we need an approximation algorithm for weighted k-median. In this setting, we have a problem instance $\mathcal{I} = \mathcal{F}, \mathcal{C}, d$ along with non-negative weights w_j for $j \in \mathcal{C}$, and we wish to find $\mathcal{S} \in \binom{\mathcal{F}}{k}$ minimizing $\sum_{j \in \mathcal{C}} w_j d(j, \mathcal{S})$. (Nearly all approximation algorithms for ordinary k-median can be easily adapted to the weighted setting, for example, by replicating clients.) If we fix an approximation algorithm $\mathcal{A}$ for (various classes of) weighted k-median, then for any problem instance $\mathcal{I}$ we define

$$\alpha_{\mathcal{I}} = \sup_{\text{weights } w} \frac{\sum_{j \in \mathcal{C}} w_j d(j, \mathcal{A}(\mathcal{I}, w))}{\min_{\mathcal{S} \in \binom{\mathcal{F}}{k}} \sum_{j \in \mathcal{C}} w_j d(j, \mathcal{S})}.$$

We first show how to use the k-median approximation algorithm to achieve a set $\mathcal{S}$ which "matches" the desired distances t_j:

Proposition 22 *Given a weighted instance $\mathcal{I}$ and a parameter $\epsilon > 0$, there is a polynomial-time algorithm to produce a set $\mathcal{S} \in \binom{\mathcal{F}}{k}$ satisfying:*

1. $\sum_{j \in \mathcal{C}} w_j \frac{d(j,\mathcal{S})}{t_j} \leq (\alpha_{\mathcal{I}} + O(\epsilon)) \sum_{j \in \mathcal{C}} w_j$,
2. *Every $j \in \mathcal{C}$ has $d(j, \mathcal{S}) \leq n t_j / \epsilon$.*

Proof We assume $\alpha_{\mathcal{I}} = O(1)$, as constant-factor approximation algorithms for k-median exist. By rescaling w, we assume without loss of generality that $\sum_{j \in \mathcal{C}} w_j = 1$. By rescaling ϵ, it suffices to show that $d(j, \mathcal{S}) \leq O(n t_j / \epsilon)$.

Let us define the weight vector $z_j = \frac{\epsilon/n + w_j}{t_j}$. Letting Ω be a distribution satisfying (8), we have

$$\mathbf{E}_{\mathcal{S} \sim \Omega}\big[\sum_{j \in \mathcal{C}} z_j d(j, \mathcal{S})\big] = \sum_{j \in \mathcal{C}} z_j t_j \leq \sum_{j \in \mathcal{C}} (\frac{\epsilon}{n t_j} + \frac{w_j}{t_j}) t_j = \epsilon |\mathcal{C}|/n + \sum_{j \in \mathcal{C}} w_j = 1 + \epsilon.$$

In particular, there exists some $\mathcal{S} \in \binom{\mathcal{F}}{k}$ with $\sum_{j \in \mathcal{C}} z_j d(j, \mathcal{S}) \leq 1 + \epsilon$. When we apply algorithm $\mathcal{A}$ with weight vector z, we thus get a set $\mathcal{S} \in \binom{\mathcal{F}}{k}$ with $\sum_{j \in \mathcal{C}} z_j d(j, \mathcal{S}) \leq \alpha_{\mathcal{I}}(1+\epsilon)$. We claim that this set $\mathcal{S}$ satisfies the two conditions of the theorem. First, we have

$$\sum_{j \in \mathcal{C}} \frac{w_j d(j, \mathcal{S})}{t_j} \leq \sum_{j \in \mathcal{C}} z_j d(j, \mathcal{S}) \leq \alpha_{\mathcal{I}}(1 + \epsilon) \leq (\alpha_{\mathcal{I}} + O(\epsilon)) \sum_j w_j.$$

Next, for any given $j \in \mathcal{C}$, we have

$$\frac{d(j, \mathcal{S})}{t_j} \leq d(j, \mathcal{S}) z_j (n/\epsilon) \leq (n/\epsilon) \sum_{w \in \mathcal{C}} z_w d(w, \mathcal{S}) \leq (n/\epsilon) \alpha_{\mathcal{I}}(1 + \epsilon) \leq O(n/\epsilon). \qquad \blacksquare$$

Theorem 23 *There is an algorithm which takes as input an instance $\mathcal{I}$, a parameter $\epsilon > 0$ and a feasible vector t_j, runs in time poly$(n, 1/\epsilon)$, and returns an explicitly enumerated distribution $\tilde{\Omega}$ with support size n and $\mathbf{E}_{S \sim \tilde{\Omega}}[d(j, \mathcal{S})] \leq (\alpha_{\mathcal{I}} + \epsilon) t_j$ for all $j \in \mathcal{C}$.*

Proof We assume without loss of generality that $\epsilon \leq 1$; by rescaling ϵ it suffices to show that $\mathbf{E}[d(j,\mathcal{S})] \leq (\alpha_{\mathcal{I}} + O(\epsilon))t_j$.

We begin with the following Algorithm 8, which uses a form of multiplicative weights update with repeated applications of Proposition 22.

Algorithm 8 Approximation algorithm for $\mathbf{E}[d(j,\mathcal{S})]$: first phase

1: **for** $\ell = 1, \ldots, r = \frac{n \ln n}{\epsilon^3}$ **do**

2: Let $X_\ell \in \binom{\mathcal{F}}{k}$ be the resulting of applying the algorithm of Proposition 22 with parameters ϵ, t_j and weight vector w given by

$$w_j = \exp\Big(\epsilon^2 \sum_{s=1}^{\ell-1} \frac{d(j,X_s)}{nt_j}\Big),$$

3: Set $\tilde{\Omega}'$ to be the uniform distribution on $X_1, \ldots, X_r$

Let us define $\phi = \epsilon^2/n$. For each iteration $\ell = 1, \ldots, r+1$ let $u_j^{(\ell)} = \phi d(j, X_\ell)/t_j$, and let $w_j^{(\ell)} = e^{\sum_{s=1}^{\ell-1} u_j^{(s)}}$ denote the weight vector. Proposition 22 ensures that $u_j^{(\ell)} \leq \epsilon$, and thus $e^{u_j^{(\ell)}} \leq 1 + \frac{e^\epsilon - 1}{\epsilon} u_j^{(\ell)} \leq 1 + (1+\epsilon)u_j^{(\ell)}$, as well as ensuring that $\sum_j w_j^{(\ell)} u_j^{(\ell)} \leq \phi(\alpha_I + O(\epsilon)) \sum_j w_j^{(\ell)}$.

Now let $\Phi_\ell = \sum_{j \in \mathcal{C}} w_j^{(\ell)}$. Note that $\Phi_1 = n$, and for each $\ell \geq 1$, we have

$$\begin{aligned}\Phi_{\ell+1} &= \sum_{j \in \mathcal{C}} w_j^{(\ell)} e^{u_j^{(\ell)}} \leq \sum_{j \in \mathcal{C}} w_j^{(\ell)} \big(1 + (1+\epsilon)u_j^{(\ell)}\big) \leq \sum_{j \in \mathcal{C}} w_j^{(\ell)} + (1+\epsilon) \sum_{j \in \mathcal{C}} w_j^{(\ell)} u_j^{(\ell)} \\ &\leq \Phi_\ell \big(1 + (1+\epsilon)\phi(\alpha_{\mathcal{I}} + O(\epsilon))\big) \leq \Phi_\ell e^{\phi(\alpha_{\mathcal{I}} + O(\epsilon))}\end{aligned}$$

This recurrence relation implies that $\Phi_\ell \leq n e^{(\ell-1)\phi(\alpha_{\mathcal{I}} + O(\epsilon))}$. Since $w_j^{(r+1)} \leq \Phi^{r+1}$, this implies

$$\sum_{\ell=1}^{r} \phi d(j, X_\ell)/t_j = \ln w_j^{(r+1)} \leq \ln \Phi_{r+1} \leq \ln n + r\phi(\alpha_{\mathcal{I}} + O(\epsilon)).$$

or equivalently,

$$\sum_{\ell=1}^{r} \frac{d(j, X_\ell)}{r} = t_j \Big(\frac{\ln n}{r\phi} + (\alpha_{\mathcal{I}} + O(\epsilon))\Big)$$

As $r = \frac{n \ln n}{\epsilon^3} = \frac{\ln n}{\epsilon \phi}$, we thus have $\frac{\sum_{\ell=1}^r d(j,X_\ell)}{r} \leq (\alpha_{\mathcal{I}} + O(\epsilon))t_j$. Thus, the distribution $\tilde{\Omega}'$ satisfies

$$\forall j \in \mathcal{C} \qquad \mathbf{E}_{\mathcal{S} \sim \tilde{\Omega}'}[d(j,\mathcal{S})] \leq (\alpha_I + O(\epsilon))t_j. \tag{9}$$

Now the distribution $\tilde{\Omega}'$ satisfies the condition on $\mathbf{E}[d(j,\mathcal{S})]$, but its support is too large. We can reduce the support size to $|\mathcal{C}|$ by moving in the nullspace of the $|\mathcal{C}|$ linear constraints (9). ■

Byrka et al. (2017) have shown a $2.675+\epsilon$-approximation algorithm for k-median, which automatically gives a $2.675+\epsilon$-approximation algorithm for k-lottery as well. Some special

cases of k-median have more efficient approximation algorithms. For instance, Cohen-Addad et al. (2016) gives a PTAS for k-median problems derived from a planar graph, and Ahmadian et al. (2017) gives a $2.633+\epsilon$-approximation for Euclidan distances. These immediately give approximation algorithms for the corresponding k-lotteries. We also note that, by Theorem 20, one cannot obtain a general approximation ratio better than $1+2/e$ (or $1+1/e$ in the SCC setting).

6. Determinizing a k-lottery

Suppose that we have a set of feasible weights t_j such some k-lottery distribution Ω satisfies $\mathbf{E}_{\mathcal{S}\sim\Omega}[d(j,\mathcal{S})] \le t_j$; let us examine how to find a *single, deterministic* set $\mathcal{S}$ with $d(j,\mathcal{S}) \approx t_j$. We refer to this as the problem of *determinizing* the lottery Ω. Note that this can be viewed as a converse to the problem considered in Section 3.

We will see that, in order to obtain reasonable approximation ratios, we may need $|\mathcal{S}|$ to be significantly larger than k. We thus define an (α,β)-*determinization* to be a set $\mathcal{S} \in \binom{\mathcal{F}}{k'}$ with $k' \le \alpha k$ and $d(j,\mathcal{S}) \le \beta t_j$ for all $j \in \mathcal{C}$. We emphasize that we cannot necessarily obtain $(1,1)$-determinizations, even with unbounded computational resources. The following simple example illustrates the tradeoff between parameters α and β:

Observation 24 *Let $\alpha,\beta,k \ge 1$. If $\beta < \frac{\alpha k+1}{(\alpha-1)k+1}$, there is a homogeneous SCC instance for which no (α,β)-determinization exists.*

Proof Let $k' = \alpha k$ and consider a problem instance with $\mathcal{F} = \mathcal{C} = \{1,\dots,k'+1\}$, and $d(i,j) = 1$ for every distinct i,j. Clearly, every $\mathcal{S} \in \binom{\mathcal{F}}{k'}$ satisfies $\min_j d(j,\mathcal{S}) = 1$. When Ω is the uniform distribution on $\binom{\mathcal{F}}{k}$, we have $\mathbf{E}[d(j,\mathcal{S})] = 1 - \frac{k}{k'+1}$. Thus $t_j = \frac{k}{k'+1}$ is feasible and therefore $\beta \ge \frac{1}{1-\frac{k}{k'+1}} = \frac{\alpha k+1}{(\alpha-1)k+1}$. ■

In particular, when $\alpha = 1$ we must have $\beta \ge k+1$ and when $k \to \infty$, we must have $\beta \gtrsim \frac{\alpha}{\alpha-1}$.

We examine three main regimes for the parameters (α,β): (1) the case where α,β are scale-free constants; (2) the case where β is close to one, in which case α must be of order $\log n$; (3) the case where $\alpha = 1$, in which case β must be order k.

Our determinization algorithms for the first two cases will based on the following LP denoted $\mathcal{P}_{\text{expectation}}$, defined in terms of fractional vectors $b_i, a_{i,j}$ where i ranges over $\mathcal{F}$ and j ranges over $\mathcal{C}$:

(A1) $\forall j \in \mathcal{C}, \quad \sum_{i\in\mathcal{F}} a_{i,j} d(i,j) \le t_j,$

(A2) $\forall j \in \mathcal{C}, \quad \sum_{i\in\mathcal{F}} a_{i,j} = 1,$

(A3) $\forall i \in \mathcal{F}, y \in \mathcal{C}, \quad 0 \le a_{i,j} \le b_i,$

(A4) $\forall i \in \mathcal{F}, \quad 0 \le b_i \le 1,$

(A5) $\sum_{i\in\mathcal{F}} b_i \le k.$

Theorem 25 *If t_j is feasible, then $\mathcal{P}_{expectation}$ has a fractional solution.*

Proof Let Ω be a probability distribution with $\mathbf{E}[d(j,\mathcal{S})] \leq t_j$. For any draw $\mathcal{S} \sim \Omega$, define random variable Z_j to be the facility of $\mathcal{S}$ matched by j. Now consider the fractional vector defined by

$$b_i = \Pr_{\mathcal{S}\sim\Omega}[i \in \mathcal{S}], \qquad a_{i,j} = \Pr_{\mathcal{S}\sim\Omega}[Z_j = i]$$

We claim that this satisfies (A1) — (A5). For (A1), we have

$$\mathbf{E}[d(j,\mathcal{S})] = \mathbf{E}[d(j,Z_j)] = \sum_{i\in\mathcal{F}} d(i,j)\Pr[Z_j = i] = \sum_{i\in F} d(i,j)a_{i,j} \leq t_j.$$

For (A2), note that $\sum_i \Pr[Z_j = i] = 1$. For (A3), note that $Z_j = i$ can only occur if $i \in \mathcal{S}$. (A4) is clear, and (A5) holds as $|\mathcal{S}| = k$ with probability one. ■

We next describe upper and lower bounds for these three regimes.

6.1. The Case Where α, β are Scale-free Constants.

For this regime (with all parameters independent of problem size n and k), we may use the following Algorithm 9, which is based on greedy clustering using a solution to $\mathcal{P}_{\text{expectation}}$.

Algorithm 9 (α,β)-determinization algorithm

1: Let a, b be a solution to $\mathcal{P}_{\text{expectation}}$.
2: For every $j \in \mathcal{C}$, select $r_j \geq 0$ to be minimal such that $\sum_{i\in B(j,r_j)} a_{i,j} \geq 1/\alpha$
3: By splitting facilities, form a set $F_j \subseteq B(j,r_j)$ with $b(F_j) = 1/\alpha$.
4: Set $C' = \textsc{GreedyCluster}(F_j, \theta(j) + r_j)$
5: Output solution set $\mathcal{S} = \{V_j \mid j \in C'\}$.

Step (3) is well-defined, as (A3) ensures that $b(B(j,r_j)) \geq \sum_{i\in B(j,r_j)} a_{i,j} \geq 1/\alpha$. Let us analyze the resulting approximation factor β.

Proposition 26 *Every client $j \in \mathcal{C}$ has $r_j \leq \frac{\alpha t_j - \theta(j)}{\alpha - 1}$.*

Proof Let $s = \frac{\alpha t_j - \theta(j)}{\alpha - 1}$. It suffices to show that

$$\sum_{i\in\mathcal{F}, d(i,j)>s} a_{i,j} \leq 1 - 1/\alpha.$$

As $d(i,j) \geq \theta(j)$ for all $i \in \mathcal{F}$, we have

$$\sum_{\substack{i\in\mathcal{F}\\ d(i,j)>s}} a_{i,j} \leq \sum_{\substack{i\in\mathcal{F}\\ d(i,j)>s}} a_{i,j}\frac{d(i,j)-\theta(j)}{s-\theta(j)} \leq \sum_{i\in\mathcal{F}} a_{i,j}\frac{d(i,j)-\theta(j)}{s-\theta(j)} = \frac{\sum_{i\in\mathcal{F}} a_{i,j}d(i,j) - \theta(j)\sum_{i\in\mathcal{F}} a_{i,j}}{s-\theta(j)}$$
$$\leq \frac{t_j - \theta(j)}{s-\theta(j)} = 1 - 1/\alpha, \qquad \text{by (A1), (A2).}$$

■

Theorem 27 *Algorithm 9 gives an (α, β)-determinization with the following parameter β:*

1. *In the general setting, $\beta = \max(3, \frac{2\alpha}{\alpha-1})$.*
2. *In the SCC setting, $\beta = \frac{2\alpha}{\alpha-1}$.*

Proof We first claim that the resulting set $\mathcal{S}$ has $|\mathcal{S}| \leq \alpha k$. The algorithm opens at most $|C'|$ facilities. The sets F_j are pairwise disjoint for $j \in C'$ and $b(F_j) = 1/\alpha$ for $j \in C'$. Thus $\sum_{j \in C'} b(F_j) = |C'|/\alpha$. On the other hand, $b(\mathcal{F}) = k$, and so $k \geq |C'|/\alpha$.

Next, consider some $j \in \mathcal{C}$; we want to show that $d(j, \mathcal{S}) \leq \beta t_j$. By Observation 2, there is $z \in C'$ with $F_j \cap F_z \neq \emptyset$ and $\theta(z) + r_z \leq \theta(j) + r_j$. Thus $d(j, \mathcal{S}) \leq d(z, \mathcal{S}) + d(z, i) + d(j, i)$ where $i \in F_j \cap F_z$. Step (5) ensures $d(z, \mathcal{S}) = \theta(z)$. We have $d(z, i) \leq r_z$ and $d(i, j) \leq r_j$ since $i \in F_j \subseteq B(j, r_j)$ and $i \in F_z \subseteq B(z, r_z)$. So

$$d(j, \mathcal{S}) \leq \theta(z) + r_z + r_j \leq 2r_j + \theta(j).$$

By Proposition 26, we therefore have

$$d(j, \mathcal{S}) \leq \frac{2\alpha t_j - 2\theta(j)}{\alpha - 1} + \theta(j) = \frac{2\alpha t_j}{\alpha - 1} + \frac{\alpha - 3}{\alpha - 1}\theta(j) \tag{10}$$

This immediately shows the claim for the SCC setting where $\theta(j) = 0$.

In the general setting, for $\alpha \leq 3$, the second coefficient in the RHS of (10) is non-positive and hence the RHS is at most $\frac{2\alpha t_j}{\alpha-1}$ as desired. When $\alpha \geq 3$, then in order for t to be feasible we must have $t_j \geq \theta(j)$; substituting this upper bound on $\theta(j)$ into (10) gives

$$d(j, \mathcal{S}) \leq \frac{2\alpha t_j}{\alpha - 1} + \frac{\alpha - 3}{\alpha - 1} t_j = 3t_j$$

■

We note that these approximation ratios are, for α close to 1, within a factor of 2 compared to the lower bound of Observation 24. As $\alpha \to \infty$, the approximation ratio approaches to limiting values 3 (or 2 in the SCC setting).

6.2. The Case of Small β

We now consider what occurs when β becomes smaller than the critical threshold values 3 (or 2 in the SCC setting). We show that in this regime we must take $\alpha = \Omega(\log n)$. Of particular interest is the case when β approaches 1; here, in order to get $\beta = 1 + \epsilon$ for small ϵ we show it is necessary and sufficient to take $\alpha = \Theta(\frac{\log n}{\epsilon})$.

Proposition 28 *For any $\epsilon < 1/2$, there is a randomized polynomial-time algorithm to obtain a $(\frac{3 \log n}{\epsilon}, 1 + \epsilon)$ determinization.*

Proof First, let a, b be a solution to $\mathcal{P}_{\text{expectation}}$. Define $p_i = \min(1, \frac{2 \log n}{\epsilon} b_i)$ for each $i \in \mathcal{F}$ and form $\mathcal{S} = \text{DepRound}(p)$. Observe then that $|\mathcal{S}| \leq \lceil \sum_i p_i \rceil \leq \lceil \frac{2 \log n}{\epsilon} \sum b_i \rceil \leq 1 + \frac{2k \log n}{\epsilon} \leq \frac{3k \log n}{\epsilon}$.

For $j \in \mathcal{C}$, define $A = B_{j,(1+\epsilon)t_j}$. Let us note that, by properties (A3), (A1) and (A2), we have

$$\sum_{i \in A} b_i \geq \sum_{i \in A} a_{i,j} = 1 - \sum_{i : d(i,j) > (1+\epsilon)t_j} a_{i,j} \geq 1 - \sum_{i : d(i,j) > (1+\epsilon)t_j} a_{i,j} \frac{d(i,j)}{(1+\epsilon)t_j} \geq 1 - \frac{1}{1+\epsilon}$$

So by property (P3) of DepRound, and using the bound $\epsilon < 1/2$, have

$$\Pr[d(j,\mathcal{S}) > (1+\epsilon)t_j] = \Pr[A \cap \mathcal{S} = \emptyset] \leq \prod_{i \in A} (1-p_i) \leq \prod_{i \in A} e^{-\frac{2\log n}{\epsilon} b_i} \leq e^{\frac{-2\log n}{\epsilon}(1-\frac{1}{1+\epsilon})} \leq n^{-4/3}$$

A union bound over $j \in \mathcal{C}$ shows that solution set $\mathcal{S}$ satisfies $d(j,\mathcal{S}) \leq (1+\epsilon)t_j$ for all j with high probability. ■

The following shows matching lower bounds:

Proposition 29 *1. There is a universal constant K with the following properties. For any $k \geq 1, \epsilon \in (0, 1/3)$ there is some integer $N_{k,\epsilon}$ such that for $n > N_{k,\epsilon}$, there is a homogeneous SCC instance of size n in which* every $(\alpha, 1+\epsilon)$*-determinization satisfies $\alpha \geq \frac{K \log n}{\epsilon}$.*

2. For each $\beta \in (1,2)$ and each $k \geq 1$, there is a constant $K'_{\beta,k}$ such that, for all $n \geq 1$, there is a homogeneous SCC instance of size n in which every (α, β)*-determinization satisfies $\alpha \geq K'_{\beta,k} \log n$.*

3. For each $\beta \in (1,3)$ and each $k \geq 1$, there is a constant $K''_{\beta,k}$ such that, for all $n \geq 1$, there is a homogeneous instance of size n in which every (α, β)*-determinization satisfies $\alpha \geq K''_{\beta,k} \log n$*

Proof These three results are very similar, so we show the first one in detail and sketch the difference between the other two.

Consider an Erdős-Rényi random graph $G \sim \mathcal{G}(n,p)$, where $p = 3\epsilon/k$; note that $p \in (0,1)$. As shown by Glebov et al. (2015) asymptotically almost surely the domination number J of G satisfies $J = \Omega(\frac{k \log n}{\epsilon})$.

We construct a related instance with $\mathcal{F} = \mathcal{C} = [n]$, and where $d(i,j) = 1$ if (i,j) is an edge, and $d(i,j) = 2$ otherwise. Note that if X is not a dominating set of G, then some vertex of G has distance at least 2 from it; equivalently, $\max_j d(j,X) \geq 2$ for every set X with $|X| < J$.

Chernoff's bound shows that every vertex of G has degree at least $u = 0.9np$ with high probability. Assuming this event has occured, we calculate $\mathbf{E}[d(j,\mathcal{S})]$ where $\mathcal{S}$ is drawn from the uniform distribution on $\binom{\mathcal{F}}{k}$. Note that $d(j,\mathcal{S}) \leq 1$ if j is a neighbor of X and $d(j,\mathcal{S}) = 2$ otherwise, so

$$\mathbf{E}[d(j,\mathcal{S})] \leq 1 + \frac{\binom{n-u}{k}}{\binom{n}{k}} \leq 1 + e^{-0.9pk} = 1 + e^{-2.7\epsilon}.$$

Both the bound on the domination number and the minimum degree of G hold with positive probability for n sufficiently large (as a function of k, ϵ). In this case, $t_j = 1 + e^{-2.7\epsilon}$

is a feasible homogeneous demand vector. At the same time, every set $\mathcal{S} \in \binom{F}{J-1}$ satisfies $\min_{j \in \mathcal{C}} d(j, \mathcal{S}) \geq 2$. Thus, an (α, β)-determinization cannot have $\alpha < \frac{J}{k} = \Theta(\frac{\log n}{\epsilon})$ and $\beta \leq \frac{2}{1+e^{-2.7\epsilon}}$. Note that $\frac{2}{1+e^{-2.7\epsilon}} \geq 1 + \epsilon$ for $\epsilon < 1/3$. Thus, whenever $\beta \leq 1 + \epsilon$, we have $\alpha \geq \Theta(\frac{\log n}{\epsilon})$.

For the second result, we use the same construction as above with $p = 1 - \frac{1}{2}(\lambda/2)^{1/k}$ where $\lambda = 2 - \beta$. A similar analysis shows that the vector $t_j = 1 + \lambda/2$ is feasible with high probability and $|J| \geq \Omega(k \log n)$ (where the hidden constant may depend upon β, k). Thus, unless $\alpha \geq \Omega(\log n)$, the approximation ratio achieved is $\frac{2}{1+\lambda/2} \geq \beta$.

The third result is similar to the second one, except that we use a random bipartite graph. The left-nodes are associated with $\mathcal{F}$ and the right-nodes with $\mathcal{C}$. For $i \in \mathcal{F}$ and $j \in \mathcal{C}$, we define $d(i, j) = 1$ if (i, j) is an edge and $d(i, j) = 3$ otherwise. ■

6.3. The Case of $\alpha = 1$

We finally consider the case $\alpha = 1$, that is, where the constraint on the number of open facilities is respected *exactly*. By Observation 24, we must have $\beta \geq k + 1$ here. The following greedy algorithm gives a $(1, k + 2)$-determinization, nearly matching this lower bound.

Algorithm 10 $(1, k + 2)$-determinization algorithm

1: Initialize $\mathcal{S} = \emptyset$
2: **for** $\ell = 1, \ldots, |\mathcal{F}|$ **do**
3: Let $\mathcal{C}_\ell$ denote the set of points $j \in \mathcal{C}$ with $d(j, \mathcal{S}) > (k + 2)t_j$
4: If $\mathcal{C}_\ell = \emptyset$, then return $\mathcal{S}$.
5: Select the point $j_\ell \in \mathcal{C}_\ell$ with the smallest value of t_{j_ℓ}.
6: Update $\mathcal{S} \leftarrow \mathcal{S} \cup \{V_{j_\ell}\}$

Theorem 30 *If the values t_j are feasible, then Algorithm 10 outputs a $(1, k+2)$-determinization in $O(|\mathcal{F}||\mathcal{C}|)$ time.*

Proof For the runtime bound, we first compute V_j for each $j \in \mathcal{C}$; this requires $O(|\mathcal{F}||\mathcal{C}|)$ time upfront. When we update $\mathcal{S}$ at each iteration ℓ, we update and maintain the quantities $d(j, \mathcal{S})$ quantities by computing $d(j, V_{j_\ell})$ for each $j \in \mathcal{C}$. This takes $O(|\mathcal{C}|)$ time per iteration.

To show correctness, note that if this procedure terminates at iteration ℓ, we have $\mathcal{C}_\ell = \emptyset$ and so every point $j \in \mathcal{C}$ has $d(j, \mathcal{S}) \leq (k + 2)t_j$. The resulting set $\mathcal{S}$ at this point has cardinality $\ell - 1$. So we need to show that the algorithm terminates before reaching iteration $\ell = k + 2$.

Suppose not; let the resulting points be $j_1, \ldots, j_{k+1}$ and for each $\ell = 1, \ldots, k + 1$ let $w_\ell = t_{j_\ell}$. Because j_ℓ is selected to minimze t_{j_ℓ} we have $w_1 \leq w_2 \leq \cdots \leq w_{k+1}$.

Now, let Ω be a k-lottery satisfying $\mathbf{E}_{\mathcal{S} \sim \Omega}[d(j, \mathcal{S})] \leq t_j$ for every $j \in \mathcal{C}$, and consider the random process of drawing $\mathcal{S}$ from Ω. Define the random variable $D_\ell = d(j_\ell, \mathcal{S})$ for $\ell = 1, \ldots, k + 1$. For any such $\mathcal{S}$, by the pigeonhole principle there must exist some pair $j_\ell, j_{\ell'}$ with $1 \leq \ell < \ell' \leq k + 1$ which are both matched to a common facility $i \in \mathcal{S}$, that is

$$D_\ell = d(j_\ell, \mathcal{S}) = d(j_\ell, i), D_{\ell'} = d(j_{\ell'}, \mathcal{S}) = d(j_{\ell'}, i).$$

By the triangle inequality,

$$d(j_{\ell'}, V_{j_\ell}) \leq d(j_{\ell'}, i) + d(i, j_\ell) + d(j_\ell, V_{j_\ell}) = D_{\ell'} + D_\ell + \theta(j_\ell)$$

On the other hand, $j_{\ell'} \in C_{\ell'}$ and yet V_{j_ℓ} was in the partial solution set $\mathcal{S}$ seen at iteration ℓ'. Therefore, it must be that

$$d(j_{\ell'}, V_{j_\ell}) > (k+2)t_{j_{\ell'}} = (k+2)w_{\ell'}$$

Putting these two inequalities together, we have shown that

$$D_\ell + D_{\ell'} + \theta(j_\ell) > (k+2)w_{\ell'}.$$

As $\theta(j_\ell) \leq w_\ell \leq w_{\ell'}$, this implies that

$$\frac{D_\ell}{w_\ell} + \frac{D_{\ell'}}{w_{\ell'}} \geq \frac{D_\ell + D_{\ell'}}{w_{\ell'}} > \frac{(k+2)w_{\ell'} - \theta(j_\ell)}{w_{\ell'}} \geq \frac{(k+2)w_{\ell'} - w_\ell}{w_{\ell'}} \geq \frac{(k+2)w_{\ell'} - w_{\ell'}}{w_{\ell'}} = k+1.$$

We have shown that, with probability one, there is some pair $\ell < \ell'$ satisfying this inequality $D_\ell/w_\ell + D_{\ell'}/w_{\ell'} > k+1$. Therefore, with probability one it holds that

$$\sum_{\ell=1}^{k+1} D_\ell/w_\ell > k+1. \tag{11}$$

But now take expectations, observing that $\mathbf{E}[D_\ell] = \mathbf{E}[d(j_\ell, \mathcal{S})] \leq t_{j_\ell} = w_\ell$. So the LHS of (11) has expectation at most $k+1$. This is a contradiction. ■

We remark that it is possible to obtain an optimal $(1, k+1)$-determinization algorithm for the SCC or homogeneous settings, but we omit this since it is very similar to Algorithm 10.

7. Acknowledgments

Our sincere thanks to Brian Brubach and to the anonymous referees, for many useful suggestions and for helping to tighten the focus of the paper.

Appendix A. Proof of Pseudo-Theorem 17

We would like to use Lemma 16 to bound $\mathbf{E}[d(i, \mathcal{S})]$, over all possible integer values $m \geq 1$ and over all possible sequences $1 \geq u_1 \geq u_2 \geq u_3 \geq \cdots \geq u_t \geq 0$. One technical obstacle here is that this is not a compact space, due to the unbounded dimension t and unbounded parameter m. The next result removes these restrictions.

Proposition 31 *For any fixed integers $L, M \geq 1$, and every $j \in \mathcal{C}$, we have*

$$\mathbf{E}[d(j, \mathcal{S})] \leq 1 + \max_{\substack{m \in \{1,2,\ldots,M\} \\ 1 \geq u_1 \geq u_2 \geq \ldots u_L \geq 0}} \mathbf{E}_Q \hat{R}(m, u_1, u_2, \ldots, u_L),$$

where we define

$$\alpha = \prod_{\ell=1}^{L-1}(1-\overline{Q_p}(u_\ell - u_{\ell+1})) \times e^{-\overline{Q_p}u_L},$$

$$\beta = \prod_{\ell=1}^{L-1}(\overline{u_\ell} + \overline{Q_p}u_{\ell+1}) \times \begin{cases} (1-u_L) & \text{if } u_L \le Q_p \\ e^{-\frac{u_L - Q_p}{1-Q_p}}(1-Q_p) & \text{if } u_L > Q_p \end{cases},$$

$$\hat{R}(m, u_1, \ldots, u_L) = \begin{cases} (1-\overline{Q_f}\frac{\overline{u_1}}{m})^m \alpha + (\overline{Q_f}(1-\frac{\overline{u_1}}{m}))^m \beta & \text{if } m < M \\ e^{-\overline{Q_f}\overline{u_1}}\alpha + \overline{Q_f}^M e^{-\overline{u_1}}\beta & \text{if } m = M \end{cases}.$$

The expectation $\mathbf{E}_Q$ *is taken only over the randomness involved in* Q_f, Q_p.

Proof By Lemma 16,

$$\mathbf{E}[d(i,\mathcal{S}) \mid Q_{\mathrm{f}}, Q_{\mathrm{p}}]$$
$$\le 1 + \left(1 - \overline{Q_{\mathrm{f}}}\frac{\overline{u_1}}{m}\right)^m \prod_{\ell=1}^{t}(1-\overline{Q_{\mathrm{p}}}(u_\ell - u_{\ell+1})) + \left(\overline{Q_{\mathrm{f}}}(1-\frac{\overline{u_1}}{m})\right)^m \prod_{\ell=1}^{t}(\overline{u_\ell} + \overline{Q_{\mathrm{p}}}u_{\ell+1}).$$

where $u_1, \ldots, u_t, m$ are defined as in Lemma 16; in particular $1 \ge u_1 \ge u_2 \ge \cdots \ge u_t \ge u_{t+1} = 0$ and $m \ge 1$. If we define $u_j = 0$ for all integers $j \ge t$, then

$$\mathbf{E}[d(i,\mathcal{S}) \mid Q_{\mathrm{f}}, Q_{\mathrm{p}}]$$
$$\le 1 + \left(1 - \overline{Q_{\mathrm{f}}}\frac{\overline{u_1}}{m}\right)^m \prod_{\ell=1}^{\infty}(1-\overline{Q_{\mathrm{p}}}(u_\ell - u_{\ell+1})) + \left(\overline{Q_{\mathrm{f}}}(1-\frac{\overline{u_1}}{m})\right)^m \prod_{\ell=1}^{\infty}(\overline{u_\ell} + \overline{Q_{\mathrm{p}}}u_{\ell+1}).$$

The terms corresponding to $\ell > L$ telescope so we estimate these as:

$$\prod_{\ell=L}^{\infty}(1-\overline{Q_{\mathrm{p}}}(u_\ell - u_{\ell+1})) \le \prod_{\ell=L}^{\infty} e^{-\overline{Q_{\mathrm{p}}}(u_\ell - u_{\ell+1})} = e^{-\overline{Q_{\mathrm{p}}}u_L}$$

and

$$\prod_{\ell=L}^{\infty}(\overline{u_\ell} + \overline{Q_{\mathrm{p}}}u_{\ell+1}) \le (1 - u_L + \overline{Q_{\mathrm{p}}}u_{L+1}) \prod_{\ell=L+1}^{\infty} e^{-u_\ell + \overline{Q_{\mathrm{p}}}u_{\ell+1}} \le (1 - u_L + \overline{Q_{\mathrm{p}}}u_{L+1})e^{-u_{L+1}}.$$

Now consider the expression $(1-u_L+\overline{Q_{\mathrm{p}}}u_{L+1})e^{-u_{L+1}}$ as a function of u_{L+1} in the range $u_{L+1} \in [0, u_L]$. Elementary calculus shows that it satisfies the bound

$$(1 - u_L + \overline{Q_{\mathrm{p}}}u_{L+1})e^{-u_{L+1}} \le \begin{cases} (1-u_L) & \text{if } u_L \le Q_{\mathrm{p}} \\ e^{-\frac{u_L - Q_{\mathrm{p}}}{1-Q_{\mathrm{p}}}}(1-Q_{\mathrm{p}}) & \text{if } u_L > Q_{\mathrm{p}} \end{cases},$$

Thus

$$\mathbf{E}[d(i,\mathcal{S}) \mid Q_{\mathrm{f}}, Q_{\mathrm{f}}] \le 1 + \left(1 - \overline{Q_{\mathrm{f}}}\frac{\overline{u_1}}{m}\right)^m \alpha + \left(\overline{Q_{\mathrm{f}}}(1-\frac{\overline{u_1}}{m})\right)^m \beta.$$

If $m < M$ we are done. Otherwise, for $m \geq M$, we upper-bound the Q_{f} terms as:

$$(1 - \overline{Q_{\mathrm{f}}}\,\overline{u_1}/m)^m \leq e^{-\overline{Q_{\mathrm{f}}}\,\overline{u_1}}, \qquad (\overline{Q_{\mathrm{f}}}(1 - \overline{u_1}/m))^m \leq \overline{Q_{\mathrm{f}}}^M e^{-\overline{u_1}}$$ ■

In light of Proposition 31, we can get an upper bound on $\mathbf{E}[d(i, \mathcal{S})]$ by maximizing $\hat{R}$. We now discuss to bound $\hat{R}$ for a fixed choice of L, M, where we select $Q_{\mathrm{f}}, Q_{\mathrm{p}}$ according to the following type of distribution:

$$(Q_{\mathrm{f}}, Q_{\mathrm{p}}) = \begin{cases} (\gamma_{0,\mathrm{f}}, 0) & \text{with probability } p \\ (\gamma_{1,\mathrm{f}}, \gamma_{1,\mathrm{p}}) & \text{with probability } 1 - p \end{cases}.$$

Note that we are setting $\gamma_{0,\mathrm{p}} = 0$ here in order to keep the search space more manageable. We need to upper-bound the function $\mathbf{E}_Q \hat{R}(m, u_1, \ldots, u_L)$, for fixed values γ and where m, u range over the compact domain $m \in \{1, \ldots, M\}, 1 \geq u_1 \geq \cdots \geq u_L \geq 0$. The most straightforward way to do so would be to divide $(u_1, \ldots, u_L)$ into intervals of size ϵ, and then calculate upper bounds on $\hat{R}$ within each interval and for each m. This process would have a runtime of $M\epsilon^{-L}$, which is too expensive.

But suppose we have fixed $u_j, \ldots, u_L$, and we wish to continue to enumerate over $u_1, \ldots, u_{j-1}$. To compute $\hat{R}(m, u_1, \ldots, u_L)$ as a function of $m, u_1, \ldots, u_L$, observe that we do not need to know all the values $u_{j+1}, \ldots, u_L$, but only the following four summary statistics of them:

1. $a_1 = e^{-\overline{\gamma_{1,\mathrm{p}}} u_L} \prod_{\ell=j}^{L-1} (1 - \overline{\gamma_{1,\mathrm{p}}}(u_\ell - u_{\ell+1}))$,
2. $a_2 = \prod_{\ell=j}^{L-1} (\overline{u_\ell} + \overline{\gamma_{1,\mathrm{p}}} u_{\ell+1}) \times \begin{cases} (1 - u_L) & \text{if } u_L \leq \gamma_{1,\mathrm{p}} \\ e^{-\frac{u_L - \gamma_{1,\mathrm{p}}}{1 - \gamma_{1,\mathrm{p}}}} (1 - \gamma_{1,\mathrm{p}}) & \text{if } u_L > \gamma_{1,\mathrm{p}} \end{cases}$,
3. $a_3 = e^{-u_L} \prod_{\ell=j}^{L-1} (1 - (u_\ell - u_{\ell+1}))$,
4. $a_4 = u_{j+1}$.

We thus use a dynamic program wherein we track, for $j = L, \ldots, 1$, all possible values for the tuple $(a_1, \ldots, a_4)$. Furthermore, since $\hat{R}$ is a monotonic function of a_1, a_2, a_3, a_4, we only need to store the *maximal* tuples $(a_1, \ldots, a_4)$. The resulting search space has size $O(\epsilon^{-3})$.

We wrote C code to perform this computation to upper-bound $\mathbf{E}_Q \hat{R}$ with $M = 10, \epsilon = 2^{-12}, L = 7$. This runs in about an hour on a single CPU core. With some additional tricks, we can also optimize over the parameter $p \in [0, 1]$ while still keeping the stack space bounded by $O(\epsilon^{-3})$.

Note that due to the complexity of the dynamic programming algorithm, the computations were carried out in double-precision floating point arithmetic. The rounding errors were not tracked precisely, and it would be difficult to write completely correct code to do so. We believe that these errors should be orders of magnitude below the third decimal place, and that the computed value 1.592 is a valid upper bound.

References

Sara Ahmadian, Ashkan Norouzi-Fard, Ola Svensson, and Justin Ward. Better guarantees for k-means and Euclidean k-median by primal-dual algorithms. In *Proc. 58th annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 61–72, 2017.

Sharareh Alipour and Amir Jafari. Improvements on the k-center problem for uncertain data. In *Proc. 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principle of Database System (PODS)*, pages 425–433, 2018.

Ian Ayres, Mahzarin Banaji, and Christine Jolls. Race effects on eBay. *The RAND Journal of Economics*, 46(4):891–917, 2015. ISSN 1756-2171. doi: 10.1111/1756-2171.12115. URL `http://dx.doi.org/10.1111/1756-2171.12115`.

Emily Badger. How Airbnb plans to fix its racial-bias problem. *Washington Post*, 2016. URL `https://www.washingtonpost.com/news/wonk/wp/2016/09/08/how-airbnb-plans-to-fix-its-racial-bias-problem/`. September 8, 2016.

Marianne Bertrand and Sendhil Mullainathan. Are Emily and Greg More Employable Than Lakisha and Jamal? A Field Experiment on Labor Market Discrimination. *American Economic Review*, 94(4):991–1013, 2004. doi: 10.1257/0002828042002561. URL `http://www.aeaweb.org/articles.php?doi=10.1257/0002828042002561`.

Jarosław Byrka, Thomas Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. An improved approximation for k-median, and positive correlation in budgeted optimization. *ACM Transaction on Algorithms*, 13(2):23:1–23:31, 2017.

Moses Charikar and Shi Li. A dependent LP-rounding approach for the k-median problem. *Automata, Languages, and Programming*, pages 194–205, 2012.

Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets. In *Advances in Neural Information Processing Systems 30*, pages 5036–5044, 2017.

Vincent Cohen-Addad, Philip N Klein, and Claire Mathieu. Local search yields approximation schemes for k-means and k-median in Euclidean and minor-free metrics. In *Proc. 57th annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 353–364, 2016.

Amit Datta, Michael Carl Tschantz, and Anupam Datta. Automated experiments on ad privacy settings: A tale of opacity, choice, and discrimination. In *Proceedings on Privacy Enhancing Technologies*, pages 92–112, 2015.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proc. 3rd Innovations in Theoretical Computer Science conference (ITCS)*, pages 214–226, 2012.

Laura P Edgerley, Yasser Y El-Sayed, Maurice L Druzin, Michaela Kiernan, and Kay I Daniels. Use of a community mobile health van to increase early access to prenatal care. *Maternal and Child Health Journal*, 11(3):235–239, 2007.

Dan Feldman, Morteza Monemizadeh, and Christian Sohler. A PTAS for k-means clustering based on weak coresets. In *Proc. 23rd ACM Symposium on Computational Geometry (SOCG)*, pages 11–18, 2007.

Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *Proc. 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 259–268, 2015.

Roman Glebov, Anita Liebenau, and Tibor Szabó. On the concentration of the domination number of the random graph. *SIAM J. Discrete Math.*, 29(3):1186–1206, 2015. doi: 10.1137/12090054X. URL `https://doi.org/10.1137/12090054X`.

Sudipto Guha and Samir Khuller. Greedy strikes back: improved facility location algorithms. *Journal of algorithms*, 31(1):228–248, 1999.

Sepali Guruge, JoAnne Hunter, Keegan Barker, Mary Jane McNally, and Lilian Magalhães. Immigrant womens experiences of receiving care in a mobile health clinic. *Journal of Advanced Nursing*, 66(2):350–359, 2010.

David G. Harris, Thomas Pensyl, Aravind Srinivasan, and Khoa Trinh. Dependent rounding for knapsack/partition constraints and facility location. *arxiv*, abs/1709.06995, 2017.

David G Harris, Thomas Pensyl, Aravind Srinivasan, and Khoa Trinh. A lottery model for center-type problems with outliers. *ACM Transactions on Algorithms*, 15(3):Article #36, 2019.

Dorit S. Hochbaum and David B. Shmoys. A unified approach to approximation algorithms for bottleneck problems. *J. ACM*, 33(3):533–550, 1986. doi: 10.1145/5925.5933. URL `http://doi.acm.org/10.1145/5925.5933`.

Lingxiao Huang and Jian Li. Stochastic k-center and j-flat-center problems. In *Proc. 28th annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 110–129, 2017.

Imanol Arrieta Ibarra, Leonard Goff, Diego Jiménez Hernández, Jaron Lanier, and E. Glen Weyl. Should we treat data as labor? Moving beyond 'free'. *American Economic Association Papers & Proceedings*, 1(1), 2018.

Kamal Jain, Mohammad Mahdian, and Amin Saberi. A new greedy approach for facility location problems. In *Proc. 34th annual ACM Symposium on Theory of Computing (STOC)*, pages 731–740, 2002.

Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. A local search approximation algorithm for k-means clustering. *Comput. Geom.*, 28(2-3):89–112, 2004. doi: 10.1016/j.comgeo.2004.03.003. URL `https://doi.org/10.1016/j.comgeo.2004.03.003`.

David Kempe, Jon M. Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. *Theory of Computing*, 11:105–147, 2015.

Ravishankar Krishnaswamy, Shi Li, and Sai Sandeep. Constant approximation for k-median and k-means with outliers via iterative rounding. In *Proc. 50th annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 646–659, 2018.

J. Lanier. *Who Owns the Future?* Simon Schuster, 2014.

Jiří Matoušek. On approximate geometric k-clustering. *Discrete & Computational Geometry*, 24(1):61–84, 2000. doi: 10.1007/s004540010019. URL `https://doi.org/10.1007/s004540010019`.

John F McCarthy, Frederic C Blow, Marcia Valenstein, Ellen P Fischer, Richard R Owen, Kristen L Barry, Teresa J Hudson, and Rosalinda V Ignacio. Veterans affairs health system and mental health treatment retention among patients with serious mental illness: evaluating accessibility and availability barriers. *Health services research*, 42(3p1):1042–1060, 2007.

Cathleen Mooney, Jack Zwanziger, Ciaran S Phibbs, and Susan Schmitt. Is travel distance a barrier to veterans' use of VA hospitals for medical surgical care? *Social science & medicine*, 50(12):1743–1755, 2000.

Dana Moshkovitz. The projection games conjecture and the NP-hardness of $\ln n$-approximating set-cover. *Theory of Computing*, 11(7):221–235, 2015.

David B Reuben, Lawrence W Bassett, Susan H Hirsch, Catherine A Jackson, and Roshan Bastani. A randomized clinical trial to assess the benefit of offering on-site mobile mammography in addition to health education for older women. *American Journal of Roentgenology*, 179(6):1509–1514, 2002.

Susan K Schmitt, Ciaran S Phibbs, and John D Piette. The influence of distance on utilization of outpatient mental health aftercare following inpatient substance abuse treatment. *Addictive Behaviors*, 28(6):1183–1192, 2003.

Kevin A. Schulman, Jesse A. Berlin, William Harless, Jon F. Kerner, Shyrl Sistrunk, Bernard J. Gersh, Ross Dub, Christopher K. Taleghani, Jennifer E. Burke, Sankey Williams, John M. Eisenberg, William Ayers, and Jos J. Escarce. The effect of race and sex on physicians' recommendations for cardiac catheterization. *New England Journal of Medicine*, 340(8):618–626, 1999.

Aravind Srinivasan. Distributions on level-sets with applications to approximation algorithms. In *Proc. 42nd annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 588–597, 2001.

Richard S. Zemel, Yu Wu, Kevin Swersky, Toniann Pitassi, and Cynthia Dwork. Learning fair representations. In *Proc. 30th International Conference on Machine Learning (ICML)*, pages 325–333, 2013.