

Non-Uniform k -Center and Greedy Clustering

Tanmay Inamdar ✉ 

Department of Informatics, University of Bergen, Norway

Kasturi Varadarajan ✉

Department of Computer Science, University of Iowa, Iowa City, IA, USA

Abstract

In the Non-Uniform k -Center (NU k C) problem, a generalization of the famous k -center clustering problem, we want to cover the given set of points in a metric space by finding a placement of balls with specified radii. In t -NU k C, we assume that the number of distinct radii is equal to t , and we are allowed to use k_i balls of radius r_i , for $1 \leq i \leq t$. This problem was introduced by Chakrabarty et al. [ACM Trans. Alg. 16(4):46:1-46:19], who showed that a constant approximation for t -NU k C is not possible if t is unbounded, assuming $P \neq NP$. On the other hand, they gave a bicriteria approximation that violates the number of allowed balls as well as the given radii by a constant factor. They also conjectured that a constant approximation for t -NU k C should be possible if t is a fixed constant. Since then, there has been steady progress towards resolving this conjecture – currently, a constant approximation for 3-NU k C is known via the results of Chakrabarty and Negahbani [IPCO 2021], and Jia et al. [SOSA 2022]. We push the horizon by giving an $O(1)$ -approximation for the Non-Uniform k -Center for 4 distinct types of radii. Our result is obtained via a novel combination of tools and techniques from the k -center literature, which also demonstrates that the different generalizations of k -center involving non-uniform radii, and multiple coverage constraints (i.e., *colorful k -center*), are closely interlinked with each other. We hope that our ideas will contribute towards a deeper understanding of the t -NU k C problem, eventually bringing us closer to the resolution of the CGK conjecture.

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

Keywords and phrases k -center, approximation algorithms, non-uniform k -center, clustering

Digital Object Identifier 10.4230/LIPIcs.SWAT.2022.28

Related Version *Full Version*: <https://arxiv.org/abs/2111.06362>

Funding *Tanmay Inamdar*: Supported by the European Research Council (ERC) via grant LOPPRE, reference 819416.

Kasturi Varadarajan: Supported by National Science Foundation (NSF) Award CCF-1615845.

1 Introduction

The k -center problem is one of the most fundamental problems in clustering. The input to the k -center problem consists of a finite metric space (X, d) , where X is a set of n points, and $d : X \times X \rightarrow \mathbb{R}^+$ is the associated distance function satisfying triangle inequality. We are also given a parameter k , where $1 \leq k \leq n$. A solution to the k -center problem consists of a set $C \subseteq X$ of size at most k , and the cost of this solution is $\max_{p \in X} d(p, C)$, i.e., the maximum distance of a point to its nearest center in C . Alternatively, a solution can be thought of as a set of k balls of radius $\max_{p \in X} d(p, C)$, centered around points in C , that covers the entire set of points X . The goal is to find a solution of smallest radius. We say that a solution C' is an α -approximation, if the cost of C' is at most α times the optimal radius. Several 2-approximations are known for the k -center problem [10, 9]. A simple reduction from the Minimum Dominating Set problem shows that the k -center problem is NP-hard. In fact, the same reduction also shows that it is NP-hard to get a $(2 - \epsilon)$ -approximation for any $\epsilon > 0$.



© Tanmay Inamdar and Kasturi Varadarajan;

licensed under Creative Commons License CC-BY 4.0

18th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2022).

Editors: Artur Czumaj and Qin Xin; Article No. 28; pp. 28:1–28:20

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Several generalizations of the vanilla k -center problem have been considered in the literature, given its fundamental nature in the domain of clustering and approximation algorithms. One natural generalization is the *Robust k -center* or *k -center with outliers* problem, where we are additionally given a parameter m , and the goal is to find a solution that covers at least m points of X . Note that the remaining at most $n - m$ points can be thought of as outliers with respect to the clustering computed. Charikar et al. [7], who introduced this problem, showed that a simple greedy algorithm gives a 3-approximation for the problem. Subsequently, the approximation guarantee was improved by [4, 8], who gave a 2-approximation, which is optimal in light of the aforementioned $(2 - \epsilon)$ -hardness result.

The focus of our paper is the Non-Uniform k -Center (NUkC), which was introduced by Chakrabarty et al. [4]. A formal definition follows.

► **Definition 1** (*t*-NUkC). *The input is an instance $\mathcal{I} = ((X, d), (k_1, k_2, \dots, k_t), (r_1, r_1, \dots, r_t))$, where $r_1 \geq r_2 \geq \dots \geq r_t \geq 0$, and the k_i are positive integers. The goal is to find sets $C_i \subseteq X$ for $1 \leq i \leq t$, such that $|C_i| \leq k_i$, and the union of balls of radius αr_i around the centers in C_i , over $1 \leq i \leq t$, covers the entire set of points X . The objective is to minimize the value of the dilation factor α .*

In the Robust *t*-NUkC problem, we are required to cover at least m points of X using such a solution. We note that the special case of (Robust) *t*-NUkC with $t = 1$ corresponds to the (Robust) k -center problem. Chakrabarty et al. [4] gave a bicriteria approximation for *t*-NUkC for arbitrary t , i.e., they give a solution containing $O(k_i)$ balls of radius $O(\alpha^*)r_i$ for $1 \leq i \leq t$, where α^* is the optimal dilation.¹ They also give a $(1 + \sqrt{5})$ -approximation for 2-NUkC. Furthermore, they conjectured that there exists a polynomial-time $O(1)$ -approximation for *t*-NUkC for constant t . Subsequently, Chakrabarty and Negahbani [6] made some progress by giving a 10-approximation for Robust 2-NUkC. Very recently, Jia et al. [12] showed an approximate equivalence between $(t + 1)$ -NUkC and Robust *t*-NUkC, thereby observing that the previous result of [6] readily implies a 23-approximation for 3-NUkC. We note that the techniques from Inamdar and Varadarajan [11] implicitly give an $O(1)$ -approximation for *t*-NUkC for any $t \geq 1$, in $k^{O(k)} \cdot n^{O(1)}$ time, where $k = \sum_t k_t$. That is, one gets an *FPT approximation*. Finally, we also note that Bandyapadhyay [2] gave an exact algorithm for perturbation resilient instances of NUkC in polynomial time.

Another related variant of k -center is the *Colorful k -center* problem. Here, the set of points X is partitioned into ℓ color classes, $X_1 \sqcup \dots \sqcup X_\ell$. Each color class X_j has a coverage requirement m_j , and the goal is to find a set of k balls of smallest radius that satisfy the coverage requirements of all the color classes. Note that this is a generalization of Robust k -center to multiple types of coverage constraints. Bandyapadhyay et al. [3] introduced this problem, and gave a *pseudo-approximation*, i.e., their algorithm returns an 2-approximate solution using at most $k + \ell - 1$ centers. Furthermore, they managed to improve this to a true $O(1)$ -approximation in the Euclidean plane for constant number of color classes. Subsequently, Jia et al. [13] and Anegg et al. [1] independently gave (true) 3 and 4-approximations respectively for this problem (with constant ℓ) in arbitrary metrics.

¹ Chakrabarty et al. [4] use a slightly different, but equivalent, formulation of NUkC: the input contains a sequence of allowed radii $r_1 \geq r_2 \geq \dots \geq r_k$, and we are allowed to use one ball of radius αr_i , for each i . In this setting, they design an approximation algorithm that returns a solution containing $O(1)$ balls of radius $O(\alpha^*)r_i$, for each i . When there are only t distinct classes of radii, this is equivalent to the result as formulated above.

Our Results and Techniques. Our main result is an $O(1)$ -approximation for 4-NUkC. We obtain this result via a sequence of reductions; some of these reductions are from prior work while some are developed here and constitute our main contribution. Along the way, we combine various tools and techniques from the aforementioned literature of Robust, Colorful, and Non-Uniform versions of k -center.

First, we reduce the 4-NUkC problem to the Robust 3-NUkC problem, following Jia et al. [12]. Next, we reduce the Robust 3-NUkC to *well-separated* Robust 3-NUkC, by adapting the approach of Chakrabarty and Negahbani [6].² In a well-separated instance, we are given a set of potential centers for the balls of radius r_1 , such that the distance between any two of these centers is at least $c \cdot r_1$, for a parameter $c \geq 2$.

To solve Well-Separated Robust 3-NUkC, we give a sequence of reductions, which constitute the technical core of our paper. First, we show that any instance of Robust t -NUkC can be transformed to an instance of “Colorful” $(t - 1)$ -NUkC, where we want to cover certain number of red and blue points using the specified number of balls of $t - 1$ distinct radii. Thus, this reduction reduces the number of radii classes from t to $t - 1$ at the expense of increasing the number of coverage constraints from 1 to 2. In our next reduction, we show that Colorful $(t - 1)$ -NUkC can be reduced to Colorful $(t - 1)$ -NUkC with an additional “self-coverage” property, i.e., the radius r_{t-1} can be assumed to be 0. Just like the aforementioned reduction from [12], these two reductions are generic, and hold for any value of $t \geq 2$. These reductions crucially appeal to the classical greedy algorithm and its analysis from Charikar et al. [7], which is a tool that has not been exploited in the NUkC literature thus far. We believe that these connections between Colorful and Robust versions of NUkC are interesting in their own right, and may be helpful toward obtaining a true $O(1)$ -approximation for t -NUkC for any fixed t . Indeed, one possible avenue to this result is to find suitable generalizations of some of our reductions.

We apply these two new reductions to transform Well-Separated Robust 3-NUkC to *Well-Separated Colorful* 2-NUkC, with $r_2 = 0$. The latter problem can be solved in polynomial time using dynamic programming in a straightforward way – the details can be found in Section 5. Since each of our reductions preserves the approximation factor up to a constant, this implies an $O(1)$ -approximation for 4-NUkC.

Our overall algorithm for 4-NUkC is combinatorial, except for the step where we reduce Robust 3-NUkC to Well-Separated Robust 3-NUkC using the round-or-cut approach of [6]. Thus, we avoid an additional “inner loop” of round-or-cut that is employed in recent work [6, 12].³

We conclude this section by explaining the bottleneck in employing the techniques used in this paper to obtain an $O_t(1)$ approximation for t -NUkC. For concreteness, we focus on 5-NUkC. Using the general reductions from prior work and this paper, we can reduce 5-NUkC to Well-Separated Colorful 3-NUkC, with $r_3 = 0$ (self-coverage at level 3). However, we do not know how to solve this problem exactly via DP, or even obtain an $O(1)$ approximation. The difficulty pertains to the balls of radius $r_2 > 0$, which are not constrained in any way. It is not straightforward to use our techniques to also obtain self-coverage at level 2 ($r_2 = 0$); doing so would result in a problem that can be solved exactly using DP.

² In this discussion, “reduction” refers to a polynomial time (possibly Turing) reduction from problem A to problem B , such that (i) a feasible instance of A yields (possibly polynomially many) instance(s) of B , and (ii) a constant approximation for B implies a constant approximation for A .

³ A by-product of one of our reductions is a purely combinatorial approximation algorithm for colorful k -center, in contrast with the LP-based approaches in [3, 1, 13].

2 Definitions, Main Result, and Greedy Clustering

2.1 Problem Definitions

In the following, we set up the basic notation and define the problems we will consider in the paper. We consider a finite metric space (X, d) , where X is a finite set of (usually n) points, and d is a distance function satisfying triangle inequality. If Y is a subset of X , then by slightly abusing the notation, we use (Y, d) to denote the metric space where the distance function d is restricted to the points of Y . Let $p \in X$, $Y \subseteq X$, and $r \geq 0$. Then, we use $d(p, Y) := \min_{y \in Y} d(p, y)$, and denote by $B(p, r)$ the ball of radius r centered at p , i.e., $B(p, r) := \{q \in X : d(p, q) \leq r\}$. We say that a ball $B(p, r)$ covers a point q iff $q \in B(p, r)$; a set of balls \mathcal{B} (resp. a tuple of sets of balls $(\mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_t)$) covers q if there exists a ball in \mathcal{B} that covers q (resp. $\bigcup_{1 \leq i \leq t} \mathcal{B}_i$ that covers q). Analogously, a set of points $Y \subseteq X$ is covered iff every point in Y is covered. For a function $f : S \rightarrow \mathbb{R}^+$ or $f : S \rightarrow \mathbb{N}$, and $R \subseteq S$, we define $f(R) := \sum_{r \in R} f(r)$.

► **Definition 2** (Decision Version of t -NUkC).

The input is an instance $\mathcal{I} = ((X, d), (k_1, k_2, \dots, k_t), (r_1, r_2, \dots, r_t))$, where $r_1 \geq r_2 \geq \dots \geq r_t \geq 0$, and each k_i is a non-negative integer. The goal is to determine whether there exists a solution $(\mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_t)$, where for each $1 \leq i \leq t$, \mathcal{B}_i is a set with at most k_i balls of radius r_i , that covers the entire set of points X . Such a solution is called a feasible solution, and if the instance \mathcal{I} has a feasible solution, then \mathcal{I} is said to be feasible.

An algorithm is said to be an α -approximation algorithm (with $\alpha \geq 1$), if given a feasible instance \mathcal{I} , it returns a solution $(\mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_t)$, where for each $1 \leq i \leq t$, \mathcal{B}_i is a collection of at most k_i balls of radius αr_i , such that the solution covers X .

Next, we define the robust version of t -NUkC.

► **Definition 3** (Decision Version of Robust t -NUkC).

The input is an instance $\mathcal{I} = ((X, d), (\omega, m), (k_1, k_2, \dots, k_t), (r_1, r_2, \dots, r_t))$. The setup is the same as in t -NUkC, except for the following: $\omega : X \rightarrow \mathbb{Z}^+$ is a weight function, and $1 \leq m \leq \omega(X)$ is a parameter. The goal is to determine whether there exists a feasible solution, i.e., $(\mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_t)$ of appropriate sizes and radii (as defined above), such that the total weight of the points covered is at least m . An α -approximate solution covers points of weight at least m while using at most k_i balls of radius αr_i for each $1 \leq i \leq t$.

We will frequently consider the *unweighted* version of Robust t -NUkC, i.e., where the weight of every point in X is unit. Let $\mathbb{1}$ denote this unit weight function. Now we define the Colorful t -NUkC problem, which generalizes Robust t -NUkC.

► **Definition 4** (Decision Version of Colorful t -NUkC).

The input is an instance $\mathcal{I} = ((X, d), (\omega_r, \omega_b, m_r, m_b), (k_1, k_2, \dots, k_t), (r_1, r_2, \dots, r_t))$. The setup is similar as in Robust t -NUkC, except that we have two weight functions $\omega_r, \omega_b : X \rightarrow \mathbb{Z}^+$ (corresponding to red and blue weight respectively). A feasible solution covers a set of points with red weight at least m_r , and blue weight at least m_b . The notion of approximation is the same as above.

We note that the preceding definition naturally extends to an arbitrary number $\chi \geq 2$ of colors (i.e., χ different weight functions over X). However, we will not need that level of generality in this paper.

2.2 Main Algorithm for 4-NUkC

Let $\mathcal{I} = ((X, d), (k_1, \dots, k_4), (r_1, \dots, r_4))$ be the given instance of 4-NUkC, which we assume is feasible. First, using a reduction from [12] (or a variant described in Section 6), we reduce \mathcal{I} to an instance $\mathcal{I}' = ((X, d), (\mathbb{1}, m)(r'_1, r'_2, r'_3), (k_1, k_2, k_3))$ of Robust 3-NUkC. The reduction has the property that \mathcal{I}' is feasible, and furthermore an $O(1)$ -approximation for \mathcal{I}' implies an $O(1)$ -approximation for \mathcal{I} .

Next, we perform a Turing reduction from Robust 3-NUkC to *Well-Separated* Robust 3-NUkC, by adapting a very similar reduction of [6]. In a well-separated instance, we are given a set of potential centers for the balls of radius r'_1 , such that the distance between any two potential centers is at least $3r'_1$. This Turing reduction is described in Sections A and B of the appendix, and uses the round-or-cut methodology on the instance \mathcal{I}' . At a high level, we run the ellipsoid algorithm, and each iteration of the ellipsoid algorithm returns a candidate LP solution such that, (1) it can be rounded to obtain an $O(1)$ -approximate solution for \mathcal{I}' , or (2) one can obtain polynomially many instances of *well-separated* Robust 3-NUkC, at least one of which is feasible, or (3) If none of the obtained instances is feasible, then one can obtain a hyperplane separating the LP solution from the integer hull of coverages.

Solving a Well-Separated Instance. For the sake of simplicity let \mathcal{J} be one of the instances of Well-Separated Robust 3-NUkC, along with a well-separated set Y that is a candidate set for the centers of balls of radius r'_1 . Furthermore, let us assume that \mathcal{J} is feasible. First, the reduction in Section 3 (Theorem 11), given the instance \mathcal{J} , produces $O(n)$ instances $\mathcal{J}(\ell)$ of Colorful 2-NUkC, such that at least one of the instances is feasible. Then, we apply the reduction from Section 4 (Theorem 17) on each of these instances to ensure the *self-coverage* property, i.e., we obtain an instance $\mathcal{J}'(\ell)$ of Colorful 2-NUkC with $r''_1 = r'_1 + c_2 r'_2 + c_3 r'_3$ and, crucially, $r''_2 = 0$. Finally, assuming that the resulting instance $\mathcal{J}'(\ell)$ is feasible, it is possible to find a feasible solution using straightforward dynamic programming, as described in Section 5. This dynamic programming only requires that the instance is *Well-Separated* w.r.t. a smaller separation factor of 2. We argue in the next paragraph that this property holds in each of the instances $\mathcal{J}'(\ell)$.

In the reductions in Sections 3 and Section 4, the solution transformations preserve the centers for the balls of the largest radius class. In order to show that the set Y of candidate centers is well-separated w.r.t. the new top level radius r''_1 , we need to show that $3r'_1 \geq 2r''_1$, i.e., $r'_1 \geq 2c_2 r'_2 + 2c_3 r'_3 \geq \beta \cdot r'_2$ for some sufficiently large constant β . This assumption is without loss of generality, since, if two consecutive radii classes in instance \mathcal{J} are within a β factor, it is possible to combine them into a single radius class, at the expense of an $O(\beta)$ factor in the approximation guarantee.

Assuming the instance \mathcal{J} is feasible, a feasible solution to an instance $\mathcal{J}'(\ell)$ can be mapped back to an $O(1)$ -approximate solution to \mathcal{J} , and then to \mathcal{I} , since each reduction preserves the approximation guarantee up to an $O(1)$ factor.

► **Theorem 5.** *There exists a polynomial time $O(1)$ -approximation algorithm for 4-NUkC.*

We have overviewed how the various sections of the paper come together in deriving Theorem 5. Before proceeding to these sections, we describe a greedy clustering procedure that we need.

2.3 Greedy Clustering

Assume we are given (i) a metric space (X, d) , where X is finite, (ii) a radius $r \geq 0$, (iii) an expansion parameter $\gamma \geq 1$, (iv) a subset $Y \subseteq X$ and a weight function $\omega : Y \rightarrow \mathbb{Z}^+$. The weight $\omega(y)$ can be thought of as the multiplicity of $y \in Y$, or how many points are co-located

28:6 Non-Uniform k -Center and Greedy Clustering

at y . We describe a greedy clustering procedure, from Charikar et al. [7], that is used to partition the point set Y into clusters, each of which is contained in a ball of radius γr . This clustering procedure, together with its properties, is a crucial ingredient of our approach.

■ **Algorithm 1** GREEDYCLUSTERING($Y, X, r \geq 0, \gamma \geq 1, \omega : Y \rightarrow \mathbb{Z}^+$).

We require that $Y \subseteq X$

- 1: Let $U \leftarrow Y, M \leftarrow \emptyset$
- 2: **while** $U \neq \emptyset$ **do**
- 3: $p = \arg \max_{q \in X} \omega(U \cap B(q, r))$
- 4: $C(p) := U \cap B(p, \gamma r); wt(p) := \omega(C(p))$
- 5: $U \leftarrow U \setminus C(p)$
- 6: $M \leftarrow M \cup \{p\}$ ▷ We refer to $C(p)$ as a cluster whose center is p
- 7: **end while**
- 8: **return** $(M, \{C(p)\}_{p \in M}, \{wt(p)\}_{p \in M})$

In line 3, we only consider $q \in X$ such that $U \cap B(q, r) \neq \emptyset$. Notice that it is possible that $\omega(U \cap B(q, r)) = 0$ if $\omega(y) = 0$ for each $y \in U$. Furthermore, notice that we do not require that $q \in U$ for it to be an eligible point in line 3.

We summarize some of the key properties of this algorithm in the following observations.

- **Observation 6. 1.** For any $p \in M$, $C(p) \subseteq B(p, \gamma r)$,
- 2. Point $y \in Y$ belongs to the cluster $C(p)$, such that p is the first among all $q \in M$ satisfying $d(y, q) \leq \gamma r$.
- 3. The sets $\{C(p)\}_{p \in M}$ partition Y , which implies that
- 4. $\sum_{p \in M} wt(p) = \omega(Y)$, where $\omega(Z) = \sum_{z \in Z} \omega(z)$ for any $Z \subseteq Y$.
- 5. If p_i and p_j are the points added to M in iterations $i \leq j$, then $wt(p_i) \geq wt(p_j)$.
- 6. For any two distinct $p, q \in M$, $d(p, q) > (\gamma - 1)r$.

Proof. The first five properties are immediate from the description of the algorithm. Now, we prove the sixth property. Suppose for contradiction that there exist $p, q \in M$ with $d(p, q) \leq (\gamma - 1)r$, and without loss of generality, p was added to M before q . Then, note that at the end of this iteration, $B(q, r) \cap U = \emptyset$. Therefore, q will subsequently never be a candidate for being added to M in line 3. ◀

A key property of this greedy clustering, established by Charikar et al. [7], is that for any $k \geq 1$ balls of radius r , the weight of the points in the first k clusters is at least as large as the weight of the points covered by the k balls.

► **Lemma 7.** Suppose that the parameter γ used in Algorithm 1 is at least 3. Let \mathcal{B} be any collection of $k \geq 1$ balls of radius r , each centered at a point in X . Let M' consist of the first k' points of M chosen by the algorithm, where $k' = \min\{k, |M|\}$. We have

$$\sum_{p \in M'} wt(p) = \omega \left(\bigcup_{p \in M'} C(p) \right) \geq \omega \left(Y \cap \bigcup_{B \in \mathcal{B}} B \right).$$

The equality follows from the definition of $wt(p)$ and the fact that the clusters partition Y , as stated in Observation 6.

3 From Robust t -NUkC to Colorful $(t - 1)$ -NUkC

Let $\mathcal{I} = ((X, d), (\omega, m), (k_1, k_2, \dots, k_t), (r_1, r_2, \dots, r_t))$ be an instance of Robust t -NUkC. The reduction to Colorful $(t - 1)$ -NUkC consists of two phases. In the first phase, we use Algorithm 1 to reduce the instance \mathcal{I} to an instance \mathcal{I}' focused on the cluster centers output by the greedy algorithm. A key property of this reduction is that we may set $r_t = 0$ in the instance \mathcal{I}' – each ball at level t is allowed to cover at most one point.

In the second phase, we transform \mathcal{I}' to $O(n)$ instances of Colorful $(t - 1)$ -NUkC. Assuming there exists a feasible solution for \mathcal{I}' , at least one of the instances \mathcal{I}'' of Colorful $(t - 1)$ -NUkC has a feasible solution, and any approximate solution to \mathcal{I}'' can be used to obtain an approximate solution to \mathcal{I}' (and thus to \mathcal{I}).

Phase 1. Let $\mathcal{I} = ((X, d), (\omega, m), (k_1, k_2, \dots, k_t), (r_1, r_2, \dots, r_t))$ be an instance of Robust t -NUkC. We call the algorithm $\text{GREEDYCLUSTERING}(X, X, r_t, 3, \omega)$, and obtain a set of points M with the corresponding clusters $C(p)$ for $p \in M$. The greedy algorithm also returns a weight $wt(p) = \omega(C(p))$ for each $p \in M$. Let us number the points of M as p_i , where i is the iteration in which p_i was added to the set M by $\text{GREEDYCLUSTERING}(X, X, r_t, 3, \omega)$. This gives an ordering $\sigma = \langle p_1, p_2, \dots, p_{|M|} \rangle$ of the points in M . Note that $wt(p_i) \geq wt(p_j)$ for $i \geq j$.

We define a weight function $\lambda : X \rightarrow \mathbb{Z}^+$. Let $\lambda(p) = wt(p)$ for $p \in M$ and $\lambda(p) = 0$ for $p \in X \setminus M$. Note that for $p \in M$, $\lambda(p) = wt(p) = \omega(C(p))$. Thus, for each $p \in M$, we are moving the weight from points in cluster $C(p)$ to the cluster center p . Clearly, $\omega(X) = \lambda(X)$.

The output of Phase 1 is the instance $\mathcal{I}' = ((X, d), (\lambda, m), (k_1, k_2, \dots, k_t), (r'_1, r'_2, \dots, r'_{t-1}, 0))$ of Robust t -NUkC, where $r'_i = r_i + 3r_t$. Note that in the instance \mathcal{I}' , we have $r'_t = 0$, whereas the other radii in \mathcal{I} have been increased by an additive factor of $3r_t$. The following claim relates instances \mathcal{I} and \mathcal{I}' .

► **Lemma 8.** (a) If instance \mathcal{I} has a feasible solution, then so does the instance \mathcal{I}' . (b) Given a solution $(\mathcal{B}'_i)_{i \in [t]}$ for \mathcal{I}' that uses at most k_i balls of radius $\alpha r'_i$ for every $i \in [t]$, we can obtain a solution $(\mathcal{B}_i)_{i \in [t]}$ for \mathcal{I} that uses at most k_i balls of radius at most $\alpha r'_i + 3r_t \leq \alpha r_i + (3\alpha + 3)r_t$ for $1 \leq i \leq t$.

Proof. We begin with part (b). For each ball in $B(p, r)$ that is part of the solution $(\mathcal{B}'_i)_{i \in [t]}$, we replace it with the ball $B(p, r + 3r_t)$ to obtain a solution $(\mathcal{B}_i)_{i \in [t]}$ for \mathcal{I} . That is, we expand each ball by an additive $3r_t$. If $B(p, r)$ covers $q \in M$, then $B(p, r + 3r_t)$ covers $C(q)$, and $\lambda(q) = \omega(C(q))$. Let $M' \subseteq M$ denote the points covered by $(\mathcal{B}'_i)_{i \in [t]}$. The weight of the points covered by $(\mathcal{B}_i)_{i \in [t]}$ is at least

$$\sum_{p \in M'} \omega(C(p)) = \sum_{p \in M'} \lambda(p) \geq m.$$

We now establish (a). Fix a feasible solution $(\mathcal{B}_i)_{i \in [t]}$ to \mathcal{I} that covers ω -weight at least m , where \mathcal{B}_i is a set of at most k_i balls of radius r_i , for $i \in [t]$. Let $M_1 \subseteq M$ be the set of points p such that some point in $C(p)$ is covered by a ball in $\mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_{t-1}$.

Now let $M_2 = M \setminus M_1$ be the set of points p , such that any point in $C(p)$ is either covered by a ball from \mathcal{B}_t , or is an outlier. Let $X_i := \bigcup_{p \in M_i} C(p)$ for $i = 1, 2$. Note that $X = X_1 \sqcup X_2$.

Note that in the sequence $\sigma = \langle p_1, p_2, \dots, p_{|M|} \rangle$, the points of M_1 and M_2 may appear in an interleaved fashion. Let $p_{i_1}, p_{i_2}, \dots, p_{i_{|M_2|}}$ be the subsequence restricted to the points in M_2 . In the following lemma, we argue that the first k_t points in this subsequence are sufficient to replace the balls in \mathcal{B}_t . Let $k' = \min\{|\mathcal{B}_t|, |M_2|\} \leq k_t$.

► **Lemma 9.** *There exists a subset $M_2^+ \subseteq M_2$ of size at most k' such that $\sum_{p \in M_2^+} wt(p) \geq \omega(X_2 \cap \bigcup_{\text{Ball} \in \mathcal{B}_t} \text{Ball})$.*

Proof. Let $M_2^+ = \{p_{i_1}, p_{i_2}, \dots, p_{i_{k'}}\}$. That is, M_2^+ consists of the first k' points of M_2 picked by the greedy algorithm. Recall that $M_2^+ \subseteq M_2$, and thus for $p_{i_j} \in M_2^+$, it holds that $C(p_{i_j}) \subseteq X_2$.

Now imagine calling the algorithm $\text{GREEDYCLUSTERING}(X_2, X, r_t, 3, \omega)$. Observe that in the iteration $1 \leq j \leq |M_2|$, this algorithm will select point p_{i_j} (as defined above) in Line 3, and the corresponding cluster and its weight will be $C(p_{i_j})$ and $wt(p_{i_j})$ – exactly as in the execution of $\text{GREEDYCLUSTERING}(X, X, r_t, 3, \omega)$. That is, the algorithm $\text{GREEDYCLUSTERING}(X_2, X, r_t, 3, \omega)$ will output M_2 and the clusters $C(p)$ for each $p \in M_2$.

Now, \mathcal{B}_t consists of a set of $|\mathcal{B}_t|$ balls of radius r_t . The lemma now follows from Lemma 7 applied to $\text{GREEDYCLUSTERING}(X_2, X, r_t, 3, \omega)$. ◀

Using Lemma 9, we now construct a solution to instance \mathcal{I}' . Fix index $1 \leq i \leq t-1$, and \mathcal{B}'_i denote the set of balls obtained by expanding each ball in \mathcal{B}_i by an additive $3r_t$. Note that each ball in \mathcal{B}'_i has radius $r'_i = r_i + 3r_t$. For every point $p \in M_2^+$, we add a ball of radius 0 around it and let \mathcal{B}'_t be the resulting set of balls. Note that $|\mathcal{B}'_t| = |M_2^+| \leq k' \leq k_t$.

By definition, for each point $p \in M_1$, there is a ball in $(\mathcal{B}_i)_{i \in [t-1]}$ that intersects cluster $C(p)$, whose points are at distance at most $3r_t$ from p . It follows that the balls in $(\mathcal{B}'_i)_{i \in [t-1]}$ cover each point in M_1 .

Using Lemma 9, the coverage of $(\mathcal{B}'_i)_{i \in [t]}$ in instance \mathcal{I}' is at least

$$\sum_{p \in M_1} wt(p) + \sum_{p \in M_2^+} wt(p) \geq \omega(X_1) + \omega\left(X_2 \cap \bigcup_{\text{Ball} \in \mathcal{B}_t} \text{Ball}\right) \geq m.$$

The final inequality follows because any point covered by solution $(\mathcal{B}_i)_{i \in [t]}$ for \mathcal{I} either belongs to X_1 or to $X_2 \cap \bigcup_{\text{Ball} \in \mathcal{B}_t} \text{Ball}$. Thus, we have shown that \mathcal{I}' has a feasible solution. ◀

Phase 2. Now we describe the second phase of the algorithm. We have the instance $\mathcal{I}' = ((X, d), (\lambda, m), (k_1, k_2, \dots, k_t), (r'_1, r'_2, \dots, r'_{t-1}, 0))$ of Robust t -NUkC that is output by Phase 1. Phase 2 takes \mathcal{I}' as input and generates an instance $\mathcal{I}(\ell)$, for each $0 \leq \ell \leq |X|$, of the Colorful $(t-1)$ -NUkC problem. Note that the number of generated instances is $|X| + 1 = O(n)$. If \mathcal{I}' is feasible, at least one of these $|X| + 1$ instances will be feasible.

Let $\sigma = \langle p_1, p_2, \dots, p_{|X|} \rangle$ be an ordering of the points in X by non-increasing λ . That is, $\lambda(p_i) \geq \lambda(p_j)$ for $i \leq j$.

Fix an index $0 \leq \ell \leq |X|$. We now describe the instance $\mathcal{I}(\ell)$ of colorful $(t-1)$ -NUkC. Let $R = \{p_1, p_2, \dots, p_\ell\}$ denote the set of *red* points, and $B = \{p_{\ell+1}, p_{\ell+2}, \dots, p_{|X|}\}$ denote the set of *blue* points. For each $p \in B$, define its blue weight as $\omega_b(p) := \lambda(p)$; for each $p \in R$, define its blue weight as $\omega_b(p) := 0$. Define the blue coverage m_b for instance $\mathcal{I}(\ell)$ as $m_b := m - \lambda(R)$. We define the red weight function ω_r in a slightly different manner. For each red point $p \in R$, let its red weight $\omega_r(p) := 1$; for each $p \in B$, let red weight $\omega_r(p) := 0$. Let $m_r := \sum_{p \in R} \omega_r(p) - k_t = |R| - k_t$ denote the red coverage for instance $\mathcal{I}(\ell)$. Note that ω_r is supported on R and ω_b on B . Let $\mathcal{I}(\ell) := ((X, d), (\omega_r, \omega_b, m_r, m_b), (k_1, k_2, \dots, k_{t-1}), (r'_1, r'_2, \dots, r'_{t-1}))$ denote the resulting instance of Colorful $(t-1)$ -NUkC problem. Recall that a solution to this instance is required to cover red weight that adds up to at least m_r , and blue weight that adds up to at least m_b . (In instance $\mathcal{I}(\ell)$, the point sets R and B , the red and blue weights, and total coverage requirements m_r and m_b all depend on the index ℓ . This dependence is not made explicit in the notation, so as to keep it simple.)

We now relate the instance \mathcal{I}' to the instances $\mathcal{I}(\ell)$, for $0 \leq \ell \leq |X|$.

► **Lemma 10.** (a) *If the instance $\mathcal{I}' = ((X, d), (\lambda, m), (k_1, k_2, \dots, k_t), (r'_1, r'_2, \dots, r'_{t-1}, 0))$ is feasible, then there exists an $0 \leq \ell^* \leq |X|$ such that instance $\mathcal{I}(\ell^*)$ is feasible.*

(b) *Let $\mathcal{I}(\ell) = ((X, d), (\omega_r, \omega_b, m_r, m_b), (k_1, k_2, \dots, k_{t-1}), (r'_1, r'_2, \dots, r'_{t-1}))$ be a generated instance of Colorful $(t-1)$ -NUkC, and suppose $(\mathcal{B}''_i)_{i \in [t-1]}$ is a solution to this instance such that \mathcal{B}''_i contains at most k_i balls of radius $\alpha r'_i$ for $1 \leq i \leq t-1$, and covers red weight at least m_r and blue weight at least m_b . Then, we can efficiently obtain a solution to the instance \mathcal{I}' that uses at most k_i balls of radius $\alpha r'_i$ for $1 \leq i \leq t-1$, and at most k_t balls of radius 0.*

Proof. We first show part (b). In instance $\mathcal{I}(\ell)$, the red weight $\omega_r(p) = 1$ for each $p \in R$, so the solution $(\mathcal{B}''_i)_{i \in [t-1]}$ covers at least $m_r = \sum_{p \in R} \omega_r(p) - k_t = |R| - k_t$ red points. So the number of red points that are not covered is at most k_t . Construct \mathcal{B}'_t by adding a ball of radius 0 at each uncovered point in R . Thus, $|\mathcal{B}'_t| \leq k_t$.

Let $\mathcal{B}'_i = \mathcal{B}''_i$ for each $1 \leq i \leq t-1$. Now, we argue that the solution $(\mathcal{B}'_i)_{i \in [t]}$ covers weight at least m in instance \mathcal{I}' . Note that this solution covers all points in R , and a subset $C \subseteq B$ such that $\omega_b(C) \geq m_b = m - \lambda(R)$. Thus the coverage for \mathcal{I}' is at least

$$\lambda(R) + \lambda(C) = \lambda(R) + \omega_b(C) \geq \lambda(R) + m_b = m.$$

We now turn to part (a). Fix a feasible solution $(\mathcal{B}'_i)_{i \in [t]}$ to \mathcal{I}' . Let $M_1 \subseteq X$ denote the subset consisting of each point covered by a ball in \mathcal{B}'_i , for $1 \leq i \leq t-1$. Let $M_2 = X \setminus M_1$. Each point in M_2 is either an outlier or is covered by a ball in \mathcal{B}'_t . Note that in the sequence $\sigma = \langle p_1, p_2, \dots, p_{|X|} \rangle$, the points of M_1 and M_2 may appear in an interleaved fashion. Let $p_{i_1}, p_{i_2}, \dots, p_{i_{|M_2|}}$ be the subsequence restricted to the points in M_2 . Let $k' = \min\{k_t, |M_2|\}$, and let $M_2^+ = \{p_{i_1}, p_{i_2}, \dots, p_{i_{k'}}\}$. A key observation is that $\lambda(M_2^+)$ is at least as large as the total weight of the points in M_2 covered by balls in \mathcal{B}'_t . This is because each ball in \mathcal{B}'_t has radius 0 and can cover only one point in M_2 ; and the maximum coverage using such balls is obtained by placing them at the points in M_2 with the highest weights, i.e., M_2^+ . Without loss of generality, we assume that \mathcal{B}'_t consists of balls of radius 0 placed at each point in M_2^+ .

Now, let the index $\ell^* := i_{k'}$. We now argue that the instance $\mathcal{I}(\ell^*)$ of colorful $(t-1)$ -NUkC is feasible. In particular, we argue that $(\mathcal{B}'_i)_{i \in [t-1]}$ is a solution. Consider the set $R = \{p_1, p_2, \dots, p_{\ell^*}\}$ of red points in $\mathcal{I}(\ell^*)$. Each point in R is either in M_1 or in M_2^+ , and is therefore covered by $(\mathcal{B}'_i)_{i \in [t]}$. It follows that $(\mathcal{B}'_i)_{i \in [t-1]}$ covers at least $|R| - |\mathcal{B}'_t| \geq |R| - |k_t| = m_r$ points of R . In other words, the red weight in $\mathcal{I}(\ell^*)$ covered by $(\mathcal{B}'_i)_{i \in [t-1]}$ is at least m_r .

Now consider the set $B = \{p_{\ell^*+1}, p_{\ell^*+2}, \dots, p_{|X|}\}$ of blue points in $\mathcal{I}(\ell^*)$. Let $C \subseteq B$ denote the blue points covered by solution $(\mathcal{B}'_i)_{i \in [t]}$. As $(\mathcal{B}'_i)_{i \in [t]}$ covers points with weight at least m in instance \mathcal{I}' , we have $\lambda(R) + \lambda(C) \geq m$; thus, $\lambda(C) \geq m - \lambda(R) = m_b$. However, the balls in \mathcal{B}'_t do not cover any point in B . Thus, the balls in $(\mathcal{B}'_i)_{i \in [t-1]}$ cover all points in C . For any $p \in B$, we have $\lambda(p) = \omega_b(p)$. It follows that the blue weight in $\mathcal{I}(\ell^*)$ covered by $(\mathcal{B}'_i)_{i \in [t-1]}$ is at least $\omega_b(C) = \lambda(C) \geq m_b$. This concludes the proof of part (a). ◀

Combining Lemmas 8 and 10 from Phases 1 and 2, we obtain the following reduction from robust t -NUkC to colorful $(t-1)$ -NUkC.

► **Theorem 11.** *There is a polynomial-time algorithm that, given an instance $\mathcal{I} = ((X, d), (\omega, m), (k_1, \dots, k_t), (r_1, \dots, r_t))$ of Robust t -NUkC, outputs a collection of $O(n)$ instances of Colorful $(t-1)$ -NUkC with the following properties: (a) If \mathcal{I} is feasible, then at least one of the instances $\mathcal{I}(\ell) = ((X, d), (\omega_r, \omega_b, m_r, m_b), (k_1, \dots, k_{t-1}), (r'_1, \dots, r'_{t-1}))$ of Colorful $(t-1)$ -NUkC is feasible; (b) given an α -approximate solution to some instance $\mathcal{I}(\ell)$, we can efficiently construct a solution to \mathcal{I} that uses at most k_i balls of radius at most $\alpha r_i + (3\alpha + 3)r_t$.*

► **Remark 12.** In part (a), the feasible solution for $\mathcal{I}(\ell)$ that is constructed from the feasible solution for \mathcal{I} has the following useful property: for any **Ball** of radius $r'_i = r_i + 3r_t$ in the feasible solution for $\mathcal{I}(\ell)$, the center of **Ball** is also the center of some ball of radius r_i in the feasible solution for \mathcal{I} .

4 Ensuring Self-Coverage in Colorful 2-NU k C

We assume that we are given as input a Colorful 2-NU k C instance $\mathcal{I} = ((X, d), (\omega_r, \omega_b, m_r, m_b), (k_1, k_2), (r_1, r_2))$. Recall that $\omega_r : X \rightarrow \mathbb{Z}^+$ (resp. $\omega_b : X \rightarrow \mathbb{Z}^+$) is the red (resp. blue) weight function. The task in Colorful 2-NU k C is to find a solution $(\mathcal{B}_1, \mathcal{B}_2)$ such that (1) $|\mathcal{B}_i| \leq k_i$ for $i = 1, 2$, and (2) the point set $Y \subseteq X$ covered by the solution satisfies $\omega_r(Y) \geq m_r$ and $\omega_b(Y) \geq m_b$, (i.e., the solution covers points with total red weight at least m_r , and blue weight at least m_b .) In this section, we show that \mathcal{I} can be reduced to an instance of Colorful 2-NU k C with $r_2 = 0$. The fact that each ball of radius r_2 can only cover its center in the target instance is what we mean by the term *self-coverage*. This reduction actually generalizes to Colorful t -NU k C, but we address the case $t = 2$ to keep the notation simpler.

Our reduction proceeds in two phases. In Phase 1, we construct an intermediate instance where we can ensure blue self-coverage. Then in Phase 2, we modify the intermediate instance so as to obtain red self-coverage as well.

Phase 1. In this step, we call the greedy clustering algorithm using the *blue weight function* ω_b . In particular, we call `GREEDYCLUSTERING`($X, X, r_2, 3, \omega_b$) (See Algorithm 1). This algorithm returns a set of points $M \subseteq X$, where every $p \in M$ has a cluster $C(p)$ and weight $wt(p)$ such that (1) $\{C(p)\}_{p \in M}$ is a partition of X ; (2) for any $p \in M$, $wt(p) = \omega_b(C(p))$, the blue weight of the cluster, and (3) $d(q, p) \leq 3r_2$ for any $q \in C(p)$. Furthermore, the greedy algorithm naturally defines an ordering $\sigma = \langle p_1, p_2, \dots, p_{|M|} \rangle$ of M – this is the order in which the points were added to M .

We define a new weight function $\lambda_b : X \rightarrow \mathbb{Z}^+$ as follows: $\lambda_b(p) := wt(p)$ if $p \in M$ and $\lambda_b(p) := 0$ if $p \in X \setminus M$. Note that for $p \in M$, we have $wt(p) = \omega_b(C(p))$. So the new weight function λ_b is obtained from ω_b by moving weight from each cluster $C(p)$ to its center p .

Phase 1 outputs the intermediate instance $\mathcal{I}' = ((X, d), (\omega_r, \lambda_b, m_r, m_b), (k_1, k_2), (r'_1, r'_2))$ of Colorful 2-NU k C, where $r'_1 = r_1 + 6r_2$ and $r'_2 = 5r_2$. A solution $(\mathcal{B}'_1, \mathcal{B}'_2)$ for \mathcal{I}' is said to be *structured* if it has the following properties.

1. It is a solution to \mathcal{I}' viewed as an instance of Colorful 2-NU k C.
2. Let $Y \subseteq X$, the set of points *self-covered* by solution $(\mathcal{B}'_1, \mathcal{B}'_2)$, consist of points $p \in X$ such that either (a) p is covered by \mathcal{B}'_1 , or (b) p is the center of some ball in \mathcal{B}'_2 . We require that $\lambda_b(Y) \geq m_b$.

Thus, a structured solution covers red weight in the usual way; whereas a ball in \mathcal{B}'_2 can only contribute blue coverage for its center. The following lemma relates instances \mathcal{I} and \mathcal{I}' .

► **Lemma 13.** (a) *If instance \mathcal{I} has a feasible solution, then the instance \mathcal{I}' has a feasible solution that is also structured.* (b) *Given a solution $(\mathcal{B}'_1, \mathcal{B}'_2)$ for \mathcal{I}' that uses at most k_i balls of radius $\alpha r'_i$ for every $i \in \{1, 2\}$, we can obtain a solution $(\mathcal{B}_1, \mathcal{B}_2)$ for \mathcal{I} that uses at most k_i balls of radius $\alpha r'_i + 3r_2 \leq \alpha r_i + (6\alpha + 3)r_2$ for $i \in \{1, 2\}$.*

Part (b) is straightforward as the red weights are unchanged in going from \mathcal{I} to \mathcal{I}' , and the blue weights are moved by at most $3r_2$. (Note that we don't require in part (b) that the solution to \mathcal{I}' be structured.)

In the rest of this section, we establish (a). Fix a feasible solution $(\mathcal{B}_1, \mathcal{B}_2)$ to \mathcal{I} . Thus, (1) $|\mathcal{B}_i| \leq k_i$ for $i = 1, 2$, and (2) the point set $Y \subseteq X$ covered by the solution satisfies $\omega_r(Y) \geq m_r$ and $\omega_b(Y) \geq m_b$, (i.e., the solution covers points with total red weight at least m_r , and blue weight at least m_b .)

Let $M_1 \subseteq M$ be the set of points p such that some point in $C(p)$ is covered by a ball in \mathcal{B}_1 . Now let $M_2 = M \setminus M_1$ be the set of points p such that any point in $C(p)$ is either covered by a ball from \mathcal{B}_2 , or is an outlier. Let $X_i := \bigcup_{p \in M_i} C(p)$ for $i = 1, 2$. Note that $X = X_1 \sqcup X_2$.

We construct a solution $(\mathcal{B}'_1, \mathcal{B}'_2)$ for instance \mathcal{I}' as follows. The set \mathcal{B}'_1 is obtained by expanding each ball in \mathcal{B}_1 by an additive factor of $6r_2$. Thus, the balls in \mathcal{B}'_1 cover X_1 . As in the proof of Lemma 8, we construct a subset $N \subseteq M_2$ of size at most $|\mathcal{B}_2|$. We let \mathcal{B}'_2 consist of the balls of radius $r'_2 = 5r_2$, each centered at a point in N . The set N will have the following properties:

$$\omega_r(X_2 \cap \bigcup_{\text{Ball} \in \mathcal{B}'_2} \text{Ball}) \geq \omega_r(X_2 \cap \bigcup_{\text{Ball} \in \mathcal{B}_2} \text{Ball}) \quad (1)$$

$$\sum_{p \in N} wt(p) \geq \omega_b(X_2 \cap \bigcup_{\text{Ball} \in \mathcal{B}_2} \text{Ball}) \quad (2)$$

It is easy to verify that these two guarantees imply that $(\mathcal{B}'_1, \mathcal{B}'_2)$ is a structured, feasible solution to \mathcal{I}' :

The red weight covered by $(\mathcal{B}'_1, \mathcal{B}'_2)$ is at least

$$\omega_r(X_1) + \omega_r(X_2 \cap \bigcup_{\text{Ball} \in \mathcal{B}'_2} \text{Ball}) \geq \omega_r(X_1) + \omega_r(X_2 \cap \bigcup_{\text{Ball} \in \mathcal{B}_2} \text{Ball}) \geq m_r.$$

The set $M_1 \cup N$ is self-covered by $(\mathcal{B}'_1, \mathcal{B}'_2)$. We have

$$\lambda_b(M_1) + \lambda_b(N) = \omega_b(X_1) + \sum_{p \in N} wt(p) \geq \omega_b(X_1) + \omega_b(X_2 \cap \bigcup_{\text{Ball} \in \mathcal{B}_2} \text{Ball}) \geq m_b.$$

We now describe the construction of N and establish properties (1) and (2). At a high level, this is similar to what we did for M_2^+ in Lemma 8; but it is more involved as we need to ensure that both properties hold.

Let $\widehat{\mathcal{B}}_2 = \{\text{Ball} \in \mathcal{B}_2 \mid \text{Ball} \cap X_2 \neq \emptyset\}$. The set N is obtained via `MAPPING PROCEDURE`, given in Algorithm 2. In particular, we invoke `MAPPING PROCEDURE` $(M_2, \sigma, \widehat{\mathcal{B}}_2, \{C(p)\}_{p \in M_2})$. We describe Algorithm 2 at a high level. First, we map every ball in $\widehat{\mathcal{B}}_2$ to the *first* (according to σ) point q in M_2 whose cluster $C(q)$ has a non-empty intersection with the ball – this is the definition of φ . Now, some points $q \in M_2$ may get mapped by more than one ball. Then, we create a “grouping procedure” that creates pairs (N_ℓ, D_ℓ) as follows. We start from the *first* (according to σ) point q_i that is mapped by at least one ball. We add q_i to N_ℓ , and the balls that were mapped to q_i to the set D_ℓ . Now, if $|\varphi^{-1}(q_i)| > 1$, then we aim to find $|\varphi^{-1}(q_i)| - 1$ additional points after q_i to be added to N_ℓ . Furthermore, it is important in the analysis that these points be *consecutive* according to $\sigma|_{M_2}$. The variable `pending` keeps track of how many additional distinct points need to be added to N_ℓ to match the number of distinct balls in D_ℓ at the current time. Thus, if $|\varphi^{-1}(q_i)| > 1$, we add q_{i+1} to N_ℓ as well. At this stage, it may happen that $\varphi^{-1}(q_{i+1}) \neq \emptyset$. Then, we add $\varphi^{-1}(q_{i+1})$ to D_ℓ , and update the variable `pending` appropriately. If the variable `pending` becomes 0, then $|N_\ell| = |D_\ell|$, at which point the inner while loop terminates. By construction, the points added to N_ℓ form a contiguous sub-sequence of $\sigma|_{M_2}$. We add the pair (N_ℓ, D_ℓ) to \mathcal{T} . At this point, if there still exists a ball of $\widehat{\mathcal{B}}_2$ that does not belong to any D_j with $j \leq \ell$, we

28:12 Non-Uniform k -Center and Greedy Clustering

■ **Algorithm 2** MAPPING PROCEDURE($\widehat{M}, \sigma, \widehat{\mathcal{B}}, \{C(p)\}_{p \in \widehat{M}}$).

```

1: Index the points of  $\widehat{M}$  as  $q_1, q_2, \dots$  according to the ordering  $\sigma$ 
2: For every  $\text{Ball} \in \widehat{\mathcal{B}}, \varphi(\text{Ball}) := q_i$ , where  $q_i \in \widehat{M}$  is the first point  $q$  s.t.  $\text{Ball} \cap C(q) \neq \emptyset$ 
3:  $\ell = 0; \mathcal{T} \leftarrow \emptyset$ 
4: while there exists a  $\text{Ball} \in \widehat{\mathcal{B}}$  that does not belong to any  $D_j$  with  $j \leq \ell$  do
5:    $\ell \leftarrow \ell + 1$ 
6:    $q_i \in \widehat{M} \setminus \bigcup_{j=1}^{\ell-1} N_j$  be the first point  $q$  with  $|\varphi^{-1}(q)| > 0$ 
7:   pending  $\leftarrow |\varphi^{-1}(q_i)| - 1$ 
8:    $N_\ell \leftarrow \{q_i\}, D_\ell \leftarrow \varphi^{-1}(q_i)$ 
9:   while pending  $> 0$  and  $i + 1 \leq |\widehat{M}|$  do
10:     $i \leftarrow i + 1$ 
11:    pending  $\leftarrow$  pending  $+$   $|\varphi^{-1}(q_i)| - 1$ 
12:     $N_\ell \leftarrow N_\ell \cup \{q_i\}, D_\ell \leftarrow D_\ell \cup \varphi^{-1}(q_i)$ 
13:   end while
14:   Add  $(N_\ell, D_\ell)$  to  $\mathcal{T}$ 
15: end while
16: Return  $\mathcal{T}$ 

```

start the construction of the next pair $(N_{\ell+1}, D_{\ell+1})$. Note that in all but the last iteration of the outer while loop, it holds that $|N_\ell| = |D_\ell|$. However, in the last iteration t , the loop may terminate with $|N_t| \leq |D_t|$.

The invocation of MAPPING PROCEDURE($M_2, \sigma, \widehat{\mathcal{B}}_2, \{C(p)\}_{p \in M_2}$) returns $\mathcal{T} = \{(N_1, D_1), (N_2, D_2), \dots, (N_t, D_t)\}$. In the following observation, we summarize a few key properties of this collection of pairs.

► **Observation 14.** $\mathcal{T} = \{(N_1, D_1), (N_2, D_2), \dots, (N_t, D_t)\}$ satisfies the following properties.

1. For each $1 \leq \ell \leq t$, we have $\emptyset \neq N_\ell \subseteq M_2$; Furthermore, the points of N_ℓ form a contiguous subsequence of M_2 ordered according to σ . The sets N_1, N_2, \dots, N_t are pairwise disjoint.
2. For each $1 \leq \ell \leq t$, we have $\emptyset \neq D_\ell \subseteq \widehat{\mathcal{B}}_2$. The sets D_1, \dots, D_t form a partition of $\widehat{\mathcal{B}}_2$.
3. $|N_\ell| = |D_\ell|$ for $\ell < t$, and $|N_t| \leq |D_t|$.

Now we prove the following key lemma.

► **Lemma 15.** For any $1 \leq \ell \leq t$, the following properties hold.

A. For any ball $B(c, r_2) \in D_\ell$, there exists a $q \in N_\ell$ such that $B(c, r_2) \subseteq B(q, 5r_2)$.

B. $\omega_b \left(X_2 \cap \bigcup_{B(c, r_2) \in D_\ell} B(c, r_2) \right) \leq \sum_{p \in N_\ell} wt(p)$.

Proof. For any $\text{Ball} = B(c, r_2) \in D_\ell$, $q_i = \varphi(\text{Ball}) \in N_\ell$. By the definition of q_i , it holds that $C(q_i) \cap \text{Ball} \neq \emptyset$. Therefore, for any point $p \in \text{Ball}$, it holds that $d(p, q_i) \leq d(p, c) + d(c, p') + d(p', q_i) \leq r_2 + r_2 + 3r_2 = 5r_2$, where $p' \in C(q_i) \cap \text{Ball}$. This proves property A.

Let $\mathcal{X}_\ell := X_2 \cap \left(\left(\bigcup_{q \in N_\ell} C(q) \right) \cup \left(\bigcup_{\text{Ball} \in D_\ell} \text{Ball} \right) \right)$. That is, \mathcal{X}_ℓ denotes the set of those points in X_2 that belong to the clusters of all the points in N_ℓ , as well as those in the balls in D_ℓ . Now, imagine calling GREEDYCLUSTERING($\mathcal{X}_\ell, X, r_2, 3, \omega_b$). As in the proof of Lemma 9, the main observation is that the set of clusters computed in the first $|N_\ell|$ iterations is exactly $\{C(q)\}_{q \in N_\ell}$. Thus, property B in the lemma follows from Lemma 7 applied to GREEDYCLUSTERING($\mathcal{X}_\ell, X, r_2, 3, \omega_b$). ◀

We now set $N = \bigcup_{1 \leq \ell \leq t} N_\ell$. Note that

$$|N| = \sum_{\ell} |N_\ell| \leq \sum_{\ell} |D_\ell| = |\widehat{\mathcal{B}}_2| \leq |\mathcal{B}_2|.$$

Recall that for instance \mathcal{I}' , we set $\mathcal{B}'_2 = \{B(q, 5r_2) \mid q \in N\}$. We now argue that N satisfies properties (1) and (2).

By Property A of Lemma 15, we have that for any $\text{Ball} \in \mathcal{B}_2$, there is a $\text{Ball}' \in \mathcal{B}'_2$ such that $X_2 \cap \text{Ball} \subseteq X_2 \cap \text{Ball}'$. Thus, $(X_2 \cap \bigcup_{\text{Ball} \in \mathcal{B}_2} \text{Ball}) \subseteq (X_2 \cap \bigcup_{\text{Ball}' \in \mathcal{B}'_2} \text{Ball}')$, which implies property (1).

Using Property B of Lemma 15, we can prove property (1) as follows:

$$\sum_{p \in N} wt(p) = \sum_{\ell} \sum_{p \in N_\ell} wt(p) \geq \sum_{\ell} \omega_b \left(X_2 \cap \bigcup_{\text{Ball} \in D_\ell} \text{Ball} \right) \geq \omega_b \left(X_2 \cap \bigcup_{\text{Ball} \in \mathcal{B}_2} \text{Ball} \right),$$

Phase 2. Phase 1 outputs an instance $\mathcal{I}' = ((X, d), (\lambda_r, \lambda_b, m_r, m_b), (k_1, k_2), (r'_1, r'_2))$ of Colorful 2-NUkC. In Phase 2, we transform this into an instance $\mathcal{I}'' = ((X, d), (\chi_r, \chi_b, m_r, m_b), (k_1, k_2), (r''_1, 0))$ of Colorful 2-NUkC where the radius at the second level is 0.

In this step, we call the greedy clustering algorithm (Algorithm 1) using the *red weight function* λ_r . In particular, we will call $\text{GREEDYCLUSTERING}(X, X, r'_2, 3, \lambda_r)$. This algorithm returns a set of points $M \subseteq X$, where every $p \in M$ has a cluster $C(p)$ and weight $wt(p)$ such that (1) $\{C(p)\}_{p \in M}$ is a partition of X , (2) For any $p \in M$, $wt(p) = \lambda_r(C(p))$, the red weight of the cluster, and (3) $d(q, p) \leq 3r'_2$ for any $q \in C(p)$. Furthermore, the greedy algorithm naturally defines an ordering $\sigma = \langle p_1, p_2, \dots, p_{|M|} \rangle$ of M – this is the order in which the points were added to M .

We define the red weight function χ_r for \mathcal{I}'' as follows: $\chi_r(p) := \lambda_r(C(p))$ for $p \in M$, and $\chi_r(p) := 0$ for $p \in X \setminus M$. We also define $\phi : X \rightarrow M$ as follows: $\phi(p)$ is the first point in M (according to σ) such that $B(p, r'_2) \cap C(p) \neq \emptyset$. Note that $\phi(p)$ exists and $d(p, \phi(p)) \leq 4r'_2$. We define the blue weight function χ_b for \mathcal{I}'' as follows: $\chi_b(p) := \sum_{q \in \phi^{-1}(p)} \lambda_b(q)$ for $p \in M$, and $\chi_b(p) := 0$ for $p \in X \setminus M$.

Finally, we let $r''_1 = r'_1 + 4r'_2$, and obtain the instance $\mathcal{I}'' = ((X, d), (\chi_r, \chi_b, m_r, m_b), (k_1, k_2), (r''_1, 0))$ of Colorful 2-NUkC. The following lemma relates instances \mathcal{I}' and \mathcal{I}'' .

► **Lemma 16.** (a) *If instance \mathcal{I}' has a feasible solution that is structured, then the instance \mathcal{I}'' has a feasible solution.* (b) *Given a solution $(\mathcal{B}'_1, \mathcal{B}'_2)$ for \mathcal{I}' that uses at most k_i balls of radius $\alpha r'_i$ for each $i \in \{1, 2\}$, we can obtain a solution $(\mathcal{B}'_1, \mathcal{B}'_2)$ for \mathcal{I}' that uses at most k_i balls of radius $\alpha r'_i + 4r'_2 \leq \alpha r'_i + (4\alpha + 4)r'_2$ for $i \in \{1, 2\}$.*

Again, part (b) follows from the fact that in constructing \mathcal{I}'' from \mathcal{I}' , we move weights by a distance of at most $4r'_2$. Note that we do not claim that the solution to \mathcal{I}' constructed in part (b) is structured.

In the rest of this section, we establish part (a). Fix a feasible solution $(\mathcal{B}'_1, \mathcal{B}'_2)$ for \mathcal{I}' that is also structured. Our construction of a feasible solution for \mathcal{I}'' is analogous to what we did in Phase 1.

Let $M_1 \subseteq M$ be the set of points p such that there exists some point x satisfying (i) x is covered by a ball in \mathcal{B}'_1 , and (ii) $d(x, p) \leq 4r'_1$. Note that M_1 includes any $p \in M$ such $C(p)$ contains a point covered by a ball in \mathcal{B}'_1 . Now let $M_2 = M \setminus M_1$; note that for $p \in M_2$, any point in $C(p)$ is either covered by a ball from \mathcal{B}'_2 , or is an outlier. Let $X_i := \bigcup_{p \in M_i} C(p)$ for $i = 1, 2$. Note that $X = X_1 \sqcup X_2$.

Let $\widehat{\mathcal{B}}'_2 = \{\text{Ball} \in \mathcal{B}'_2 \mid \text{Ball} \cap X_2 \neq \emptyset\}$. We invoke `MAPPING PROCEDURE`($M_2, \sigma, \widehat{\mathcal{B}}'_2, \{C(p)\}_{p \in M_2}$) and $\mathcal{T} = \{(N_1, D_1), (N_2, D_2), \dots, (N_t, D_t)\}$. We let $N = \bigcup_{1 \leq \ell \leq t} N_\ell$.

As in phase 1, we have that $|N| \leq |\widehat{\mathcal{B}}'_2| \leq |\mathcal{B}'_2|$. The set N also satisfies the following property, which is the analog of Property 2.

$$\sum_{p \in N} wt(p) \geq \lambda_r(X_2 \cap \bigcup_{\text{Ball} \in \mathcal{B}'_2} \text{Ball}) \quad (3)$$

We now construct a solution $(\mathcal{B}''_1, \mathcal{B}''_2)$ for \mathcal{I}'' . The set \mathcal{B}''_1 is obtained by expanding each ball in \mathcal{B}'_1 by an additive $4r'_2$; each ball in \mathcal{B}''_1 has radius r''_1 . Note that by definition of M_1 , the balls in \mathcal{B}''_1 cover M_1 . The set \mathcal{B}''_2 is obtained by including in it a ball of radius 0 at each point in N . Note that $|\mathcal{B}''_2| = |N| \leq |\mathcal{B}'_2|$.

We now argue that $(\mathcal{B}''_1, \mathcal{B}''_2)$ provides adequate coverage. Red coverage is analogous to blue coverage in phase 1, using property 3:

$$\chi_r(M_1) + \chi_r(N) = \lambda_r(X_1) + \sum_{p \in N} wt(p) \geq \lambda_r(X_1) + \lambda_r(X_2 \cap \bigcup_{\text{Ball} \in \mathcal{B}'_2} \text{Ball}) \geq m_r.$$

For blue coverage, let $Y \subseteq X$ denote the set of points *self-covered* by the structured, feasible solution $(\mathcal{B}'_1, \mathcal{B}'_2)$ with $\lambda_b(Y) \geq m_b$. We argue that for each $y \in Y$, we have $\phi(y) \in M_1 \cup N$. If y is covered by a ball in \mathcal{B}'_1 , then as $d(y, \phi(y)) \leq 4r'_2$, we conclude that $\phi(y) \in M_1$ using the definition of M_1 . Otherwise, y is the center of some ball in $B(y, r'_2) \in \mathcal{B}'_2$. Assume $\phi(y) \notin M_1$. Then by the definition of ϕ , $\phi(y)$ is the first point $p \in M_2$ such that $B(y, r'_2)$ intersects $C(p)$. But this means $\phi(y)$ is the same as $\varphi(B(y, r'_2))$ computed in `MAPPING PROCEDURE`($M_2, \sigma, \widehat{\mathcal{B}}'_2, \{C(p)\}_{p \in M_2}$). Thus, $B(y, r'_2) \in D_\ell$ and $\phi(y) \in N_\ell$ for some pair (N_ℓ, D_ℓ) in \mathcal{T} . We conclude $\phi(y) \in N = \bigcup_\ell N_\ell$.

Thus, the blue coverage of $(\mathcal{B}''_1, \mathcal{B}''_2)$ is at least

$$\chi_b(M_1) + \chi_b(N) \geq \sum_{p \in M_1 \cup N} \phi^{-1}(p) \geq \sum_{y \in Y} \lambda_b(y) \geq m_b.$$

This completes the proof of Lemma 16 and concludes our description of Phase 2. Combining Phase 1 and Phase 2, we conclude with the main result of this section.

► **Theorem 17.** *There is a polynomial-time algorithm that transforms a Colorful 2-NUkC instance $\mathcal{I} = ((X, d), (\omega_r, \omega_b, m_r, m_b), (k_1, k_2), (r_1, r_2))$ into an instance $\mathcal{I}'' = ((X, d), (\chi_r, \chi_b, m_r, m_b), (k_1, k_2), (r''_1, 0))$ of Colorful 2-NUkC with $r''_1 = r_1 + 26r_2$, and has the following properties: (a) If \mathcal{I} has a feasible solution, then so does \mathcal{I}'' ; (b) Given an α -approximate solution to \mathcal{I}'' , we can construct, in polynomial time, a $c \cdot \alpha$ -approximate solution to \mathcal{I} , where $c > 0$ is an absolute constant.*

► **Remark 18.** In part (a), the feasible solution $(\mathcal{B}''_1, \mathcal{B}''_2)$ to \mathcal{I}'' that is constructed from feasible solution $(\mathcal{B}_1, \mathcal{B}_2)$ to \mathcal{I} has the following useful property: for any $\text{Ball} \in \mathcal{B}''_1$, the center of Ball is also the center of some ball in \mathcal{B}'_1 .

5 Solving Well-Separated Colorful 2-NUkC

We assume that we are given a *well-separated* instance $\mathcal{I} = ((X, d), (\omega_r, \omega_b, m_r, m_b), (k_1, k_2), (r_1, 0))$ of Colorful 2-NUkC. The *well-separatedness* of the instance comes with the following additional input and restriction – we are given an additional set $Y \subseteq X$ as an input. The set Y is *well-separated*, i.e., for any $u, v \in Y$, $d(u, v) > 2r_1$. The additional restriction is that, the set of centers of balls of radius r_1 must be chosen from the set Y . We sketch how to solve such an instance optimally in polynomial time using dynamic programming.

Let $z := |Y|$, and let $Y = \{y_1, y_2, \dots, y_z\}$. For $1 \leq i \leq z$, let $X_i := B(y_i, r_1) \cap X$, and let $X_{z+1} := X \setminus \left(\bigcup_{1 \leq i \leq z} X_i \right)$. Note that $\{X_i\}_{1 \leq i \leq z+1}$ is a partition of X .

For any $X' \subseteq X$ and non-negative integers k, n_r, n_b , let $F(X', k, n_r, n_b)$ be **true** if there exists a subset $X'' \subseteq X'$ of size at most k , and (red, blue) weight at least (n_r, n_b) ; and **false** otherwise.⁴ For a particular subset X' , the value of $F(X', k, n_r, n_b)$ can be found in polynomial time using dynamic programming, since the values k, n_r, n_b are at most n .

For $(1, 0, 0, 0, 0) \leq (i, k'_1, k'_2, n_r, n_b) \leq (z+1, k_1, k_2, m_r, m_b)$, let $G(i, k'_1, k'_2, n_r, n_b)$ be **true** if it is possible to obtain (red, blue) coverage of at least (n_r, n_b) from the set of points $\bigcup_{1 \leq j \leq i} X_j$, using at most k'_1 balls of radius r_1 and k'_2 balls of radius 0; and **false** otherwise. Note that if $G(i-1, k'_1, k'_2, n_r, n_b) = \mathbf{true}$, then $G(i, k'_1, k'_2, n_r, n_b)$ is trivially **true**. Otherwise, suppose some points in X_i are covered. We consider two possibilities: either (A) X_i is covered using a ball of radius r_1 (note that for $i \leq z$ this is possible by definition; for $i = z+1$ we omit this case), and the remaining (red, blue) coverage comes from $\bigcup_{1 \leq j \leq i-1} X_j$, or (B) We use some $1 \leq t \leq \min\{k'_2, |X_i|\}$ balls of radius 0 to achieve the (red, blue) coverage of (n'_r, n'_b) from within X_i , and the remaining (red, blue) coverage comes from $\bigcup_{1 \leq j \leq i-1} X_j$. Note that in case (B), for a fixed guess of (t, n'_r, n'_b) , the subproblem for X_i corresponds to $F(X_i, t, n'_r, n'_b)$ as defined in the previous paragraph, and can be solved in polynomial time. It is straightforward to convert this recursive argument to compute $G(z+1, k_1, k_2, m_r, m_b)$ into a dynamic programming algorithm that also finds a feasible solution, and it can be implemented in polynomial time. We omit the details.

6 From $(t+1)$ -NUkC to Robust t -NUkC

In this section, we show an approximate equivalence of $t+1$ -NUkC and Robust t -NUkC. Note that Jia et al. [12] recently showed a very similar result. However, our proof is slightly different from theirs, and we describe it here for the sake of completeness.

► Lemma 19.

1. Suppose there exists an α -approximation algorithm for $(t+1)$ -NUkC. Then, there exists an α -approximation algorithm for unweighted Robust t -NUkC.
2. Suppose there exists a β -approximation algorithm for unweighted Robust t -NUkC. Then there exists a $3\beta + 2$ -approximation algorithm for $(t+1)$ -NUkC.

Proof. Note that the first claim is trivial, since an instance of Robust t -NUkC is a special case of NUkC, as follows. Let $\mathcal{I} = ((X, d), (\mathbb{1}, m), (r_1, r_2, \dots, r_t), (k_1, k_2, \dots, k_t))$ be an instance of unweighted t -Robust-NUkC, where m is the coverage requirement. Then, observe that it is equivalent to the instance $\mathcal{I}' = ((X, d), (r_1, r_2, \dots, r_t, 0), (k_1, k_2, \dots, k_t, n-m))$ of $t+1$ -NUkC. An α -approximate solution to \mathcal{I}' immediately gives an α -approximate solution to \mathcal{I} . We now proceed to the second claim.

Consider an instance $\mathcal{I} = ((X, d), (r_1, r_2, \dots, r_t, r_{t+1}), (k_1, k_2, \dots, k_t, k_{t+1}))$ of $(t+1)$ -NUkC. Note that we have to cover all points of X in the instance \mathcal{I} . First, we compute a $2r_{t+1}$ -net Y of X . That is compute $Y \subseteq X$ with the following properties: (i) $d(u, v) > 2r_{t+1}$ for any $u, v \in Y$, and (ii) for any $u \in X \setminus Y$, there exists a $v \in Y$ such that $d(u, v) \leq 2r_{t+1}$. Let $\varphi : X \rightarrow Y$ be a mapping that assigns every point in X to its nearest point in Y (breaking ties arbitrarily). Our reduction constructs the instance $\mathcal{I}' = ((Y, d), (\mathbb{1}, |Y| - k_{t+1}), (k_1, k_2, \dots, k_t), (r'_1, r'_2, \dots, r'_t))$ of t -Robust-NUkC with at most k_{t+1} outliers, where $r'_i = r_i + 2r_{t+1}$ for $1 \leq i \leq t$.

⁴ We use X'' has (red, blue) weight at least (n_r, n_b) as shorthand for $\omega_r(X'') \geq n_r$ and $\omega_b(X'') \geq n_b$.

We now argue that if \mathcal{I} is feasible, then so is \mathcal{I}' . Fix a solution $(\mathcal{B}_i)_{i \in [t+1]}$ for the original instance \mathcal{I} , where \mathcal{B}_i is a set of at most k_i balls of radius r_i . Let $Y' \subseteq Y$ be the set of points in Y covered by $(\mathcal{B}_i)_{i \in [t]}$, the balls of the t largest radii types. For each ball $B(c_i, r_i) \in \mathcal{B}_i$, we add $B(\varphi(c_i), r'_i)$ to obtain the set \mathcal{B}'_i of balls; recall $r'_i = r_i + 2r_t$. Note that the resulting solution $(\mathcal{B}'_i)_{i \in [t]}$ covers the set of points Y' . Now, let $Y'' = Y \setminus Y'$ be the set of points covered by \mathcal{B}_{t+1} , the balls of radius r_{t+1} . The distance between any two points of Y , and thus Y'' , is greater than $2r_{t+1}$. Therefore, a ball of radius r_{t+1} covers at most one point of Y'' , which implies that $|Y''| \leq |\mathcal{B}_{t+1}| \leq k_{t+1}$. Thus $(\mathcal{B}'_i)_{i \in [t]}$ is a feasible solution for instance \mathcal{I}' , with the points in Y'' being the set of outliers of size at most k_{t+1} .

We now argue that from a β -approximate solution to \mathcal{I}' , we can efficiently construct a $(3\beta + 2)$ -approximate solution to \mathcal{I} . Fix a solution $(\mathcal{B}'_i)_{i \in [t]}$ for the instance \mathcal{I}' that covers at least $|Y| - k_{t+1}$ points of Y , where \mathcal{B}'_i consists of k_i balls of radius $\beta r'_i$, for $1 \leq i \leq t$. To obtain a solution for the original instance \mathcal{I} , we proceed as follows. We expand the radius of every ball in \mathcal{B}'_i by an additive factor of $2r_{t+1}$ to obtain \mathcal{B}_i . Note that the resulting radius for each ball in \mathcal{B}_i is $\beta r_i + 2\beta r_{t+1} + 2r_{t+1} \leq (3\beta + 2) \cdot r_i$. Note that if a ball in solution $(\mathcal{B}'_i)_{i \in [t]}$ covers $y \in Y$, then the additively expanded version of the ball covers every point $x \in \varphi^{-1}(y)$. For every outlier point $y \in Y$ not covered by $(\mathcal{B}'_i)_{i \in [t]}$, we add a ball of radius $2r_{t+1}$ centered at y to \mathcal{B}_{t+1} ; this ball covers all points $x \in \varphi^{-1}(y)$. As the number of outliers is at most k_{t+1} , we have $|\mathcal{B}_{t+1}| \leq k_{t+1}$. The resulting solution $(\mathcal{B}_i)_{i \in [t+1]}$ covers all the points of X , and has approximation guarantee $3\beta + 2$. \blacktriangleleft

References

- 1 Georg Anegg, Haris Angelidakis, Adam Kurpisz, and Rico Zenklusen. A technique for obtaining true approximations for k -center with covering constraints. In Daniel Bienstock and Giacomo Zambelli, editors, *Integer Programming and Combinatorial Optimization - 21st International Conference, IPCO 2020, London, UK, June 8-10, 2020, Proceedings*, volume 12125 of *Lecture Notes in Computer Science*, pages 52–65. Springer, 2020. doi:10.1007/978-3-030-45771-6_5.
- 2 Sayan Bandyapadhyay. On perturbation resilience of non-uniform k -center. In Jaroslaw Byrka and Raghu Meka, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2020, August 17-19, 2020, Virtual Conference*, volume 176 of *LIPICs*, pages 31:1–31:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.APPROX/RANDOM.2020.31.
- 3 Sayan Bandyapadhyay, Tanmay Inamdar, Shreyas Pai, and Kasturi R. Varadarajan. A constant approximation for colorful k -center. In Michael A. Bender, Ola Svensson, and Grzegorz Herman, editors, *27th Annual European Symposium on Algorithms, ESA 2019, September 9-11, 2019, Munich/Garching, Germany*, volume 144 of *LIPICs*, pages 12:1–12:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPICs.ESA.2019.12.
- 4 Deeparnab Chakrabarty, Prachi Goyal, and Ravishankar Krishnaswamy. The non-uniform k -center problem. *ACM Trans. Algorithms*, 16(4):46:1–46:19, 2020. doi:10.1145/3392720.
- 5 Deeparnab Chakrabarty and Maryam Negahbani. Generalized center problems with outliers. In *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.
- 6 Deeparnab Chakrabarty and Maryam Negahbani. Robust k -center with two types of radii. In Mohit Singh and David P. Williamson, editors, *Integer Programming and Combinatorial Optimization - 22nd International Conference, IPCO 2021, Atlanta, GA, USA, May 19-21, 2021, Proceedings*, volume 12707 of *Lecture Notes in Computer Science*, pages 268–282. Springer, 2021. doi:10.1007/978-3-030-73879-2_19.
- 7 Moses Charikar, Samir Khuller, David M Mount, and Giri Narasimhan. Algorithms for facility location problems with outliers. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 642–651. Society for Industrial and Applied Mathematics, 2001.

- 8 David G Harris, Thomas Pensyl, Aravind Srinivasan, and Khoa Trinh. A lottery model for center-type problems with outliers. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.
- 9 Dorit S Hochbaum and Wolfgang Maass. Approximation schemes for covering and packing problems in image processing and vlsi. *Journal of the ACM (JACM)*, 32(1):130–136, 1985.
- 10 Dorit S Hochbaum and David B Shmoys. A best possible heuristic for the k-center problem. *Mathematics of operations research*, 10(2):180–184, 1985.
- 11 Tanmay Inamdar and Kasturi Varadarajan. Capacitated sum-of-radii clustering: An fpt approximation. In *28th Annual European Symposium on Algorithms (ESA 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.
- 12 Xinrui Jia, Lars Rohwedder, Kshiteej Sheth, and Ola Svensson. Towards non-uniform k -center with constant types of radii. In Karl Bringmann and Timothy Chan, editors, *5th Symposium on Simplicity in Algorithms, SOSA@SODA 2022, Virtual Conference, January 10-11, 2022*, pages 228–237. SIAM, 2022. doi:10.1137/1.9781611977066.16.
- 13 Xinrui Jia, Kshiteej Sheth, and Ola Svensson. Fair colorful k -center clustering. In Daniel Bienstock and Giacomo Zambelli, editors, *Integer Programming and Combinatorial Optimization - 21st International Conference, IPCO 2020, London, UK, June 8-10, 2020, Proceedings*, volume 12125 of *Lecture Notes in Computer Science*, pages 209–222. Springer, 2020. doi:10.1007/978-3-030-45771-6_17.

A Setup for Robust t -NU k C

Let $\mathcal{I} = ((X, d), (\mathbb{1}, m)(k_1, \dots, k_t), (r_1, \dots, r_t))$ be an instance of Robust t -NU k C. First we state the natural LP relaxation for \mathcal{I} . Recall that the goal is to cover at least m points.

$$\begin{aligned}
\sum_{v \in X} \text{cov}(v) &\geq m \\
\sum_{u \in X} x_{i,u} &\leq k_i && \forall 1 \leq i \leq t \\
\text{cov}_i(v) &= \sum_{u \in B(v, r_i)} x_{i,u} && \forall 1 \leq i \leq t, \forall v \in X \\
\text{cov}(v) &= \min \left\{ \sum_{i=1}^t \text{cov}_i(v), 1 \right\} && \forall v \in X \\
x_{i,u} &\geq 0 && \forall 1 \leq i \leq t, \forall u \in X.
\end{aligned}$$

Let \mathcal{F} denote the set of all tuples of subsets (S_1, \dots, S_t) , where $|S_i| \leq k_i$ for $1 \leq i \leq t$. For $v \in X$, and $1 \leq i \leq t$, we say that $(S_1, \dots, S_t) \in \mathcal{F}$ covers v with radius r_i , if $d(v, S_i) \leq r_i$. Let $\mathcal{F}_i(v) \subseteq \mathcal{F}$ denote the subset of solutions that cover v with radius r_i – where, the sets $\mathcal{F}_i(v)$ of solutions are assumed to be disjoint by including a solution in $\mathcal{F}_i(v)$ of the smallest index i , if it appears in multiple such sets.

If the instance \mathcal{I} is feasible, then the integer hull of the coverages, $\mathcal{P}_{\text{cov}}^{\mathcal{I}}$ as given below, must be non-empty.

$$\begin{aligned}
\mathcal{P}_{\text{cov}}^{\mathcal{I}} : \quad &\sum_{v \in X} \sum_{i \in [t]} \text{cov}_i(v) \geq m \\
&\sum_{S \in \mathcal{F}_i(v)} z_S = \text{cov}_i(v) && \forall i \in [t], \forall v \in X \\
&\sum_{S \in \mathcal{F}} z_S = 1 \\
&z_S \geq 0 && \forall S \in \mathcal{F}
\end{aligned}$$

Next, we give a few definitions from [6], generalized to arbitrary $t \geq 2$, for the sake of completeness. These definitions are used in the round-or-cut framework that reduces an instance of Robust t -NUkC to Well-Separated Robust t -NUkC, as described in Section B.

t -Firefighter Problem. The input is a collection of height- t trees, where L_1 is the set of roots, and for any $v \in L_i$ with $i \geq 1$, $a_j(v)$ represents the ancestor of v that belongs to L_j , where $1 \leq j \leq i$ ($a_i(v) = v$). Furthermore, let $w : L_t \rightarrow \mathbb{N}$ be a weight function on the leaves. For a root $u \in L_1$, we use $\text{Leaf}(u)$ to denote the set of leaves, i.e., nodes in L_t in the tree rooted at u . Note that the $\{\text{Leaf}(u) : u \in L_1\}$ partitions L_t . Thus, $((L_1, \dots, L_t), (a_1, a_2, \dots, a_t), \text{Leaf}, w)$ completely describes the structure of the tree, where $a_i(v) : \bigcup_{i \leq j \leq t} L_j \rightarrow L_i$ is an ancestor function as defined above. Now we define the t -FF problem.

► **Definition 20** (t -FF Problem). *Given height- t trees $(\mathcal{T} = (L_1, \dots, L_t), (a_1, \dots, a_t), \text{Leaf}, w)$, along with budgets (k_1, \dots, k_t) , we say that $T = (T_1, \dots, T_t)$, with $T_i \subseteq L_i$ is a feasible solution, if $|T_i| \leq k_i$ for $1 \leq i \leq t$. Let $\mathcal{C}(T) = \{v \in L_t : a_i(v) \in T_i \text{ for some } 1 \leq i \leq t\}$ be the set of leaves covered by the solution. Then, the objective is to find a feasible solution maximizing the weight of the leaves covered. This instance is represented as $\mathcal{I} = (\mathcal{T} = ((L_1, \dots, L_t), (a_1, \dots, a_t), \text{Leaf}, w), (k_1, \dots, k_t))$,*

► **Definition 21** (The solution y). *Given cov , and a collection \mathcal{T} of rooted trees, let L_1 denote the set of roots, and let L_i , $i > 1$ denote the set of vertices at j -th level. Furthermore, for any node $v \in L_i$ with $i > 1$, let $a_j(v)$ denote the ancestor of v that belongs to L_j , where $1 \leq j < i$. Then, the solution y is defined as follows.*

$$y(v) = \begin{cases} \text{cov}_1(v) & \text{if } v \in L_1 \\ \min \left\{ \text{cov}_i(v), 1 - \sum_{j < i} \text{cov}_j(a_j(v)) \right\} & \text{if } v \in L_i, i > 1 \end{cases}$$

► **Definition 22** (The Sparse LP).

$$\begin{aligned} \max \quad & \sum_{v \in L_t} w(v)Y(v) \\ & \sum_{u \in L_1} y_u \leq k_1 - t \\ & \sum_{u \in L_i} y_u \leq k_i \quad \forall 2 \leq i \leq t \\ & Y(v) := y_v + \sum_{i=1}^{t-1} y_{a_i(v)} \quad \forall v \in L_t \end{aligned}$$

We now describe two subroutines that are used in the Reduction from Robust t -NUkC to Well-Separated Robust t -NUkC. We use the same notation and convention as in [6]. These two algorithms (Algorithm 3 and Algorithm 4) are named after Hochbaum, and Shmoys [10]; and Chakrabarty, Goyal, and Krishnaswamy [4], respectively.

We construct the t -FF instance based on the sets L_i 's constructed, as follows. Consider some $1 \leq i \leq t-1$, and some $u \in L_i$. Then, for every $v \in \text{Child}_i(u)$, we make v a child of u in a tree T . Note that L_1 is the set of roots of the trees constructed in this way. Then, we define $\text{Leaf}(u) = \{v \in L_t : v \text{ is a leaf in the tree rooted at } v\}$, and let $a_i : L_t \rightarrow \bigcup_{i \leq j \leq t} L_j$ be the ancestor function as defined above. Finally, for every $u \in L_t$, let $w(u) = |\text{Child}_t(u)|$. Then, we return the t -FF instance $\mathcal{I} = (\mathcal{T} = ((L_1, \dots, L_t), (a_1, \dots, a_t), \text{Leaf}, w), (k_1, \dots, k_t))$.

■ **Algorithm 3** HS(Metric space (X, d) , $r \geq 0$, assignment $\text{cov} : X \rightarrow \mathbb{R}^+$).

```

1:  $R \leftarrow 0$ 
2: while  $U \neq \emptyset$  do
3:    $u \leftarrow \arg \max_{v \in U} \text{cov}(v)$ 
4:    $R \leftarrow R \cup \{u\}$ 
5:    $\text{Child}(u) \leftarrow \{v \in U : d(u, v) \leq r\}$ 
6:    $U \leftarrow U \setminus \text{Child}(u)$ 
7: end while
8: return  $R, \{\text{Child}(u) : u \in R\}$ .
```

■ **Algorithm 4** CGK.

Input: Robust t -NUkC instance $\mathcal{I} = ((X, d), (\omega, m), (r_1, \dots, r_t), (k_1, \dots, k_t)), (\alpha_1, \dots, \alpha_t)$, where $\alpha_i > 0$ for $1 \leq i \leq t$,
 $\text{cov} = (\text{cov}_1, \dots, \text{cov}_t)$, where each $\text{cov}_i : X \rightarrow \mathbb{R}^+$

```

1: for  $i = t$  downto 1 do
2:    $(L_i, \{\text{Child}_i(v) : v \in L_i\}) \leftarrow \text{HS}((X, d), \alpha_i r_i, \text{cov}'_i := \sum_{j=1}^i \text{cov}_j)$ 
3: end for
4: Construct and Return a  $t$ -FF instance using  $\{L_i, \text{Child}_i\}_{1 \leq i \leq t}$  as described below.
```

B From Robust t -NUkC to Well-Separated Robust t -NUkC

In this section, we use the round-or-cut framework of [6] to give a Turing reduction from Robust t -NUkC to (polynomially many instances of) *Well-Separated* Robust t -NUkC. Furthermore, c -approximation for a feasible instance of the latter problem will imply an $O(c)$ -approximation for the original instance of Robust t -NUkC.

Round-or-Cut Framework. Let $\mathcal{I} = ((X, d), (\mathbb{1}, m), (k_1, k_2, \dots, k_t), (r_1, r_2, \dots, r_t))$ be the given instance of Robust t -NUkC (we assume that we are working with unit-weight instance, where we want to cover at least m points of X). We adopt the round-or-cut framework of [6] (also [5]) to separate an LP solution from the integer hull of coverages (see Section A in the appendix for the definitions thereof). Even though [6] discuss this for $t = 2$, it easily generalizes to arbitrary $t \geq 2$. Thus, we only sketch the high level idea.

Let $\text{cov} = (\text{cov}_1, \text{cov}_2, \dots, \text{cov}_t : \forall v \in X)$ be a candidate solution returned by the ellipsoid algorithm. First, we check whether $\text{cov}(X) \geq m$, and report as the separating hyperplane if this does not hold. Now, we call CGK Algorithm (see Section A) with $\alpha_1 = 6$, and $\alpha_i = 2$ for all $2 \leq i \leq t$ to get a t -FF instance $(\mathcal{T} = ((L_1, \dots, L_t), (a_1, \dots, a_t), \text{Leaf}, w), (k_1, \dots, k_t))$. Here, for any $i \in [t]$, any distinct $p, q \in L_i$ satisfy that $d(p, q) > 3r_i$. Then, we let $\{y_v : v \in \bigcup_i L_i\}$ be the solution as defined in Section A, see Definition 21. Now we check if $\text{cov}_i(L_i) \leq k_i$ for $i \in [t]$, and report if any of these t inequalities is not satisfied. Finally, the algorithm checks the value of $y(L_1)$, and branches into the following two cases.

In the first case, if $y(L_1) \leq k_1 - t$, then as argued by [6], it can be shown that a sparse LP that is related to the t -FF problem (see Definitions 20 and 22) admits an *almost-integral* solution. That is, a basic feasible solution to the sparse LP contains at most t strictly fractional variables. By rounding up all such variables to 1, one can obtain an $O(1)$ -approximation for the original instance \mathcal{I} . Note that here we need the assumption that the ratio between the values of consecutive radii is at least β – otherwise we can merge the two consecutive radii classes into a single class.

In the second case, $y(L_1) > k_1 - t$. In this case, we use a generalization of an argument from [6] as follows. We enumerate every subset $Q \subseteq X$ of size at most $t - 1$, and add a ball of radius r_1 around each point in Q . Let X' be the set of points covered by balls of radius r_1 around Q . Then, we modify the weight of the points of X' to be 0, and let $\mathbb{1}_{X \setminus X'}$ be the resulting weight function. Let $\mathcal{I}(Q) = ((X, d), (\mathbb{1}_{X \setminus X'}, m - |X'|), (2r_1, r_2, \dots, r_t), (k_1 - |Q|, k_2, \dots, k_t))$ be the resulting residual instance of *Well-Separated t -NUkC*, where the *well-separatedness* property imposes that the $2r_1$ centers must be chosen from $Y := L_1 \setminus Q$ – note that the distance between any two distinct points in L_1 , and thus Y , is at least $6r_1 = 3 \cdot 2r_1$, i.e., the set Y is well-separated w.r.t. the new radius r_1 . An argument from [6] implies that if \mathcal{I} is feasible, then either (a) at least one of the well-separated instances $\mathcal{I}(Q)$ is feasible for some $Q \subseteq X$ of size at most $t - 1$, or (b) the hyperplane $y(L_1) \leq k_1 - t$ separates the LP solution cov from the integer hull of coverages. Furthermore, an argument from [6] implies that a constant approximation to any of the instances implies a constant approximation to \mathcal{I} .

Note that the ellipsoid algorithm terminates in polynomially many iterations, and each iteration produces at most n^t instances of *Well-Separated Robust t -NUkC*. Thus, we get the following theorem.

► **Theorem 23.** *Suppose there exists an algorithm that, given an instance \mathcal{J} of Well-Separated Robust t -NUkC, in time $f(n, t)$, either finds an α -approximation to \mathcal{J} , or correctly determines that \mathcal{J} is not feasible. Then, there exists an algorithm to obtain an $c \cdot \alpha$ -approximation for any instance of Robust t -NUkC, running in time $n^{O(t)} \cdot f(n, t)$.*