

The Non-Uniform k -Center Problem

DEEPARNAB CHAKRABARTY, Dartmouth College, USA

PRACHI GOYAL, Microsoft, USA

RAVISHANKAR KRISHNASWAMY, Microsoft Research, India

In this article, we introduce and study the Non-Uniform k -Center (NUkC) problem. Given a finite metric space (X, d) and a collection of balls of radii $\{r_1 \geq \dots \geq r_k\}$, the NUkC problem is to find a placement of their centers in the metric space and find the minimum dilation α , such that the union of balls of radius $\alpha \cdot r_i$ around the i th center covers all the points in X . This problem naturally arises as a min-max vehicle routing problem with fleets of different speeds.

The NUkC problem generalizes the classic k -center problem, wherein all the k radii are the same (which can be assumed to be 1 after scaling). It also generalizes the k -center with outliers (kCwO for short) problem, in which there are k balls of radius 1 and ℓ (number of outliers) balls of radius 0. Before this work, there was a 2-approximation and 3-approximation algorithm known for these problems, respectively; the former is best possible unless $P=NP$.

We first observe that no $O(1)$ -approximation to the optimal dilation is possible unless $P=NP$, implying that the NUkC problem is harder than the above two problems. Our main algorithmic result is an $(O(1), O(1))$ -bi-criteria approximation result: We give an $O(1)$ -approximation to the optimal dilation; however, we may open $\Theta(1)$ centers of each radii. Our techniques also allow us to prove a simple (uni-criterion), optimal 2-approximation to the kCwO problem improving upon the long-standing 3-factor approximation for this problem.

Our main technical contribution is a connection between the NUkC problem and the so-called firefighter problems on trees that have been studied recently in the TCS community. We show NUkC is at least as hard as the firefighter problem. While we do not know whether the converse is true, we are able to adapt ideas from recent works [1, 3] in non-trivial ways to obtain our constant factor bi-criteria approximation.

CCS Concepts: • **Theory of computation** → **Facility location and clustering**;

Additional Key Words and Phrases: Clustering algorithms, outliers, firefighting on trees

ACM Reference format:

Deeparnab Chakrabarty, Prachi Goyal, and Ravishankar Krishnaswamy. 2020. The Non-Uniform k -Center Problem. *ACM Trans. Algorithms* 16, 4, Article 46 (June 2020), 19 pages.

<https://doi.org/10.1145/3392720>

An extended abstract appeared in the Proceedings of ICALP 2016.

Work done D. Chakrabarty and P. Goyal while at Microsoft Research, India.

Authors' addresses: D. Chakrabarty, Dartmouth College, 9, Maynard St., Hanover, NH 03755; email: deeparnab@dartmouth.edu; P. Goyal, Microsoft, 15010 NE 36th St, Redmond, WA 98052; email: prgoy@microsoft.com; R. Krishnaswamy, Microsoft Research, 9, Vigyan 1st floor, Lavelle Road, Bengaluru, Karnataka 560001, India; email: rakri@microsoft.com.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2020 Association for Computing Machinery.

1549-6325/2020/06-ART46 \$15.00

<https://doi.org/10.1145/3392720>

1 INTRODUCTION

Source location and vehicle routing problems are extremely well studied [9, 19, 23] in operations research. Consider the following location+routing problem: We are given a set of k ambulances with speeds s_1, s_2, \dots, s_k , respectively, and we have to find the depot locations for these vehicles in a metric space (X, d) such that any point in the space can be served by some ambulance as fast as possible. If all speeds were the same, then we would place the ambulances in locations S such that $\max_{v \in X} d(v, S)$ is minimized—this is the famous k -center problem. Differing speeds, however, leads to non-uniformity, thus motivating the problem we consider.

Definition 1.1 (The Non-Uniform k -Center Problem (NUkC)). The input to the problem is a metric space (X, d) and a collection of k radii $\{r_1 \geq r_2 \geq \dots \geq r_k\}$. The objective is to find a placement $C \subseteq X$ of the centers of these balls, so as to minimize the dilation parameter α such that the union of balls of radius $\alpha \cdot r_i$ around the i th center covers all of X . Equivalently, we need to find centers $\{c_1, \dots, c_k\}$ to minimize $\max_{v \in X} \min_{i=1}^k \frac{d(v, c_i)}{r_i}$.

As mentioned, when all r_i 's are the same (and equal to 1 by scaling), we get the k -center problem. The k -center problem was originally studied by Gonzalez [10] and Hochbaum and Shmoys [13] as a clustering problem of partitioning a metric space into different clusters to minimize the maximum intra-cluster distances. One issue (see Figure 1 for an illustration and refer to Reference [11] for a more detailed explanation) with k -center (and also k -median/means) as an objective function for clustering is that it favors clusters of similar sizes with respect to cluster radii. However, in the presence of qualitative information on the differing cluster sizes, the non-uniform versions of the problem can arguably provide more nuanced solutions. One such extreme special case was considered as the “clustering with outliers” problem [7] where some fixed number, say, ℓ , of points in the metric space need not be covered by the clusters. In particular, Charikar et al. [7] consider (among other problems) the k -center with outlier problem (kCwO, for short) and give a 3-approximation for this problem. It is easy to see that kCwO is a special case of the NUkC problem, where there are k balls of radius 1 and ℓ balls of radius 0.

Motivated by the aforementioned reasons (both from facility location as well as from clustering settings), we investigate the worst-case complexity of the NUkC problem. Gonzalez [10] and Hochbaum and Shmoys [13] give 2-approximations for the k -center problem and also show that no better factor is possible unless $P = NP$. Charikar et al. [7] give a 3-approximation for the kCwO problem, and this has been the best factor known for 15 years. Given these algorithms, it is natural to wonder if a simple $O(1)$ -approximation exists for the NUkC problem. In fact, our first result shows a qualitative distinction between NUkC and these problems: Constant-approximations are impossible for NUkC unless $P=NP$.

THEOREM 1.2. *Unless $P = NP$, the NUkC problem does not admit a $c(n)$ -factor approximation unless $c(n) \gg 2^{\text{poly}(n)}$, even when the underlying metric is a tree metric.*

The hardness result is by a reduction from the so-called *resource minimization for fire containment* problem on trees (RMFC-T, in short), a variant of the firefighter problem. To complement the above hardness result, we give the following bi-criteria approximation algorithm that is the main result of the article and that further highlights the connections with RMFC-T, since our algorithms heavily rely on the recent algorithms for RMFC-T [1, 3]. An (a, b) -factor bi-criteria algorithm for NUkC returns a solution that places at most a balls of each type (thus in total it may use as many as $a \cdot k$ balls), and the dilation is at most b times the optimum dilation for the instance that places exactly one ball of each type.

THEOREM 1.3. *There is an $(O(1), O(1))$ -factor bi-criteria algorithm for the NUkC problem.*

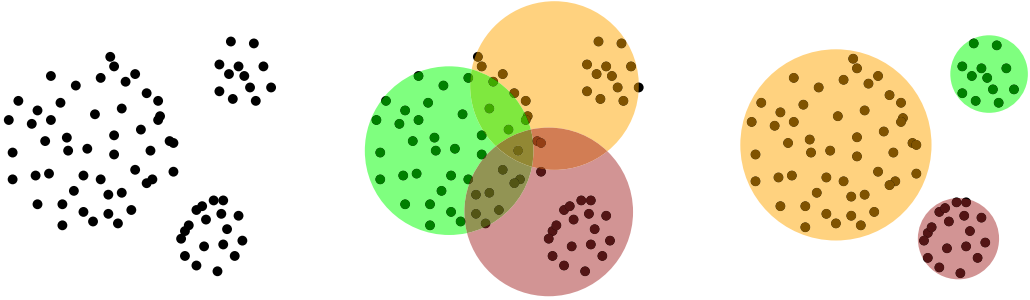


Fig. 1. The left figure shows the dataset, the middle figure shows a traditional k -center clustering, and the right figure depicts a non-uniform clustering.

Furthermore, as we elucidate below, our techniques also give uni-criteria results when the number of distinct radii is 2. In particular, we get a 2-approximation for the kCwO problem and a $(1 + \sqrt{5})$ -approximation when there are only two distinct types of radii.

THEOREM 1.4. *There is a 2-approximation for the kCwO problem.*

THEOREM 1.5. *There is a $(1 + \sqrt{5})$ -approximation for the NUKC problem when the number of distinct radii is at most 2.*

1.1 Discussion on Techniques

Our proofs of Theorems 1.2 and 1.3 show a strong connection between NUKC and the so-called *resource minimization for fire containment* problem on trees (RMFC-T, in short). This connection is one of the main findings of the article, so we first formally define this problem.

Definition 1.6 (Resource Minimization for Fire Containment on Trees (RMFC-T)). Given a rooted tree T as input, the goal is to select a collection of non-root nodes N from T such that (a) every root-leaf path has at least one vertex from N , and (b) $\max_t |N \cap L_t|$ is minimized, where L_t is the t th layer of T , that is, the vertices of T at exactly distance t from the root.

To understand the reason behind the name, consider a fire starting at the root spreading to neighboring vertices each day; the RMFC-T problem minimizes the maximum number of fire-fighters needed on any given day so as to prevent the fire spreading to the leaves of T .

It is NP-complete to decide whether the optimum of RMFC-T is at most 1 [8, 17]. Given any RMFC-T instance and any $c > 1$, we construct an NUKC instance on a tree metric such that in the “yes” case there is always a placement with dilation = 1 that covers the metric, while in the “no” case even a dilation of c does not help. Upon understanding our hardness construction, the inquisitive reader may wonder if the reduction also works in the other direction, i.e., whether we can solve NUKC using a reduction to RMFC-T problem. Unfortunately, we do not know whether this is true even for two types of radii. However, as we explain below, we can still use positive results for the RMFC-T problem to design good algorithms for the NUKC problem.

We start off by considering the natural LP relaxation for the NUKC problem and describe an *LP-aware reduction* of NUKC to RMFC-T. More precisely, given a feasible solution to the LP-relaxation for the given NUKC instance, we describe a procedure to obtain an instance of RMFC-T defined by a tree T , with the following properties: (i) We can exhibit a feasible fractional solution for the LP relaxation of the RMFC-T instance and (ii) given any feasible *integral* solution to the RMFC-T instance, we can obtain a feasible integral solution to the NUKC instance that dilates the radii by

at most a constant factor. Therefore, an LP -based ρ -approximation to RMFC-T would immediately imply $(\rho, O(1))$ -bicriteria approximation algorithms for NUKC. This already implies Theorem 1.4 and Theorem 1.5, since the corresponding RMFC-T instances have no integrality gap. Also, using a result of Chalermsook and Chuzhoy [3], we directly obtain an $(O(\log^* n), O(1))$ -bicriteria approximation algorithm for NUKC.

Here we reach a technical bottleneck: Chalermsook and Chuzhoy [3] also show that the integrality gap of the natural LP relaxation for RMFC-T is $\Omega(\log^* n)$. Therefore, the above approach cannot¹ give us an $(O(1), O(1))$ -bicriteria approximation.

To get an $(O(1), O(1))$ -algorithm, we use the $O(1)$ -approximation for the RMFC-T problem by Adjiashvili et al. [1]. At a very high level, the main technique in [1] is the following. Given an RMFC-T instance, they carefully and efficiently “guess” a subset of the optimum solution, such that the natural LP-relaxation for covering the uncovered leaves has $O(1)$ -integrality gap. However, this guessing procedure crucially uses the tree structure of T in the RMFC-T problem. Unfortunately for us though, we get the RMFC-T tree only after solving the LP for NUKC, which already has an $\Omega(\log^* n)$ -gap! Nevertheless, inspired by the ideas in [1], we show that we can also efficiently preprocess an NUKC instance, “guessing” the positions of a certain number of balls in an optimum solution, such that the standard LP-relaxation for covering the uncovered points indeed has $O(1)$ integrality gap. We can then invoke the LP-aware embedding reduction to RMFC-T at this juncture to solve our problem. This is quite delicate, and is the most technically involved part of the article.

1.2 Related Work

The k -center problem [10, 13] and the k -center with outliers [7] problems are classic problems in approximation algorithms and clustering. These problems have also been investigated under various settings such as the incremental model [5, 22], streaming model [4, 22], and more recently in the map-reduce model [14, 21]. Similarly, the k -median [2, 6, 15, 20] and k -means [12, 15, 16, 18] problems are also classic problems studied extensively in approximation algorithms and clustering. The generalization of k -median to a routing+location problem was also studied recently [9]. It would be interesting to explore the complexity of the non-uniform versions of these problems. Another direction would be to explore if the new non-uniform model can be useful in solving clustering problems arising in practice.

2 HARDNESS REDUCTION

In this section, we prove Theorem 1.2 based on the following NP-hardness [17] for RMFC-T.

THEOREM 2.1. [17] *Given a tree T whose leaves are at the same distance from the root, it is NP-hard to distinguish between the following two cases. YES: There is a solution to the RMFC-T instance of value 1. NO: All solutions to the RMFC-T instance have value 2.*

Suppose, for contradiction’s sake, there is a $c(n)$ -factor approximation algorithm for NUKC for some function $c(n) \leq 2^{p(n)}$ for some fixed polynomial p . Given an RMFC-T instance defined by tree T , we now describe the construction of our NUKC instance. Let n be the number of nodes in T . Let $h \leq n$ be the height of the tree, and let L_t denote the vertices of the tree at distance exactly t from the root. So the leaves constitute L_h , since all leaves are at the same distance from the root. Let

¹Indeed, our hardness reduction in Theorem 1.2 can be generalized to obtain an $(\Omega(\log^* n), c)$ integrality gap for any constant $c > 1$ for the natural LP relaxation for NUKC.

$c = \lceil c(n) \rceil + 1$ be any fixed constant strictly larger than the desired approximation factor function evaluated at n .

The NUKC instance, $\mathcal{I}(T)$, is defined by the metric space (X, d) , and a collection of balls. The points in our metric space will correspond to the leaves of the tree, i.e., $X = L_h$, and thus $|X| \leq n$. To define the metric, we assign a weight $d(e) = (2c + 1)^{h-i+1}$ for each edge whose one endpoint is in L_i and the other in L_{i-1} ; we then define d be the shortest-path metric on X induced by this weighted tree. Finally, we set $k = h$, and define the k radii $r_1 \geq r_2 \geq \dots \geq r_k$ iteratively as follows: Define $r_k := 0$, and for $k \geq i > 1$, set $r_{i-1} := (2c + 1) \cdot r_i + 2(2c + 1)$. This completes the NUKC instance. Before proceeding we make the simple observation: For any two leaves u and u' with lca $v \in L_t$, we have $d(u, u') = 2(2c + 1) + (2c + 1)^2 + \dots + (2c + 1)^{h-t} = r_t$. Thus, the maximum distance between two points is $\leq (2n)(2c)^n \leq 2^{\text{poly}(n)}$, implying the size of the number is polynomially bounded, which in turn implies an efficient reduction.

The following lemma, therefore, proves Theorem 1.2. If there was a $c(n)$ -approximation algorithm for NUKC, when run on $\mathcal{I}(T)$, then it would distinguish between the following two cases.

LEMMA 2.2. *If T is the YES case of Theorem 2.1, then $\mathcal{I}(T)$ has optimum dilation = 2. If T is the NO case of Theorem 2.1, then $\mathcal{I}(T)$ has optimum dilation $\geq 2c$.*

PROOF. Suppose T is in the YES case, and there is a solution to RMFC-T, which selects at most 1 node from each level L_t . If $v \in L_t$ is selected, then select a center c_v arbitrarily from any leaf in the sub-tree rooted at v and open the ball of radius r_t . We now need to show all points in $X = L_h$ are covered by these balls. Let u be any leaf; there must be a vertex v in some level L_t in u 's path to the root such that a ball of radius r_t is opened at c_v . However, $d(u, c_v) \leq d(u, v) + d(v, c_v) = 2r_t$ and so the ball of radius $2r_t$ around c_v covers u .

Now suppose T is in the NO case, and the NUKC instance has a solution with optimum dilation $< 2c$. We build a good solution for the RMFC-T instance N as follows: Suppose the NUKC solution opens the ball of radius $< 2c \cdot r_t$ around center u . Let v be the vertex on the u -root path appearing in level L_t . We then pick this node in N . Observe two things: First, this ball covers all the leaves in the sub-tree rooted at v , since $r_t \geq d(u, u')$ for any such u' . Furthermore, since the NUKC solution has only one ball of each radius, we get that $|N \cap L_t| \leq 1$. Finally, since $d(u, w) \geq 2c \cdot r_t$ for all leaves w not in the sub-tree rooted at v , the ball of radius r_t around u does not contain any leaves other than those rooted at v . Contra-positively, since all leaves w are covered in some ball, every leaf must lie in the sub-tree of some vertex picked in N . That is, N is a solution to RMFC-T with value = 1 contradicting the NO case. \square

3 LP-AWARE REDUCTION FROM NUKC TO RMFC-T

For reasons that will be apparent soon, we consider instances \mathcal{I} of NUKC counting multiplicities. That is, we consider an instance to be a collection of tuples $(k_1, r_1), \dots, (k_h, r_h)$ to indicate there are k_i balls of radius r_i . So $\sum_{i=1}^h k_i r_i = k$. Intuitively, the reason we do this is that if two radii r_t and r_{t+1} are “close-by” then it makes sense to round up r_{t+1} to r_t and increase k_t , losing only a constant-factor loss in the dilation.

LP-relaxation for NUKC. We now state the natural LP relaxation for a given NUKC instance \mathcal{I} . For each point $p \in X$ and radius type r_i , we have an indicator variable $x_{p,i} \geq 0$ for whether we place a ball of radius r_i centered at p . By doing a binary search on the optimal dilation and scaling, we may assume that the optimum dilation is 1. Then, the following linear program must

be feasible. We use $B(q, r_i)$ as follows to denote the set of points within distance r_i from q :

$$\begin{aligned} \forall p \in X, \sum_{t=1}^h \sum_{q \in B(p, r_t)} x_{q,t} &\geq 1 \\ \forall t \in 1, \dots, h \sum_{p \in X} x_{p,t} &\leq k_t. \end{aligned} \quad (\text{NUkC LP})$$

LP-relaxation for RMFC-T. Since we reduce fractional NUkC to fractional RMFC-T, we now state the natural LP relaxation for RMFC-T on a tree T of depth $h + 1$. In fact, we will work with the following budgeted-version of RMFC-T (that is equivalent to the original RMFC-T problem – for a proof, see Reference [1]): Instead of minimizing the maximum number of “firefighters” at any level t (that is, $|N \cap L_t|$, where N is the chosen solution), suppose we specify a budget limit of k_t on $|N \cap L_t|$. The goal is to minimize the maximum dilation of these budgets. Then the following is a natural LP relaxation for the budgeted RMFC-T problem on trees. Here $L = L_h$ is the set of leaves, and L_t are the layer t -nodes. For a leaf node v , let P_v denote the vertex set of the unique leaf-root path excluding the root,

$$\begin{aligned} \min \alpha \\ \forall v \in L, \sum_{u \in P_v} y_u &\geq 1 \\ \forall t \in 1, \dots, h \sum_{u \in L_t} y_u &\leq \alpha \cdot k_t. \end{aligned} \quad (\text{RMFC-T LP})$$

The LP-aware Reduction to Tree metrics. We now describe our main reduction algorithm, which takes as input an NUkC instance $\mathcal{I} = \{(X, d); (k_1, r_1), \dots, (k_h, r_h)\}$ and a feasible solution x to (NUkC LP) and returns a budgeted RMFC-T instance \mathcal{I}_T defined by a tree T along with budgets for each level and a feasible solution y to (RMFC-T LP) with dilation 1. The tree we construct will have height $h + 1$ and the budgeted RMFC-T instance will have budgets precisely k_t at level $1 \leq t \leq h$, and the budget for the leaf level is 0. For clarity, throughout this section we use the word *points* to denote elements of the metric space in \mathcal{I} and the word *vertices/nodes* to denote the tree nodes in the RMFC-T instance that we construct.

We build the tree T in a bottom-up manner, where in each round i , we build the $(h + 1 - i)$ th layer of the tree and connect it to the vertices on the $(h + 2 - i)$ th layer. To construct the vertices of the $(h + 1 - i)$ th layer, we pick a set of far-away representative points (the distance scale increases as we move up the tree) and cluster all points to their nearest representative. This is similar to a clustering step in many known algorithms for facility location (see, e.g., Reference [6]), but whereas an arbitrary set of far-away representatives would suffice in the facility location algorithms, we need to be careful in how we choose this set to make the overall algorithm work. Formally, each vertex of the tree T is mapped to some point in X , and we denote the mapping of the vertices at level t by $\psi_t : L_t \rightarrow X$. We will maintain that each ψ_t will be injective, so $\psi_t(u) \neq \psi_t(v)$ for $u \neq v$ in L_t . So, ψ_t^{-1} is well defined for the range of ψ_t . The complete algorithm runs in rounds $h + 1$ to 2 building the tree one level per round. To begin with, the ψ_{h+1} mapping is an arbitrary bijective mapping between $L := L_{h+1}$, the set of leaves of the tree, and the points of X (so, in particular, $|L| = |X|$). We may assume it to be the identity bijection.

In each round t , the range of the mappings become progressively smaller, that is², $\psi_{t-1}(L_{t-1}) \subseteq \psi_t(L_t)$. We call $\psi_{t-1}(L_{t-1})$ as the **winners at level t** . Let $\text{Cov}_t(p) := \sum_{q \in B(p, r_t)} x_{q,t}$ denote the

²We are using the notation $\psi(X) := \bigcup_{x \in X} \psi(x)$.

ALGORITHM 1: Round t of the LP-aware Reduction.**Input:** Level L_t , subtrees below L_t , the mappings $\psi_s : L_s \rightarrow X$ for all $t \leq s \leq h$.**Output:** Level L_{t-1} , the connections between L_{t-1} and L_t , and the mapping ψ_{t-1} .

- 1 Define $A = \psi_t(L_t)$ the set of points who are winners at level t ;
- 2 **while** $A \neq \emptyset$ **do**
- 3 (a) Choose the point $p \in A$ with *minimum coverage* $\text{Cov}_{\geq t}(p)$;
- 4 (b) Let $N(p) := \{q \in A : d(p, q) \leq 2r_{t-1}\}$ be the set of all nearby points in A to p ;
- 5 (c) Create a new tree vertex $w \in L_{t-1}$ corresponding to p and set $\psi_{t-1}(w) := p$. *Call p a winner at level $t - 1$, and each $q \in N(p) \subseteq A$ a loser to p at this level;*
- 6 (d) Create edge (w, v) for tree vertices $v \in \psi_t^{-1}(N(p))$ associated with $N(p)$ at level t ;
- 7 (e) Set $A \leftarrow A \setminus (N(p))$;
- 8 (f) Set $y_w = \text{Cov}_{t-1}(p)$;
- 9 **end**

ALGORITHM 2: Convert NUKC to RMFC-T**Input:** Instance I of NUKC; Feasible solution x to (NUK LP)**Output:** Level $(h + 1)$ -tree instance $I(T)$ of RMFC-T; Feasible solution y to (RMFC-T LP); $\psi_t : L_t \rightarrow X$ injective function for $h + 1 \geq t \geq 1$.

- 1 L_{h+1} is an arbitrary set of $|X|$ vertices. Ψ_{h+1} is an arbitrary bijection from L_{h+1} to X . Set $k_{h+1} = 0$ and $y_w = 0$ for all $w \in L_{h+1}$;
- 2 **for** $t = h + 1$ **to** 2: **do**
- 3 Run Algorithm 1;
- 4 Set the budget of level $t - 1$ for the RMFC-T instance as the k_{t-1} of the NUKC instance
- 5 **end**
- 6 L_0 is a root node node r with edges to every vertex in L_1 . Set $k_r = 0$ and set $y_r = 0$.

fractional amount the point p is covered by radius r_t balls in the solution x , with $\text{Cov}_{h+1}(p) = 0$ for all p . Also define $\text{Cov}_{\geq t}(p) := \sum_{s \geq t} \text{Cov}_s(p)$ denoting the fractional amount p is covered by radius r_t or smaller balls. To decide the vertices that will be winners at level $t - 1$, we perform a clustering step as described earlier: Choose the vertex p with *smallest* $\text{Cov}_{\geq t}$ and remove all other vertices within a distance of $2r_{t-1}$, and repeat. We introduce $w \in L_{t-1}$ and set $\psi_{t-1}(w) = p$, connect w to all nodes v such that $\psi_t(v)$ was removed by p , and set $y_w = \text{Cov}_{t-1}(p)$. The choice of the winner as the vertex with the smallest coverage is crucial in establishing that the RMFC-T LP will admit a feasible solution. The full algorithm is described in Algorithm 1 and Algorithm 2.

We now move to the analysis. In the following proofs, let $W_t \subseteq X$ denote the winners at level t , that is, $W_t = \psi_t(L_t)$. The following claim asserts that the algorithm is well defined.

LEMMA 3.1. *The solution y is a feasible solution to (RMFC-T LP) on I_T with dilation 1.*

PROOF. The proof is via two claims for the two different set of inequalities.

CLAIM 1. *For all $1 \leq t \leq h + 1$, we have $\sum_{w \in L_t} y_w \leq k_t$.*

PROOF. Since we set the budget as 0 for level $h + 1$, and all y_w values are set to 0 as well, the constraint is trivially satisfied for this level. Hence, for the remainder of this proof, fix a level t such that $1 \leq t \leq h$. Recall that $W_t = \psi_t(L_t)$ denotes the winners at level t . By definition of the algorithm, we have that $\sum_{w \in L_t} y_w = \sum_{p \in W_t} \text{Cov}_t(p)$. Now note that for any two points $p, q \in W_t$, we have $B(p, r_t) \cap B(q, r_t) = \emptyset$. To see this, consider the first point that enters A in the $(t + 1)$ th round when L_t was being formed. If this is p , then all points in the radius $2r_t$ ball around p are deleted from A .

Therefore, since the r_t -balls around the winners in W_t are disjoint, the second inequality of NUKC LP implies $\sum_{p \in W_t} \sum_{q \in B(p, r_t)} x_{q,t} \leq k_t$, and the inner summand in the LHS here is precisely $\text{Cov}_t(p)$. \square

CLAIM 2. For any leaf node $w \in L_{h+1}$, we have $\sum_{v \in P_w} y_v \geq 1$.

PROOF. We start with an observation. Fix a level t and a node $u \in L_t$, and suppose $\psi_t(u) = p$. By construction, it follows that p is a winner at every level $t' \geq t$, i.e., there is a leaf v in the sub-tree of T rooted at u , such that $\psi_{t'}(v) = p$ for all $t' \geq t$. We define such a leaf node v to be the *associated leaf* for node u . Since p is a winner at all levels up to t , we also observe that the total y value in the path from v to u is precisely $\text{Cov}_{\geq t}(p)$, i.e., $\sum_{w' \in [u, v]\text{-path}} y_{w'} = \text{Cov}_{\geq t}(p)$.

We now prove the claim using the following simple induction hypothesis that, for every level t , and every node $u \in L_t$, consider any leaf w in the sub-tree rooted at u . Then, we have that $\sum_{w' \in [u, w]\text{-path}} y_{w'} \geq \sum_{w' \in [u, v]\text{-path}} y_{w'}$, where v is the associated leaf for u as defined in the preceding paragraph. In words, it says that the total y -value in the u - w path is at least that in the u - v path for all u , and for all w in u 's sub-tree.

The base case is trivially true for level $h+1$. Now suppose the claim holds for level t , and consider level $t-1$ and node $u \in L_{t-1}$. Suppose u is connected to nodes u_1, u_2, \dots, u_m from level L_t . Let v_1, v_2, \dots, v_m denote the associated leaves of u_1, u_2, \dots, u_m . Note that one of these will also be the associated leaf of node u , let this be v_1 without loss of generality. Finally, let p_1, p_2, \dots, p_m denote the points in X corresponding to u_1, u_2, \dots, u_m , i.e., $\psi_t(u_i) = p_i$ for $1 \leq i \leq m$. Note that $\psi_{t-1}(u) = \psi_t(u_1) = p_1$. Also note that, by construction, we have that $\sum_{w' \in [u_i, v_i]\text{-path}} y_{w'} = \text{Cov}_{\geq t}(p_i)$.

Now consider any w in the sub-tree of T rooted at u , and without loss of generality, suppose w belongs to the sub-tree rooted at u_2 in T . Since the algorithm chooses p_1 as the winner at level $t-1$ to construct node u , we get that $\text{Cov}_{\geq t}(p_1) = \sum_{w' \in [u_1, v_1]\text{-path}} y_{w'} \leq \sum_{w' \in [u_2, v_2]\text{-path}} y_{w'} = \text{Cov}_{\geq t}(p_2)$. Moreover, from the induction hypothesis, we know that $\sum_{w' \in [u_2, w]\text{-path}} y_{w'} \geq \sum_{w' \in [u_2, v_2]\text{-path}} y_{w'}$. By combining these two inequalities, we get that $\sum_{w' \in [u_2, w]\text{-path}} y_{w'} \geq \sum_{w' \in [u_1, v_1]\text{-path}} y_{w'}$. Adding y_u to both sides completes the proof of the induction hypothesis.

The proof of Claim 2 now follows easily by noting that for every node $u \in L_1$ such that $\psi_1(u) = p$ and such that v is the associated leaf of u , we have that $\sum_{w' \in [u, v]\text{-path}} y_{w'} = \text{Cov}_{\geq 1}(p) = 1$. \square

Claim 1 and Claim 2 together complete the proof of Lemma 3.1. \square

The following lemma shows that any good *integral* solution to the RMFC-T instance \mathcal{I}_T can be converted to a good integral solution for the NUKC instance \mathcal{I} .

LEMMA 3.2. Given an NUKC instance \mathcal{I} , let \mathcal{I}_T be the instance created by Algorithm 2. Suppose there exists a feasible solution N to \mathcal{I}_T such that for all $1 \leq t \leq h$, $|N \cap L_t| \leq \alpha k_t$. Then there is a solution to the NUKC instance \mathcal{I} that opens, for each $1 \leq t \leq h$, at most αk_t balls of radius $\leq 2r_{\geq t}$, where $r_{\geq t} := r_t + r_{t+1} + \dots + r_h$.

PROOF. Construct the NUKC solution as follows: for level $1 \leq t \leq h$ and every vertex $w \in N \cap L_t$, place the center at $\psi_t(w)$ of radius $2r_{\geq t}$. We claim that every point in X is covered by some ball. Indeed, for any $p \in X$, look at the leaf $v = \psi_{h+1}^{-1}(p)$, and let $w \in N$ be a node in the root-leaf path. Let $w \in L_t$ for some t . Now observe that $d(p, \psi_t(w)) \leq 2r_{\geq t}$; this is because for any edge (u', v') in the tree where u' is in L_t and is the parent of v' , we have that $d(\psi_{t+1}(v'), \psi_{t+1}(u')) < 2r_t$. \square

This completes the reduction, and we now prove a few results that follow easily from known results about the firefighter problem.

THEOREM 3.3. There is a polynomial time ($O(\log^* n)$, 8)-bi-criteria algorithm for NUKC.

PROOF. Given any instance \mathcal{I} of NUKC, we first club the radii to the nearest power of 2 to get an instance \mathcal{I}' with radii $(k_1, r_1), \dots, (k_h, r_h)$ such that an (a, b) -factor solution for \mathcal{I}' is an $(a, 2b)$ -solution for \mathcal{I} . Now, by scaling, we assume that the optimal dilation for \mathcal{I}' is 1; we let x be the feasible solution to the NUKC LP. Then, using Algorithm 2, we can construct the tree \mathcal{I}'_T and a feasible solution y to the RMFC-T LP. We can now use the following theorem of Chalermsook and Chuzhoy [3]: Given any feasible solution to the RMFC-T LP, we can obtain a feasible set N covering all the leaves such that for all t , $|N \cap L_t| \leq O(\log^* n)k_t$. Finally, we can apply Lemma 3.2 to obtain a $(O(\log^* n), 4)$ solution to \mathcal{I}' (since $r_{\geq t} \leq 2r_t$). \square

PROOFS OF THEOREM 1.4 AND THEOREM 1.5. We use the following claim regarding the integrality gap of RMFC-T LP for depth 2 trees.

CLAIM 3. *When $h = 2$ and k_t 's are integers, given any fractional solution to RMFC-T LP, we can find a feasible integral solution as well.*

PROOF. Given a feasible solution y to (RMFC-T LP), we need to find a set N such that $|N \cap L_t| \leq k_t$ for $t = 1, 2$. There must exist at least one vertex $w \in L_1$ such that $y_w \in (0, 1)$ for otherwise the solution y is trivially integral. If only one vertex $w \in L_1$ is fractional, then since k_1 is an integer, we can raise this y_w to be an integer as well. So at least two vertices w and w' in L_1 are fractional. Now, without loss of generality, let us assume that $|C(w)| \geq |C(w')|$, where $C(w)$ is the set of children of w . Now for some small constant $0 < \epsilon < 1$, we do the following: $y'_w := y_w + \epsilon$, $y'_{w'} := y_{w'} - \epsilon$, $\forall c \in C(w)$, $y'_c := y_c - \epsilon$, and $\forall c \in C(w')$, $y'_c := y_c + \epsilon$. Note that $y(L_1)$ remains unchanged, $y(L_2)$ can only decrease, and root-leaf paths still add to at least 1. We repeat this till we rule out all fractional values. \square

To see the proof of Theorem 1.4, note that an instance of the k -center with outliers problem is an NUKC instance with $(k, 1), (\ell, 0)$, that is, $r_1 = 1$ and $r_2 = 0$. We solve the LP relaxation and obtain the tree and an RMFC-T solution with height $h = 3$. However, since the budget for level 3 and all the y_w values for vertices in that level are 0, we may ignore it. Then we can apply the above claim to obtain a feasible integral solution to RMFC-T on the depth-2 tree. Now we can apply Lemma 3.2 and plug in $r_{\geq 1} = r_1 = 1$ for kCwO, to obtain the 2-factor approximation guarantee.

The proof of Theorem 1.5 is similar. If $r_1 < \theta r_2$, where $\theta = (\sqrt{5} + 1)/2$, then we simply run k -center with $k = k_1 + k_2$. This gives a $2\theta = \sqrt{5} + 1$ -approximation. Otherwise, we apply Lemma 3.2 to get a $2(1 + \frac{1}{\theta}) = \sqrt{5} + 1$ -approximation. \square

We end this section with a general theorem, which is an improvement over Lemma 3.2 in the case when many of the radius types are close to each other, in which case $r_{\geq t}$ could be much larger than r_t . Indeed, the natural way to overcome this would be to *group* the radius types into geometrically increasing values as we did in the proof of Theorem 3.3. This ensures that $r_t \geq 2r_{t+1}$ for all t , and therefore also ensures that $r_{\geq t} \leq O(1)r_t$. While this strategy suffices for Theorem 3.3, we will not be able to use it directly for our bi-criteria result discussed in next section. Indeed, as we will see, we need to perform a different kind of grouping of the radii, to ensure that the *budget for radius type i is 2^i for all $1 \leq i \leq L$* . This is along the lines of the grouping step in Adjashvili et al. [1] for RMFC-T, which ensures that the budget on the firefighters at depth i of the tree is 2^i . If we were to use the old approach, then we will not have both properties holding together. For this reason, we need a generalization of the LP-aware reduction Algorithm 2 that does this geometric grouping inside the algorithm itself, and builds the tree by focusing only on radius types where the radii grow geometrically.

THEOREM 3.4. *Given an NUKC instance $\mathcal{I} = \{M = (X, d), (k_1, r_1), (k_2, r_2), \dots, (k_h, r_h)\}$ and an LP solution x for (NUKC LP), there is an efficient reduction that generates an RMFC-T instance \mathcal{I}_T and an LP solution y to (RMFC-T LP), such that the following holds:*

ALGORITHM 3: Round t of the Improved Reduction.

Input: Level L_t , subtrees below L_t , the mappings $\psi_s : L_s \rightarrow X$ for all $t \leq s \leq h$.
Output: Level L_{t-1} , the connections between L_{t-1} and L_t , and the mapping ψ_{t-1} .

- 1 Let $t' = \min_s$ s.t. $r_s \leq 2r_{t-1}$ be the type of the largest radius smaller than $2r_{t-1}$;
- 2 Define $A = \psi_t(L_t)$ the set of points who are winners at level t ;
- 3 **while** $A \neq \emptyset$ **do**
- 4 (a) Choose the point $p \in A$ with minimum coverage $\text{Cov}_{\geq t}(p)$;
- 5 (b) Let $N(p) := \{q \in A : d(p, q) \leq 2r_{t'}\}$ denote all points in A within $2r_{t'}$ from p ;
- 6 (c) Create new vertices $w_{t-1}, \dots, w_{t'-1} \in L_{t-1}, \dots, L_{t'-1}$ levels respectively, all corresponding to p , i.e., set $\psi_i(w) := p$ for all $t' - 1 \leq i \leq t - 1$. Connect each pair of these vertices in successive levels with edges. Call p a **winner** at levels $t - 1, \dots, t' - 1$;
- 7 (d) Create edge (w_{t-1}, v) for vertices $v \in \psi_{t-1}^{-1}(N(p))$ associated with $N(p)$ at level t ;
- 8 (e) Set $A \leftarrow A \setminus (N(p))$;
- 9 (f) Set $y_{w_i} = \text{Cov}_i(p)$ for all $t - 1 \leq i \leq t' - 1$;
- 10 **end**
- 11 Jump to round $t' - 1$ of the algorithm. Add $t' - 1$ to the set of *barrier levels*;

- (i) For any two tree vertices $w \in L_t$ and $v \in L_{t'}$ where w is an ancestor of v (that is, $t \leq t'$), suppose p and q are the corresponding points in the metric space, i.e., $p = \psi_t(w)$ and $q = \psi_{t'}(v)$, then it holds that $d(p, q) \leq 8r_t$.
- (ii) Suppose there exists a feasible solution N to \mathcal{I}_T such that for all $1 \leq t \leq h$, $|N \cap L_t| \leq \alpha k_t$. Then there is a solution to the NUKC instance \mathcal{I} that opens, for each $1 \leq t \leq h$, at most αk_t balls of radius at most $8r_t$.

3.1 Proof of Theorem 3.4

Both the algorithm and the proof are very similar to the ones we have just seen before. At a high level, the only difference occurs when we identify and propagate winners: Instead of doing it for each radius type, we identify barrier levels where the radius doubles, and perform the clustering step only at the barrier levels. We now present the algorithm, which again proceeds in rounds $h + 1, h, h - 1, \dots, 2$, but makes jumps whenever there are many clusters of similar radius type. To start with, define $r_{h+1} = 0$.

Our proof proceeds almost in an identical manner to those of Lemma 3.1 and Lemma 3.2, but now our tree has an additional property that for any two nodes $u \in L_i$ and $v \in L_{i'}$, where u is an ancestor of v , the distance between the corresponding points in the metric space $p = \psi_i(u)$ and $q = \psi_{i'}(v)$ is at most $d(p, q) \leq 8r_i$, which was the property not true in the earlier reduction. This is easy to see because as we traverse a tree path from u to v , notice that each time we change winners, the distance between the corresponding points in the metric space decreases geometrically. This proves property (i) of Theorem 3.4. The proof of the second property is almost identical to that described in Section 3, and we sketch it below for completeness.

LEMMA 3.5. *The solution y is a feasible solution to (RMFC-T LP) on \mathcal{I}_T with dilation 1.*

PROOF. The proof is via two claims for the two different sets of inequalities. For both the claims, let $W_t \subseteq X$ denote the winners at level t , that is, $W_t = \psi_t(L_t)$.

CLAIM 4. *For all $1 \leq t \leq h$, we have $\sum_{w \in L_t} y_w \leq k_t$.*

PROOF. Fix a barrier level t . By definition of the algorithm, $\sum_{w \in L_t} y_w = \sum_{p \in W_t} \text{Cov}_t(p)$. Now note that for any two points $p, q \in W_t$, we have $B(p, r_t) \cap B(q, r_t) = \emptyset$. To see this, consider the

first point that enters A in the round (corresponding to the previous barrier) when L_t was being formed. If this is p , then all points in the radius $2r_t$ ball is deleted from A . Since the balls are disjoint, the second inequality of NUKC LP implies $\sum_{p \in W_t} \sum_{q \in B_t(p)} x_{q,t} \leq k_t$. The second summand in the LHS is the definition of $\text{Cov}_t(p)$. The same argument holds for all levels t between two consecutive barrier levels t_1 and t_2 s.t. $t_1 > t_2$, as the winner set remains the same, and the radius r_t is only smaller than the radius r_{t_2} at the barrier t_2 . \square

CLAIM 5. For any leaf node $w \in L$, we have $\sum_{v \in P_w} y_v \geq 1$.

PROOF. This proof is identical to that of Claim 2. \square

Finally, the following lemma shows that any good integral solution to the RMFC-T instance \mathcal{I}_T can be converted to a good integral solution for the NUKC instance \mathcal{I} .

LEMMA 3.6. Suppose there exists a feasible solution N to \mathcal{I}_T such that for all $1 \leq t \leq h$, $|N \cap L_t| \leq \alpha k_t$. Then there is a solution to the NUKC instance \mathcal{I} that opens, for each $1 \leq t \leq h$, at most αk_t balls of radius at most $8r_t$.

PROOF. Construct the NUKC solution as follows: for level $1 \leq t \leq h$ and every vertex $w \in N \cap L_t$, place the center at $\psi_t(w)$ of radius $8r_t$. We claim that every point in X is covered by some ball. Indeed, for any $p \in X$, look at the leaf $v = \psi_{h+1}(p)$, and let $w \in N$ be a node in the root-leaf path that covers it in the instance \mathcal{I}_T . By property (i) of Theorem 3.4, we have that the distance between $\psi_t(w)$ and p is at most $8r_t$, and hence the ball of radius $8 \cdot r_t$ around $\psi_t(w)$ covers p . The number of balls of radius type t is trivially at most αk_t . \square

4 GETTING AN $(O(1), O(1))$ -APPROXIMATION ALGORITHM

In this section, we improve our approximation factor on the number of clusters from $O(\log^* n)$ to $O(1)$, while maintaining a constant-approximation in the radius dilation. As mentioned in the Introduction, this requires more ideas, since using (NUKC LP) one cannot get any factor better than $(O(\log^* n), O(1))$ -bicriteria approximation, since any integrality gap for (RMFC-T LP) translates to a $(\Omega(\log^* n), \Omega(1))$ integrality gap for (NUKC LP).

Our algorithm is heavily inspired by the recent paper of Adjiashvili et al. [1], who give an $O(1)$ -approximation for the RMFC-T problem. In fact, the structure of our algorithms follows the same three “steps” of their algorithm. Given an RMFC-T instance, the authors of Reference [1] first “compress” the input tree to get a new tree whose depth is bounded, and, second, [1] give a partial rounding result that saves “bottom heavy” leaves, that is, leaves that in the LP solution are covered by nodes from low levels; and finally, Adjiashvili et al. [1] give a clever partial enumeration algorithm for guessing the nodes from the top levels chosen by the optimum solution. We also proceed in these three steps with the first two being very similar to the first two steps in Reference [1]. However, the enumeration step requires new ideas for our problem. In particular, the enumeration procedure in Reference [1] crucially uses the tree structure of the firefighter instance, and the way our reduction generates the tree for the RMFC-T instance is by using the optimal LP solution for the NUKC instance, which in itself suffers from the $\Omega(\log^* n)$ integrality gap. Therefore, we need to devise a more sophisticated enumeration scheme although the basic ideas are guided by those in Reference [1]. Throughout this section, we do not optimize for the constants.

4.1 Part I: Radii Reduction

In this part, we describe a preprocessing step that decreases the number of types of radii. This is similar to Theorem 5 in Reference [1].

THEOREM 4.1. *Let \mathcal{I} be an instance of NUKC with radii $\{r_1, r_2, \dots, r_k\}$. Then we can efficiently construct a new instance $\widehat{\mathcal{I}}$ with radii multiplicities $(k_0, \widehat{r}_0), \dots, (k_L, \widehat{r}_L)$ and $L = \Theta(\log k)$ such that:*

- (i) $k_i := 2^i$ for all $0 \leq i < L$ and $k_L \leq 2^L$.
- (ii) *If the NUKC instance \mathcal{I} has a feasible solution, then there exists a feasible solution for $\widehat{\mathcal{I}}$.*
- (iii) *Given an (α, β) -bicriteria solution to $\widehat{\mathcal{I}}$, we can efficiently obtain a $(3\alpha, \beta)$ -bicriteria solution to \mathcal{I} .*

PROOF. For an instance \mathcal{I} , we construct the compressed instance $\widehat{\mathcal{I}}$ as follows. Partition the radii into $\Theta(\log k)$ classes by defining barriers at $\widehat{r}_i = r_{2^i}$ for $0 \leq i \leq \lfloor \log k \rfloor$. Now to create instance $\widehat{\mathcal{I}}$, we simply round up all the radii r_j for $2^i \leq j < 2^{i+1}$ to the value $\widehat{r}_i = r_{2^i}$. Notice that the multiplicity of \widehat{r}_i is precisely 2^i (except maybe for the last bucket, where there might be fewer radii rounded up than the budget allowed).

Property (i) is just by construction of instance. Property (ii) follows from the way we rounded up the radii. Indeed, if the optimal solution for \mathcal{I} opens a ball of radius r_j around a point p , then we can open a cluster of radius \widehat{r}_i around p , where i is such that $2^i \leq j < 2^{i+1}$. Clearly, the number of clusters of radius \widehat{r}_i is at most 2^i , because OPT uses at most one cluster of each radius r_j .

For property (iii), suppose we have a solution \widehat{S} for $\widehat{\mathcal{I}}$ that opens $\alpha 2^i$ clusters of radius $\beta \widehat{r}_i$ for all $0 \leq i \leq L$. Construct a solution S for \mathcal{I} as follows. For each $1 \leq i \leq L$, let C_i denote the set of centers where \widehat{S} opens balls of radius $\beta \widehat{r}_i$. In the solution S , we also open balls at precisely these centers with 2α balls of radius βr_j for every $2^{i-1} \leq j < 2^i$. Since $|C_i| \leq \alpha \cdot 2^i$, we can open a ball at every point in C_i ; furthermore, since $j < 2^i$, we have $r_j \geq \widehat{r}_i$ and so we cover whatever the balls from \widehat{S} covered.

Finally, we also open the α clusters (corresponding to $i = 0$) of radius $\beta r_1 = \beta \widehat{r}_0$ at the respective centers C_0 , where \widehat{S} opens centers of radius \widehat{r}_0 . Therefore, the total number of clusters of radius type is at most 2α with the exception of r_1 , which may have 3α clusters. \square

4.2 Part II: Satisfying Bottom Heavy Points

One main reason why the above height reduction step is useful, is the following theorem from [1] for RMFC-T instances on trees; we provide a proof sketch for completeness.

THEOREM 4.2 ([1]). *Given a tree T of height h and a feasible solution y to RMFC-T LP with objective value at most 1, we can find a feasible integral solution N to RMFC-T such that for all $1 \leq t \leq h$, $|N \cap L_t| \leq k_t + h$.*

PROOF. Let y be a basic feasible solution of (RMFC-T LP). Call a vertex v of the tree *loose* if $y_v > 0$ and the sum of y -mass on the vertices from v to the root (inclusive of v) is < 1 . Let V_L be the set of loose vertices of the tree, and let V_I be the set of vertices with $y_v = 1$. Clearly, $N = V_L \cup V_I$ is a feasible solution: Every leaf-to-root path either contains an integral vertex or at least two fractional vertices with the vertex closer to root being loose. Next we claim that $|V_L| \leq h$; this proves the theorem, since $|N \cap L_t| \leq |V_I \cap L_t| + |V_L| \leq k_t + |V_L|$.

The full proof can be found in Lemma 6 in Reference [1]; here is a high level sketch. There are $|L| + h$ inequalities in (RMFC-T LP), and so the number of fractional variables is at most $|L| + h$. We may assume there are no $y_v = 1$ vertices. Now, in every leaf-to-root path there must be at least two fractional vertices, and the one closest to the leaf must be non-loose. If the closest fractional vertex to each leaf was unique, then that would account for $|L|$ fractional non-loose vertices implying the number of loose vertices must be $\leq h$. This may not be true; however, if we look at *linearly independent* set of inequalities that are tight, then we can argue uniqueness as a clash can be used to exhibit linear dependence between the tight constraints. \square

ALGORITHM 4: Partial Round

-
- Input:** NUKC instance \widehat{I} with radii multiplicities $(k_0, \widehat{r}_0), (k_1, \widehat{r}_1), \dots, (k_L, \widehat{r}_L)$ with budgets $k_i = 2^i$ for radius type \widehat{r}_i ; feasible LP solution x to (NUKC LP) for \widehat{I} .
- Parameter:** $\tau = \log^{(q)} L + 1$, where $\log^{(q)} L$ denotes the iterated log applied q times to L
- Output:** Solution S opening $O(q) \cdot k_t$ balls of radius $O(\widehat{r}_t)$ for $\tau \leq t \leq L$; Covers all of $X' := \{p \in X : \text{Cov}_{\geq \tau}(p) \geq \frac{1}{2}\}$
- 1 Use the LP-reduction algorithms Algorithm 3 and Algorithm 2 (with Algorithm 3 called in Line 3) to obtain a tree T of height $L + 1$ and fractional solution y for (RMFC-T LP) on T ;
 - 2 **for** $q \geq \ell \geq 1$ **do**
 - 3 Let T_ℓ denote the collection of sub-trees consisting of the portion of the tree T in levels $\{\log^{(\ell)} L + 1, \dots, \log^{(\ell-1)} L\}$;
 - 4 $X_\ell \subseteq X'$ denote the set of clients for which $\sum_{\log^{(\ell)} L + 1 \leq t \leq \log^{(\ell-1)} L} \text{Cov}_t(p) \geq \frac{1}{2q}$;
 - 5 Introduce a fake root r_ℓ and attach each sub-tree in T_ℓ to obtain a single tree \widehat{T}_ℓ ;
 // Observe that the solution $2q \cdot y$, when restricted to \widehat{T}_ℓ , is feasible for saving all the leaves corresponding to points in X_ℓ .
 - 6 Use procedure described in Theorem 4.2 on \widehat{T}_ℓ to obtain integral solution N_ℓ for RMFC instance.;
 // Note: for layer i , with $i \in \{\log^{(\ell)} L + 1, \dots, \log^{(\ell-1)} L\}$, one has $|N_\ell \cap L_i| \leq 2q \cdot (k_i + \log^{(\ell-1)} L) \leq 4q \cdot k_i$, since $k_i = 2^i > \log^{(\ell-1)} L$.
 - 7 **end**
 - 8 Let $N := \cup_{1 \leq \ell \leq q} N_\ell$. Use Theorem 3.4, part (ii) to obtain solution for the NUKC instance \widehat{I} that covers all points in X' ;
 // Note $X' = \cup_{1 \leq \ell \leq q} X_\ell$ and thus is covered. Note the number of balls opened of type t for $\tau \leq t \leq L$ is at most $4q \cdot k_t$, and their radius is $\leq 8\widehat{r}_t$.
-

Using the above, we can now give a rounding algorithm that covers all points in \widehat{I} with large coverage. This subroutine will be used in the final algorithm described in the next section. The corollary encapsulates the main property of the algorithm.

COROLLARY 4.3. *Suppose we are given an NUKC instance \widehat{I} with radii multiplicities $(k_0, \widehat{r}_0), (k_1, \widehat{r}_1), \dots, (k_L, \widehat{r}_L)$ with budgets $k_i = 2^i$ for radius type \widehat{r}_i , and an LP solution x to (NUKC LP) for \widehat{I} . Let $\tau = \log^{(q)} L + 1$, where $\log^{(q)} L$ denotes the iterated log applied q times to L , and suppose $X' \subseteq X$ be the points such that $\text{Cov}_{\geq \tau}(p) \geq \frac{1}{2}$. Then Algorithm 4 returns a solution that opens at most $4q \cdot k_t$ balls of radius $8\widehat{r}_t$ for $\tau \leq t \leq L$, and covers all of X' .*

The above immediately implies a (very weakly) quasi-polynomial time $O(1)$ -approximation for NUKC. First we apply Theorem 4.1 to move to an instance \widehat{I} . Next, we enumerate the set of clusters of radii \widehat{r}_t for $0 \leq t \leq \log^{(q)} L$ for some constant q . Then we explicitly solve an LP where all the uncovered points need to be fractionally covered by only clusters of radius type \widehat{r}_t for $t > \log^{(q)} L$. This forces the set X' defined in Algorithm 4 to be the same as the set of uncovered points, and therefore the solution it returns (along with the enumerated centers) forms a feasible solution that gives a $(12q, 8)$ -factor bicriteria approximation (the 4 becomes a 12 due to Theorem 4.1). The time complexity is dominated by the enumeration of the optimal clusters of radii \widehat{r}_t for $0 \leq t \leq \log^{(q)} L$. This time is $n^{O(\log^{(q-1)} L)} = n^{O(\log^{(q)} k)}$, since the number of clusters of radius at least $\widehat{r}_{\log^{(q)} L}$ is at most $O(2^{\log^{(q)} L}) = O(\log^{(q-1)} L)$. As a result, we get the following corollary. Note that this gives an alternate way to prove Theorem 3.3. All this, in the RMFC-T context, was also observed by [1].

COROLLARY 4.4. *For any $q \geq 1$, there exists an $(12q, 8)$ -factor bicriteria algorithm for NUKC, which runs in $n^{O(\log^{(q)} k)}$ time.*

4.3 Part III: Clever Enumeration of Large Radii Clusters

In this section, we show how to obtain a $(24, 10)$ -factor bi-criteria algorithm. Given an NUKC instance \mathcal{I} , we first use Theorem 4.1 to get instance $\widehat{\mathcal{I}}$ satisfying the conditions stated in the theorem. Recall, an (α, β) -factor bicriteria algorithm for $\widehat{\mathcal{I}}$ implies an $(3\alpha, \beta)$ -factor bicriteria algorithm for \mathcal{I} . Henceforth, we focus our attention on $\widehat{\mathcal{I}}$. We will give an $(8, 10)$ -factor bicriteria algorithm for this (see Lemma 4.5).

At a high level, our algorithm tries to “guess” the centers³ A of large radius, that is, \widehat{r}_i for $i < \tau := \log \log L + 1 = \log \log \log k + 1$, which the optimum solution uses. However, this guessing is done in a cleverer way than in Corollary 4.4. In particular, given a guess that is consistent with the optimum solution (the initial “null set” guess is trivially consistent), our enumeration procedure generates a list of candidate additions to A of size *at most* $2^\tau \approx \text{poly log log } k$ (instead of n), one of which is a consistent enhancement of the guessed set A . To make this procedure work out, we also need to maintain a guess D of points where the optimum solution *does not* open centers. Given any state (A, D) consistent with the optimal solution, our algorithm ignores points already covered by balls around centers in A , and *tries to cover all the uncovered points using the small-radius balls of radius \widehat{r}_t for $t \geq \tau$* . If it succeeds and finds a solution B , then we can output our final solution to be $A \cup B$. However, if it fails, then we identify a way to make progress by enumerating a bounded number of extensions to (A, D) , one of which is (a) consistent with the optimal solution, and (b) a non-trivial improvement over the current state (A, D) . We now provide more details.

We start with some definitions. Throughout, A and D represent sets of tuples of the form (p, t) where $p \in X$ and $t \in \{0, 1, \dots, \tau\}$. Given such a set A , we associate a partial solution S_A that, for every $(p, t) \in A$, opens a ball of radius $10\widehat{r}_t$ centered at the point p . For the sake of analysis, fix an optimum solution OPT. We say the set A is **consistent** with OPT if for all $(p, t) \in A$, there exists a *distinct* $q \in X$ such that OPT opens a ball of radius \widehat{r}_t at q and $d(p, q) \leq 5\widehat{r}_t$. In particular, this implies that S_A covers all points that this OPT-ball covers. We say the set D is **consistent** with OPT if for all $(q, t) \in D$, OPT *does not* open a radius \widehat{r}_t ball at q (it may open a different radius ball at q though). Given a set D , we define the minLevel of each point p as follows:

$$\text{minLevel}_D(p) := 1 + \max\{t : (q, i) \in D \text{ for all } q \in B(p, \widehat{r}_i), \text{ and all } i \leq t\}.$$

In words, it says the following: If D is consistent with OPT and if $\text{minLevel}_D(p) = t$, then in the OPT solution, p must be covered by a ball of radius \widehat{r}_t or smaller.

Given the definitions of A and D , we now describe a more nuanced LP-relaxation for NUKC, which tries to find a solution that is consistent with A and D . To this end, let X_G be the subset of points in X covered by the partial solution S_A , and fix any subset $Y \subseteq X \setminus X_G$ of points. Define the following LP:

$$\begin{aligned} \forall p \in Y, \quad & \sum_{t=\text{minLevel}_D(p)}^L \sum_{q \in B(p, \widehat{r}_t)} x_{q,t} \geq 1 && (\text{LP}_{\text{NUKC}}(Y, A, D)) \\ \forall t \in 1, \dots, h, \quad & \sum_{q \in X} x_{q,t} \leq k_t \\ \forall (p, t) \in A, \quad & x_{p,t} = 1 \\ \forall (p, t) \in D, \quad & x_{p,t} = 0. \end{aligned}$$

³Actually, we end up guessing centers “close” to the optimum centers, but for this introductory paragraph this intuition is adequate.

The following claim encapsulates the utility of the above relaxation.

CLAIM 6. *If (A, D) is consistent with OPT, then $(\text{LP}_{\text{NUKC}}(Y, A, D))$ is feasible.*

PROOF. We describe a feasible solution to the above LP using OPT. Given OPT, define O to be the collection of tuples (q, t) where OPT opens a radius \widehat{r}_t ball at point q . Note that the number of tuples in O with second term t is at most k_t . Since A is consistent with OPT, for every $(p, t) \in A$, there exists a distinct $(q, t) \in O$ such that $d(p, q) \leq 5\widehat{r}_t$. Remove all such tuples from O to get the set O' , and define $x'_{q,t} = 1$ for all tuples in O' . We claim that x' forms a feasible solution to $(\text{LP}_{\text{NUKC}}(Y, A, D))$.

Indeed, since OPT satisfies the second inequality and we only consider a subset of it corresponding to O' , the second inequality is trivially satisfied by x' . We now show the first inequality holds for every point in $X \setminus X_G$. Let p be any such point. Let $(q, t) \in O$ be the tuple that covers p in OPT. That is, in OPT there is a ball of radius \widehat{r}_t around q that covers p . First note that since D is consistent with OPT, $t \geq \text{minLevel}_D(p)$. Thus taking the sum from $t = \text{minLevel}_D(p)$ is alright. Next, we show that $(q, t) \in O'$. This will prove the claim. Suppose not. This means there is some $(s, t) \in A$ such that $d(s, q) \leq 5\widehat{r}_t$. By triangle inequality, this would imply $d(p, s) \leq 10\widehat{r}_t$. Which in turn would imply $p \in X_G$, which is a contradiction. \square

Finally, for convenience, we define a **forbidden set** $F := \{(p, i) : p \in X, 1 \leq i < \tau\}$ that if added to D disallows any large radii balls to be placed anywhere.

Now we are ready to describe the enumeration Algorithm 5, and give a sketch of the analysis. We start with A and D being null, and thus are vacuously consistent with OPT. The procedure first solves $(\text{LP}_{\text{NUKC}}(Y, A, D))$ and uses Algorithm 4 to cover the “bottom-heavy” points (Step 8–10). If the remaining top-heavy points X_T can also be covered fractionally by small-radii balls, then we use Algorithm 4 again to get an $(O(1), O(1))$ -bicriteria algorithm (Steps 11–15). Else, the top-heavy points X_T can't be covered fractionally. The enumeration step then enhances the (A, D) tuple. This is the essence of Steps 17–24. To do so, first we note that there must exist some point $q \in X_T$ that is not covered by a “small” radius ball in OPT (otherwise the LP would've fractionally covered X_T). Therefore, this point is covered by a large radius ball in OPT. The set P_t in step 19 is sort of a “net” in that this point q must be close-by to some point $p \in P_t$. The “correct” branch in the Steps 17–24 is the one that picks this p (we do not know which one it is, so we recurse over all). Given this p , there are two cases. Either p is close to an OPT center with a certain radius \widehat{r}_t , in which case (p, t) is added to A . Or, p is far from all such centers, in which case we can add a significant number of tuples to D . At this point, a potential function argument shows that $O(2^\tau) = O(\text{poly log log } k)$ depth of recursion suffices. To show why the branching factor is under control (or why the running time is bounded), one shows that the size of the net P_t is small; indeed, $|P_t|$ is proven to be $\leq O(\text{log log } k)$. Therefore, the total running time of the enumeration procedure is $(\text{log log } k)^{(\text{poly log log } k)} = o(k)$.

Define $\gamma_0 := 8 \text{ log log } k \cdot \text{log log log } k$. This parameter sets an upper bound on the depth of the recursion. The algorithm is run with $\text{Enum}(\emptyset, \emptyset, \gamma_0)$. The proof that we get a polynomial time $(O(1), O(1))$ -bicriteria approximation algorithm follows from three lemmas. Lemma 4.5 shows that if Step 11 is true with a consistent pair (A, D) , then the output in Step 14 is a $(O(1), O(1))$ -approximation. Lemma 4.6 shows that indeed Step 11 is true for γ_0 as set. Finally, Lemma 4.7 shows with such a γ_0 , the algorithm runs in polynomial time.

LEMMA 4.5. *If (A, D) is a consistent pair such that Step 11 is true, then the solution returned is an $(8, 10)$ -factor bicriteria approximation algorithm for $\widehat{\mathcal{I}}$.*

PROOF. Since A is consistent with OPT, S_A opens at most k_t centers with radius $\leq 10\widehat{r}_t$ for all $0 \leq t < \tau$. By design, S_B and S_T open at most $8k_t$ centers with radius $8r_t$ for $\tau \leq t \leq L$. \square

ALGORITHM 5: Enum(A, D, γ)

```

1 Let  $X_G = \{p : \exists (q, t) \in A \text{ s.t. } d(p, q) \leq 10\widehat{r}_t\}$  denote points covered by  $S_A$ ;
2 if there is no feasible solution to  $LP_{\text{NUKC}}(X \setminus X_G, A, D)$  then
3   | Abort.// Claim 6 implies  $(A, D)$  is not consistent.
4 end
5 else
6   |  $x^*$  be a feasible solution to  $LP_{\text{NUKC}}(X \setminus X_G, A, D)$ ;
7 end
8 Let  $X_B = \{u \in X \setminus X_G : \text{Cov}_{\geq \tau}(u) \geq \frac{1}{2}\}$  denote bottom-heavy points in  $x^*$ ;
9 Let  $S_B$  be the solution returned by Algorithm 4;
   // By definition of  $\tau$ ,  $q = 2$ . Thus, this solution opens  $8k_t$  balls of radius  $8\widehat{r}_t$  for
    $\tau \leq t \leq L$  and covers all of  $X_B$ .
10 Let  $X_T = X \setminus (X_G \cup X_B)$  denote the top heavy points in  $x^*$ ;
11 if  $LP_{\text{NUKC}}(X_T, A, F \cup D)$  has a feasible solution  $x_T$  then
12   | By definition of  $F$ , in  $x_T$  we have  $\text{Cov}_{\geq \tau}(u) = 1$  for all  $u \in X_T$ ;
13   | Let  $S_T$  be the solution returned by Algorithm 4;
   // This solution opens  $8k_t$  balls of radius  $8\widehat{r}_t$  for  $\tau \leq t \leq L$  and covers all of  $X_T$ .
14   | Output  $(S_A \cup S_B \cup S_T)$ ;
   // This is a  $(8, 10)$ -approximation for the NUKC instance  $\widehat{I}$ ; see Theorem 4.5
15 end
16 else
17   | for every level  $1 \leq t < \tau$  do
18     | Let  $C_t = \{p \in X_T \text{ s.t. } \text{minLevel}_D(p) = t\}$ , the set of points in  $X_T$  with minLevel  $t$ ;
19     | Let  $P_t$  denote any maximal subset of  $C_t$  such that any two points in  $P_t$  are at least  $2\widehat{r}_t$  apart. ;
20     | for every node  $p$  in  $P_t$  do
21       | Enum( $A \cup \{(p, t)\}, D, \gamma - 1$ );
22       | Enum( $A, D \cup \bigcup_{p' \in B(p, 5\widehat{r}_t)} \{(p', t)\}, \gamma - 1$ );
23     | end
24   | end
25 end

```

LEMMA 4.6. Enum($\emptyset, \emptyset, \gamma_0$) finds consistent (A, D) such that Step 11 is true.

PROOF. For this we identify a particular execution path of the procedure Enum(A, D, γ), that at every point maintains a tuple (A, D) that is consistent with OPT. At the beginning of the algorithm, $A = \emptyset$ and $D = \emptyset$, which is consistent with OPT.

Now consider a tuple (A, D) that is consistent with OPT and let us assume that we are within the execution path Enum(A, D, γ). Let $X \setminus X_G$ be the points not covered by A and let x^* be a solution to $LP_{\text{NUKC}}(X \setminus X_G, A, D)$. If OPT covers all top-heavy points X_T using only smaller radii, then this implies $LP_{\text{NUKC}}(X_T, A, F \cup D)$ has a feasible solution implying that Step 11 is true. So, we may assume, there exists at least one top-heavy point $q \in X_T$ that OPT covers using a ball radii $\geq \widehat{r}_\tau$ around a center o_q . In particular, $\text{minLevel}_D(q) < \tau$. Let $q \in C_t$ for some $0 \leq t < \tau$. Then, there exists $p \in P_t$ such that $d(p, q) \leq 2\widehat{r}_t$ (due to the maximality of P_t in line 19 of the algorithm). Note that p could be the point q itself. We now show that there is at least one recursive call where we make non-trivial progress in (A, D) . Indeed, we do this in two cases:

Case (A): OPT opens a ball of radius \widehat{r}_t at a point o such that $d(o, p) \leq 5\widehat{r}_t$. In this case, Step 21 maintains consistency. Furthermore, we can “charge” (p, t) distinctly to the tuple (o, t) .

To see this, for contradiction, let us assume that before arriving to the recursive call where (p, t) is added to A , some other tuple $(u, t) \in A'$, in an earlier recursive call with (A', D') as parameters charged to (o, t) . Then by definition we know that $d(u, o) \leq 5\widehat{r}_t$ implying $d(u, p) \leq 10\widehat{r}_t$. Then p would be in X_G in all subsequent calls, contradicting that $p \in X_T$ currently.

Case (B): There is no $(o, t) \in \text{OPT}$ with $d(o, p) \leq 5\widehat{r}_t$. In this case, for all points $p' \in B(p, 5\widehat{r}_t)$ we add (p', t) to D and the recursive call in Step 22.

To summarize, one of the recursive calls is guaranteed to be consistent. Next, we bound the depth of recursion. In case (A), the measure of progress is clear: We increase the size of $|A|$, and it can be argued (we do so below) that the maximum size of A is at most poly $\log \log k$. Case (B) is subtler: We definitely increase the size of D , but D could grow as large as $\Theta(n)$. Before going to the formal proof, let us intuitively argue what “we learn” in Case (B). Recall q is covered in OPT by a ball around the center o_q . Next, since $\text{minLevel}(q) = t < \tau$, we can infer two properties: (i) By definition there is a point $v \in B(q, \widehat{r}_t)$ such that $(v, t) \notin D$, and (ii) $d(q, o_q) \leq \widehat{r}_t$ since OPT must cover q with a smaller radius than \widehat{r}_t due to (A, D) being consistent. Together, we get $d(v, o_q) \leq 2\widehat{r}_t$, that is, $v \in B(o_q, 2\widehat{r}_t)$. Now also note that since $d(p, q) \leq 2\widehat{r}_t$ by the definition of p from the paragraph above in the proof. Therefore, $d(p, o_q) \leq 3\widehat{r}_t$. Now, since in case (B), we place (u, t) in the set D for all points $u \in B(p, 5\widehat{r}_t)$, we can conclude that we have placed $(u, t) \in D$ for all points $u \in B(o_q, 2\widehat{r}_t)$ as well. This is “new information,” since for the current D we know that for at least one point $v \in B(o_q, 2\widehat{r}_t)$, we had $(v, t) \notin D$.

Formally, we define the following potential function. Let O_τ denote the centers in OPT around which balls of radius \widehat{r}_j , $j < \tau$ have been opened. Given the set D , for $0 \leq t < \tau$ and for all $o \in O_\tau$, define the indicator variable $Z_{o,t}^{(D)}$, which is 1 if for all points $u \in B(o, 2\widehat{r}_t)$, we have $(u, t) \in D$ and 0 otherwise.

$$\Phi(A, D) := |A| + \sum_{o \in O_\tau} \sum_{t=0}^{\tau} Z_{o,t}^{(D)}.$$

Note that $\Phi(\emptyset, \emptyset) = 0$. From the previous paragraph, we conclude that in both case (A) or case (B), the potential increases by *at least* 1. Finally, for any consistent A, D we can upper bound $\Phi(A, D)$ as follows. Since A is consistent, $|A| \leq \sum_{t=0}^{\tau} 2^t \leq 2^{\tau+1} = 2 \log L = 2 \log \log k$. The second term in Φ is at most $2^{\tau+1} \cdot \tau = 2 \log L (\log \log L + 1)$. Thus, in at most $4 \log \log k \cdot (\log \log \log k + 1) < \gamma_0$ steps we reach a consistent pair (A, D) with Step 11 true. \square

LEMMA 4.7. *Enum($\emptyset, \emptyset, \gamma_0$) runs in polynomial time for large enough k .*

PROOF. Each single call of Enum is clearly polynomial time, and so we bound the number of recursive calls. To this end, we first bound the number of recursive calls in a single execution of Enum(A, D, γ). For a fixed tuple (A, D) , Algorithm 5 invokes two recursive calls for each $p \in P_t$ for every level t such that $0 \leq t < \tau$. In what follows, we bound $|P_t|$ to be at most $O(\log \log k)$ for every $0 \leq t < \tau$, and hence the overall number of recursive calls in one execution of Algorithm 5 would be $O(\tau \cdot \log \log k) = O(\log \log k \cdot \log \log \log k)$. Finally, since the depth of the recursion is at most $\gamma_0 = 8 \log \log k \cdot \log \log \log k$, we get that the overall number of recursive calls to the algorithm is at most $2^{\text{poly} \log \log k} = o(k)$.

To complete the proof, we bound $|P_t|$ to be at most $O(\log \log k)$ for every $0 \leq t < \tau$. Indeed, notice that by definition of P_t in Algorithm 5 in line 19, the \widehat{r}_t balls around every point $p \in P_t$ are disjoint. Moreover, since $P_t \subseteq C_t$ and C_t are all points $p \in X_T$ with $\text{minLevel}_D(p) = t$, we can infer two things: First, by virtue of p being in X_T , we get that $\text{Cov}_{\geq t}(p) < \frac{1}{2}$, and, second, since $\text{minLevel}_D(p) = t$, p receives no coverage from balls of radius larger than \widehat{r}_t . Hence, we can conclude that $\sum_{t'=t}^{\tau-1} \text{Cov}_{t'}(p) \geq \frac{1}{2}$.

To summarize, we have that $\sum_{t'=t}^{\tau-1} \sum_{q \in B(p, \widehat{r}_{t'})} x_{q,t'}^* \geq \frac{1}{2}$ for all $p \in P_t$, and, moreover, the balls of radius \widehat{r}_t around points in P_t are disjoint. These two suffice to establish a bound on $|P_t|$. Indeed, note that $\frac{1}{2}|P_t| \leq \sum_{p \in P_t} \sum_{t'=t}^{\tau-1} \sum_{q \in B(p, \widehat{r}_{t'})} x_{q,t'}^* \leq \sum_{t'=t}^{\tau-1} \sum_{q \in X} x_{q,t'}^* \leq \sum_{t'=t}^{\tau-1} k_{t'} \leq 2 \cdot 2^{\tau-1} = O(\log \log k)$. Here, the first inequality is from the argument in the preceding paragraph, the second inequality is from the disjointness of the balls of radius \widehat{r}_t around points in P_t , the third inequality follows from the budgets at different levels, and the fourth inequality is a simple geometric sum. \square

5 CONCLUSION

In this article we initiate the study of the NUKC problem that generalizes the classic k -center problem and the k -center with outlier problem. We show that no non-trivial unicriterion approximation is possible for NUKC, and complement it with an $(O(1), O(1))$ -bicriteria result. We also give a 2-approximation for the k -center with outlier problem, and also a $(1 + \sqrt{5})$ -approximation when there are only two types of radii. At a conceptual level, we show a close connection between NUKC and the RMFC-T problem.

The main question left open from this work is to obtain an unicriterion approximation algorithm for NUKC when there are only constantly many types of radii. Our hardness reduction fails, since the RMFC-T problem is indeed polynomial time solvable when the height of the tree, which corresponds to the number of distinct radii, is a constant. We believe there should be an $O(1)$ -approximation for NUKC when the number of types of radii is a constant.

ACKNOWLEDGMENT

We thank the anonymous referees whose detailed comments have significantly improved the exposition of our article.

REFERENCES

- [1] David Adjiashvili, Andrea Baggio, and Rico Zenklusen. 2018. Firefighting on trees beyond integrality gaps. *ACM Trans. Algor.* 15, 2 (2018), 20.
- [2] J. Byrka, T. Pensch, B. Rybicki, A. Srinivasan, and K. Trinh. 2015. An improved approximation for k -median, and positive correlation in budgeted optimization. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA'15)*. 737–756.
- [3] P. Chalermsook and J. Chuzhoy. 2010. Resource minimization for fire containment. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA'10)*. 1334–1349.
- [4] M. Charikar, L. O' Callaghan, and R. Panigrahy. 2003. Better streaming algorithms for clustering problems. In *Proceedings of the ACM Symposium on Theory of Computing (STOC'03)*. 30–39.
- [5] Moses Charikar, Chandra Chekuri, Tomás Feder, and Rajeev Motwani. 2004. Incremental clustering and dynamic information retrieval. *SIAM J. Comput.* 33, 6 (2004), 1417–1440.
- [6] Moses Charikar, Sudipto Guha, Éva Tardos, and David B. Shmoys. 2002. A constant-factor approximation algorithm for the k -median problem. *J. Comput. System Sci.* 65, 1 (2002), 129–149.
- [7] M. Charikar, S. Khuller, D. M. Mount, and G. Narasimhan. 2001. Algorithms for facility location problems with outliers. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA'01)*. 642–651.
- [8] S. Finbow, A. King, G. MacGillivray, and R. Rizzi. 2007. The firefighter problem for graphs of maximum degree three. *Discr. Math.* 307, 16 (2007), 2094–2105.
- [9] I. L. Goertz and V. Nagarajan. 2011. Locating depots for capacitated vehicle routing. In *Proceedings of the International Workshop on Approximation Algorithms for Combinatorial Optimization Problems*. 230–241.
- [10] T. F. Gonzalez. 1985. Clustering to minimize the maximum intercluster distance. *Theor. Comput. Sci.* 38 (1985), 293–306. DOI: [https://doi.org/10.1016/0304-3975\(85\)90224-5](https://doi.org/10.1016/0304-3975(85)90224-5)
- [11] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. 1998. CURE: An efficient clustering algorithm for large databases. In *Proceedings of the International Conference on Management of Data (SIGMOD'98)*, Vol. 27. ACM, 73–84.
- [12] S. Har-Peled and S. Mazumdar. 2004. Coresets for k -means and k -median clustering and their applications. In *Proceedings of the ACM Symposium on Theory of Computing (STOC'04)*. 291–300.
- [13] D. S. Hochbaum and D. B. Shmoys. 1985. A best possible heuristic for the k -center problem. *Math. Operat. Res.* 10, 2 (1985), 180–184.

- [14] S. Im and B. Moseley. 2015. Brief announcement: Fast and better distributed MapReduce algorithms for k -center clustering. In *Proceedings of the ACM Symposium on Parallelism in Algorithms and Architectures*. 65–67.
- [15] K. Jain and V. V. Vazirani. 2001. Approximation algorithms for metric facility location and k -median problems using the primal-dual schema and lagrangian relaxation. *J. ACM* 48, 2 (2001), 274–296.
- [16] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. 2004. A local search approximation algorithm for k -means clustering. *Comput. Geom.* 28, 2–3 (2004), 89–112.
- [17] A. King and G. MacGillivray. 2010. The firefighter problem for cubic graphs. *Discr. Math.* 310, 3 (2010), 614–621.
- [18] A. Kumar, Y. Sabharwal, and S. Sen. 2004. A simple linear time $(1 + \epsilon)$ -approximation algorithm for k -means clustering in any dimensions.. In *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS'04)*. 454–462.
- [19] G. Laporte. 1998. Location routing problems. In *Vehicle Routing: Methods and Studies*, B. L. Golden and A. A. Assad (Eds.). 163–198.
- [20] Shi Li and Ola Svensson. 2016. Approximating k -median via pseudo-approximation. *SIAM J. Comput.* 45, 2 (2016), 530–547.
- [21] G. Malkomes, M. J. Kusner, W. Chen, K. Q. Weinberger, and B. Moseley. 2015. Fast distributed k -center clustering with outliers on massive data. In *Proceedings of the Conference on Advances in Neural Information Processing Systems (NeurIPS'15)*. 1063–1071.
- [22] R. McCutchen and S. Khuller. 2008. Streaming algorithms for k -center clustering with outliers and with anonymity. In *Proceedings of the International Workshop on Approximation Algorithms for Combinatorial Optimization Problems*. 165–178.
- [23] H. Mina, V. Jayaraman, and R. Srivastava. 1998. Combined location-routing problems: A synthesis and future research directions. *Eur. J. Operat. Res.* 1 (1998), 1–15.

Received February 2018; revised December 2019; accepted April 2020