Improved Polynomial-Time Approximations for Clustering with Minimum Sum of Radii or Diameters

Zachary Friggstad ✉
Department of Computing Science, University of Alberta, Edmonton, Canada

Mahya Jamshidian ✉
Department of Computing Science, University of Alberta, Edmonton, Canada

Abstract

We give an improved approximation algorithm for two related clustering problems. In the Minimum Sum of Radii clustering problem (MSR), we are to select k balls in a metric space to cover all points while minimizing the sum of the radii of these balls. In the Minimum Sum of Diameters clustering problem (MSD), we are to simply partition the points of a metric space into k parts while minimizing the sum of the diameters of these parts. We present a 3.389-approximation for MSR and a 6.546-approximation for MSD, improving over their respective 3.504 and 7.008 approximations developed by Charikar and Panigrahy (2001). In particular, our guarantee for MSD is better than twice our guarantee for MSR.

Our approach refines a so-called bipoint rounding procedure of Charikar and Panigrahy's algorithm by considering centering balls at some points that were not necessarily centers in the bipoint solution. This added versatility enables the analysis of our improved approximation guarantees. We also provide an alternative approach to finding the bipoint solution using a straightforward LP rounding procedure rather than a primal-dual algorithm.

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

Keywords and phrases Approximation Algorithms, Clustering, Linear Programming

Digital Object Identifier 10.4230/LIPIcs.ESA.2022.56

Funding *Zachary Friggstad*: Supported by an NSERC Discovery Grant and Discovery Accelerator Supplement.

Acknowledgements The first author would like to thank Mohammad R. Salavatipour for preliminary discussions on this problem.

1 Introduction

Clustering, as one of the fundamental problems in information technology, has been studied in computing science and several other fields to a great extent. Different methods of clustering have been used significantly in data mining, bioinformatics, pattern recognition, computer vision, etc. The goal of clustering is to partition a set of data points into partitions, called clusters. Many of clustering problems involve finding k cluster centers and a mapping σ from data points to the centers to minimize some objective function. One of the most studied such objective functions is k-Center which minimizes the maximum diameter (or radius) [9, 18]. Another examples is the k-Median problem which aims to minimize sum of distances from data points to their centers, as extensively studied in [5, 6, 19, 20, 21].

In this paper, we focus on a different objective function for clustering that is more center-focused in that the cost of a cluster is the radius of the ball used to cover that cluster. Specifically, we study the following problem.

30th Annual European Symposium on Algorithms (ESA 2022).
Editors: Shiri Chechik, Gonzalo Navarro, Eva Rotenberg, and Grzegorz Herman; Article No. 56; pp. 56:1–56:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

▶ **Definition 1.** *In the* Minimum Sum of Radii *problem (MSR), we are given a set X of n points in a metric space with distances d and a positive integer k. We are to select centers $C \subseteq X$, $|C| \le k$ and assign each $i \in C$ a* radius *r_i so that each $j \in X$ lies within distance r_i of at least one $i \in C$ (i.e. $d(j,i) \le r_i$). The goal is to minimize the total radii, i.e. $\sum_{i \in C} r_i$.*

We also consider the related problem to minimize sum of diameters of the clusters chosen. Note this variant is simply about partitioning the point set, there are no centers involved.

▶ **Definition 2.** *In the* Minimums Sum of Diameters *problem (MSD), the input is the same as in MSR and our goal is to partition the points into k clusters $X_1, X_2, \ldots, X_k$ to minimize $\sum_{i=1}^{k} \max_{j,j' \in X_i} d(j,j')$, the sum of the diameters of the clusters.*

It is easy to see that an α-approximation algorithm for MSR yields a 2α-approximation algorithm for MSD. That is, if OPT_R denotes the optimum MSR solution cost and OPT_D an optimum MSD solution cost, we have $OPT_R \le OPT_D$ because in the optimum MSD solution we could pick any point from each cluster to act as its center (with radius equal to the diameter of the cluster). So if we have an MSR solution with cost at most $\alpha \cdot OPT_R$, then if we define clusters X_i by sending each point to some center whose ball covers that point, the diameter of cluster i would be $\le 2 \cdot r_i$ so the sum of diameters would then be at most $2\alpha \cdot OPT_R \le 2\alpha \cdot OPT_D$.

1.1 Our Contributions

Prior to this work, Charikar and Panigrahy presented a 3.504-approximation for MSR [7]. Since an α-approximation for MSR yields a 2α-approximation for MSD, this also yields a 7.008-approximation for MSD. These were the best polynomial-time approximations for these problems in general metrics.

In this paper, we first present an improved polynomial-time approximation algorithm for MSR. Specifically, we prove the following.

▶ **Theorem 3.** *There is a polynomial-time 3.389-approximation for MSR.*

We obtain this primarily by refining a so-called bipoint rounding step from [7]. That is, our improvement for MSR mainly focuses in the last phase of the algorithm in [7] which combines two subsets of balls that, together, open an *average* of k centers and whose average cost is low. Their algorithm focuses on selecting k of the centers from these two subsets. We expand the set of possible centers to choose and consider some that may not be centers in the averaging of the two subsets.

While not our main result, we also present an alternative way to obtain these two subsets of balls in that we consider a straightforward rounding of a linear programming (LP) relaxation, the Lagrangian relaxation of the problem obtained by relaxing the constraint that at most k centers are chosen, rather than a primal-dual technique as in [7]. Our rounding algorithm is incredibly simple and we employ fairly generic arguments to convert it to a bipoint solution for a single Lagrangian multiplier λ. This may be of independent interest as it should be easy to adapt to other settings where one wants to get a bipoint solution where both points are obtained from a common Lagrangian value λ, as long as the LMP approximation is from direct LP rounding. We emphasize this is only an alternative approach: we could work directly with their primal-dual approach.

Our second result is an improved MSD approximation that does not just use our MSR approximation as a black box.

▶ **Theorem 4.** *There is a polynomial-time 6.546-approximation for MSD.*

In particular, notice the guarantee is better than twice our approximation guarantee for MSR. This is obtained through a variation of our new ideas behind our MSR approximation.

We emphasize this is the first improvement to the approximation guarantee from polynomial-time algorithms for these problems in over 20 years.

1.2 Related Work

Gibson et al. show MSR is **NP**-hard even in metrics with constant doubling dimension or shortest-path metrics of edge-weighted planar graphs [10]. In polynomial time, the best approximation algorithm is the stated 3.504 approximation by Charikar and Panigrahy [7]. Interestingly, [10] show that MSR can be solved exactly in $n^{O(\log n \cdot \log \Gamma)}$ where Γ is the *aspect ratio* of the metric (maximum distance divided by minimum nonzero distance). By standard techniques, this yields a quasi-PTAS for MSR: i.e. a $(1+\epsilon)$-approximation with running time $n^{O(\log 1/\epsilon + \log^2 n)}$. A major open problem is to design a PTAS for MSR, or perhaps to demonstrate there is no PTAS for MSR under some strong lower bound (eg. the exponential-time hypothesis). For now, it is of interest to get improved constant-factor approximations for MSR. By way of analogy, the *unsplittable flow* problem was known to admit a quasi-PTAS [1, 2] yet improved constant-factor approximations were subsequently produced [15, 13, 14], that is until a PTAS was finally found by Grandoni et al. [12].

On the other hand MSD is hard to approximate: Doddi et al. show that unless $\mathbf{P} = \mathbf{NP}$, there is no $(2-\epsilon)$-approximation for MSD for any $\epsilon > 0$ even if the metric is the shortest path metric of an unweighted graph [8]. Prior to our work, the best approximation for MSD is simply twice the best polynomial-time approximation for MSR, i.e. $2 \cdot 3.504 = 7.008$ using the approximation for MSR from [7].

MSR and MSD have been studied in special cases as well. In constant-dimensional Euclidean metrics, MSR can be solved exactly in polynomial time [11]. This is particularly interesting in light of the fact that MSR is hard in doubling metrics. For MSD in constant-dimensional Euclidean metrics, if k is also regarded as a constant then MSD can be solved exactly [4]. In general metrics with $k = 2$, MSD can be solved exactly by observing that if we are given the diameters of the two clusters, we can use 2SAT to determine if we can place the points in these clusters while respecting the diameters [16]. However, MSD is **NP**-hard for even $k = 3$ as it captures the problem of determining if an unweighted graph can be partitioned into 3 cliques. Finally, if one does not allow balls with radius 0 in the solution, MSR can be solved in polynomial time in shortest path metrics of unweighted graphs [3, 17].

1.3 Organization

Our MSR approximation is given in Section 2. Our algorithm follows the same general structure as the algorithm in [7] so we defer the details behind one significant step (obtaining the "bipoint" solution) and focus on our new ideas. Our MSD approximation is then given in Section 3. Finally, our new approach to obtaining a clean bipoint solution is summarized in Section 4. For the sake of space, many proofs in Section 4 are deferred to the full version. Brief concluding remarks are given in Section 5.

2 Minimum Sum of Radii

2.1 Preliminaries

Throughout, $n = |X|$. We assume $d(i, i') > 0$ for distinct $i, i' \in X$ (i.e. there are no collocated points), clearly this is without loss of generality since we could restrict X to contain only one point from each group of collocated points. A **ball** in X is a set of the form $B(i, r) = \{j \in X : d(i, j) \leq r\}$ for some point $i \in X$ and radius $r \geq 0$. We sometimes also call a pair (i, r) a ball with the understanding it is referring to the set $B(i, r)$. One can view a solution to MSR as being a collection of balls.

In some places in our algorithm, we need to guess balls from the optimal solution or use LP variables corresponding to balls that may appear in the optimal solution: in these steps we only need to consider balls $B(i,r)$ where $r = d(i,j)$ for some $j \in X$ because it is clear that an optimal MSR solution will set each radius as to be the furthest point that is covered by that ball. So there are only $O(|X|^2)$ different balls to consider. We view a solution as a collection $\mathcal{B}$ of pairs $(i,r), i \in X, r \geq 0$ describing the centers and radii of balls. For such a subset, we let $cost(\mathcal{B}) = \sum_{(i,r)\in\mathcal{B}} r$ be the total radii of these balls.

Fix some small constant $\epsilon > 0$ such that $1/\epsilon$ is an integer. Smaller ϵ lead to better guarantees with increased (but still polynomial) running times. Since the bound in the statement of Theorem 3 is just a rounded up version of the actual approximation guarantee, we will ultimately pick ϵ to be small enough to hide it in the approximation guarantee, which is why it does not appear in the statement. We assume $k > 1/\epsilon$, otherwise we can simply use brute force to find the optimum solution in $n^{O(1/\epsilon)}$ time.

Our algorithm for MSR is summarized in Algorithm 1 at the end of this section, though it makes reference to a fundamental subroutine to find our "bipoint" solution that we describe in Section 4. By bipoint, we simply mean two subsets of balls $\mathcal{B}_1, \mathcal{B}_2$ with $|\mathcal{B}_1| \geq k \geq |\mathcal{B}_2|$ so, in a sense to be described later, some averaging of these sets looks like a feasible fractional solution using exactly k balls.

2.2 Step 1: Guessing the Largest Balls

Let $\mathcal{B}^*$ denote some fixed optimum solution with $OPT := cost(\mathcal{B}^*)$. Among all optimal solutions, we assume $\mathcal{B}^*$ has the fewest balls. Thus, for distinct $(i,r),(i',r') \in \mathcal{B}^*$ we have that $i' \notin B(i,r)$ since, otherwise, $\mathcal{B}^* - \{(i,r),(i',r')\}) \cup \{(i, r+r')\}$ is another optimal solution with even fewer balls.

Similar to [7], we guess the $1/\epsilon$ largest balls in $\mathcal{B}^*$ by trying each subset $\mathcal{B}'$ of $1/\epsilon$ balls and proceeding with the algorithm we describe in the rest of this paper. Let R_m be the minimum radius of a ball in $\mathcal{B}'$ and note $R_m \leq \epsilon \cdot OPT$. We also let $k' := k - 1/\epsilon$, which is an upper bound on the number of balls in $\mathcal{B}^* - \mathcal{B}'$.

We now restrict ourselves to the instance with points $X' := X - \cup_{(i,r)\in\mathcal{B}'} B(i,r)$ to be covered. Since no center of a ball in $\mathcal{B}^*$ is contained within another ball from $\mathcal{B}^*$, the remaining balls in $\mathcal{B}^* - \mathcal{B}'$ are also centered in X'. We will let $OPT' = OPT - \cup_{(i,r)\in\mathcal{B}'} r$ denote the optimal solution value to this restricted instance. The solution $\mathcal{B}^* - \mathcal{B}'$ for this instance satisfies $r \leq R_m \leq \epsilon \cdot OPT$ for any $(i,r) \in \mathcal{B}^* - \mathcal{B}'$. We also assume $|X'| > k'$, otherwise we just open zero-radius ball at each point in X'.

Before proceeding to the main part of the algorithm, we perform a "precheck" for this guess as follows: run a standard 2-approximation for the k'-MEDIAN instance on the metric restricted to X' (eg. [18]). If the solution returned has radius $> 2 \cdot R_m$, then we reject this guess $\mathcal{B}'$. This is valid because we know for a correct guess that the remaining points can each be covered using at most k' balls each with with radius at most R_m. From now on, we let $\mathcal{A}$ denote the k' centers returned by this approximation: so each $j \in X'$ lies in at least one ball of the form $B(i, 2 \cdot R_m)$ for some $i \in \mathcal{A}$.

Summary. After guessing $\mathcal{B}'$ we have restricted ourselves to an MSR instance with points X', a bound k' on the number of balls to choose (where $k' < |X'|$), and a bound R_m. For a correct guess of $\mathcal{B}'$, there is an optimal solution that uses balls with radius at most R_m. Furthermore, $\mathcal{A}$ is a set of k' centers such that every $j \in X'$ has $d(j, \mathcal{A}) \leq 2 \cdot R_m$ balls.

When analyzing the rest of the algorithm, will assume that $\mathcal{B}'$ is guessed correctly, i.e. $\mathcal{B}' \subseteq \mathcal{B}^*$ and all $(i,r) \in \mathcal{B}^* - \mathcal{B}'$ have $r \leq R_m$. Our final solution will be the minimum-cost solution found over all guesses $\mathcal{B}'$ that were not rejected, so it will be at most the cost of the solution found when $\mathcal{B}'$ was guessed correctly.

2.3 Step 2: Getting a Bipoint Solution

The output from this step is similar to [7], except we obtain it with a different algorithm. We note that their approach would suffice for our purposes, our reasons for considering this different approach are described after the statement of Theorem 6 below. Some details are deferred to Section 4 and some to the full version of this paper, here we explain what is required to understand our ideas that lead to the improved approximation guarantee.

For a value $\lambda \geq 0$, $\mathbf{LP}(\lambda)$ is the linear program that results by considering the Lagrangian relaxation of MSR. That is, the LP has variables for each possible ball we may add except instead of restricting the number of balls to be at most k', we simply pay λ for each ball.

Note. Terms of the LP that consider pairs (i,r) corresponding to balls with $i \in X'$ and r of the form $d(i,j)$ for some $j \in X'$ but only for those where $r \leq R_m$. Thus, the LP has $O(|X|^2)$ variables.

$$\begin{array}{lrcll} \textbf{min} & \sum_{(i,r)} (r+\lambda) \cdot x_{i,r} & & & \\ \textbf{s.t.} & \sum_{(i,r): j \in B(i,r)} x_{i,r} & \geq & 1 & \forall\, j \in X' \\ & x & \geq & 0 & \end{array} \qquad (\mathbf{LP}(\lambda))$$

The following is standard and follows by considering the natural integer solution corresponding to the balls in $\mathcal{B}^* - \mathcal{B}'$.

▶ **Lemma 5.** *For any $\lambda \geq 0$, let $OPT_{\mathbf{LP}(\lambda)}$ denote the optimum value of $\mathbf{LP}(\lambda)$. Then $OPT_{LP(\lambda)} - \lambda \cdot k' \leq OPT'$.*

We summarize the main properties of a bipoint solution that is required by our algorithm. The proof of the following is the subject of Section 4.

▶ **Theorem 6.** *There is a polynomial-time algorithm that will compute a single value $\lambda \geq 0$ and two sets of balls $\mathcal{B}_1, \mathcal{B}_2$ having respective sizes k_1, k_2 where $k_1 \geq k' \geq k_2$. Furthermore, for every $(i,r) \in \mathcal{B}_1$, there is some $(i',r') \in \mathcal{B}_2$ such that $B(i,r) \cap B(i',r') \neq \emptyset$. Finally, for both $\ell = 1$ and $\ell = 2$ we have the following properties:*

- *for each $(i,r) \in \mathcal{B}_\ell$, we have $r \leq 3 \cdot R_m$,*
- *tripling the radii of each $(i,r) \in \mathcal{B}_\ell$ will cover X', i.e. for each $j \in X'$ there is some $(i,r) \in \mathcal{B}_\ell$ such that $j \in B(i, 3 \cdot r)$, and*
- *$cost(\mathcal{B}_\ell) + \lambda \cdot k_\ell \leq OPT_{\mathbf{LP}(\lambda)}$*

Again, we note that essentially the same result is found in [7], except it is slightly more technical to state since the two sets $\mathcal{B}_1$ and $\mathcal{B}_2$ are obtained through a LMP algorithm applied to different (but very close) values λ_1 and λ_2 which leads to an additional ϵ-loss in the approximation guarantee. Qualitatively speaking, the theorem statement itself is not new and the reader who is not interested in seeing a new technique can skip reading its proof.

As an easy warmup, notice that if $k_1 = k'$ then by Lemma 5 we have

$$cost(\mathcal{B}_1) \leq OPT_{\mathbf{LP}(\lambda)} - \lambda \cdot k' \leq OPT'.$$

In this case, tripling the radii of all balls in $\mathcal{B}_1$ covers all of X' with cost at most $3 \cdot OPT'$. Together with $\mathcal{B}'$, this is a feasible MSR solution with cost at most $3 \cdot OPT$. A similar approximation follows if $k_2 = k$. However, we do not distinguish these case in our full analysis below.

2.4 Step 3: Combining Bipoint Solutions

Let $\lambda, \mathcal{B}_1, \mathcal{B}_2$ be the *bipoint solution* from Theorem 6. For brevity, let $C_1 = cost(\mathcal{B}_1)$ and $C_2 = cost(\mathcal{B}_2)$. Since $k_1 \geq k' \geq k_2$, there are values $a, b \geq 0$ with $a+b = 1$ and $a \cdot k_1 + b \cdot k_2 = k'$. We fix these values throughout this section.

The following shows the *average* cost C_1 and C_2 is bounded by OPT', the first inequality is by the last property listed in Theorem 6 and the second by Lemma 5.

$$a \cdot C_1 + b \cdot C_2 \leq a \cdot (OPT_{\mathbf{LP}(\lambda)} - \lambda \cdot k_1) + b \cdot (OPT_{\mathbf{LP}(\lambda)} - \lambda \cdot k_2) = OPT_{\mathbf{LP}(\lambda)} - \lambda \cdot k \leq OPT' \quad (1)$$

The rest of our algorithm and analysis considers how to convert the two solutions $\mathcal{B}_1, \mathcal{B}_2$ to produce a feasible solution whose value is within a constant-factor of this averaging of C_1, C_2. First, note tripling the radii in all balls in $\mathcal{B}_2$ will produce a feasible solution as $k_2 \leq k'$, but it may be too expensive. So we will consider two different solutions and take the better of the two. The first solution is what we just described: formally it is $\{(i, 3r) : (i, r) \in \mathcal{B}_2\}$, which is a feasible solution with cost $3 \cdot C_2$.

Constructing the second solution is our main deviation from the work in [7]. Intuitively, we want to cover all points by using balls $(i, 3 \cdot r)$ for $(i, r) \in \mathcal{B}_1$. The cheaper of this and the first solution can easily be shown to have cost at most $3 \cdot OPT'$. The problem is that this could open more than k' centers (if $k_1 > k'$). As in [7], we consolidate some of these balls into a single group based on their common intersection with some $(i', r') \in \mathcal{B}_2$. We will select some groups and merge their balls into a single ball so the number of balls is at most k'. Our improved approximation is enabled by considering different ways to cover balls in a group using a single ball, [7] only considered one possible way to cover a group with a single ball.

We now form groups. For each $(i, r) \in \mathcal{B}_2$, we create a group $G_{i,r} \subseteq \mathcal{B}_1$ as follows: for each $(i', r') \in \mathcal{B}_1$, consider any single $(i, r) \in \mathcal{B}_2$ such that $B(i, r) \cap B(i', r') \neq \emptyset$ and add (i', r') to $G_{i,r}$. If multiple $(i, r) \in \mathcal{B}_2$ satisfy this criteria, pick one arbitrarily. Let $\mathcal{G} = \{G_{i,r} : (i, r) \in \mathcal{B}_2 \text{ s.t. } G_{i,r} \neq \emptyset\}$ be the collection of all nonempty groups formed this way, note $\mathcal{G}$ is a partitioning of $\mathcal{B}_1$.

Covering a group with a single ball

From here, the approach in [7] would describe how to merge the balls in a group $G_{i,r} \in \mathcal{G}$ simply by centering a new ball at i, and making its radius sufficiently large to cover all points covered by the tripled balls $B(i', 3r')$ for $(i', r') \in G_{i,r}$. We consider choosing a different center when we consolidate the $\mathcal{B}_1$ balls in a group. In fact, it suffices to simply pick the minimum-radius ball that covers the union of the tripled balls in a group. This ball can be centered at any point in X'.

To analyze this, we describe a few candidate balls and argue that the cheapest of these has cost at most $\frac{11}{8} \cdot r + 3 \cdot cost(G_{i,r})$ for each $G_{i,r} \in \mathcal{G}$. The exact choice of ball we use for the analysis depends on the composition of the group, namely the total and maximum radii of balls in $G_{i,r}$ versus the radius r itself. In [7], the ball they select has cost at most $r + 4 \cdot cost(G_{i,r})$. While our analysis has a higher dependence on r, when considered as an alternative solution to the one that just triples all balls in $\mathcal{B}_2$ we end up with a better overall solution.

For now, fix a single group $G_{i,r} \in \mathcal{G}$. Let R_1 denote r, R_2 be the maximum radius of a ball in $G_{i,r}$ and R_3 be the maximum radius among all other balls in $G_{i,r}$ apart from the one defining R_2. If $G_{i,r}$ has only one ball, then let $R_3 = 0$. That is, $0 \leq R_3 \leq R_2$ but it could be that $R_3 = R_2$, i.e. there could be more than one ball from $G_{i,r}$ with maximum radius. We also let i_1 denote i, i_2 be the center of any particular ball with maximum radius in $G_{i,r}$, and i_3 be any single point in $B(i_1, R_1) \cap B(i_2, R_2)$. There is at least one since each ball in $G_{i,r}$ intersects $B(i, r)$ by construction of the groups.

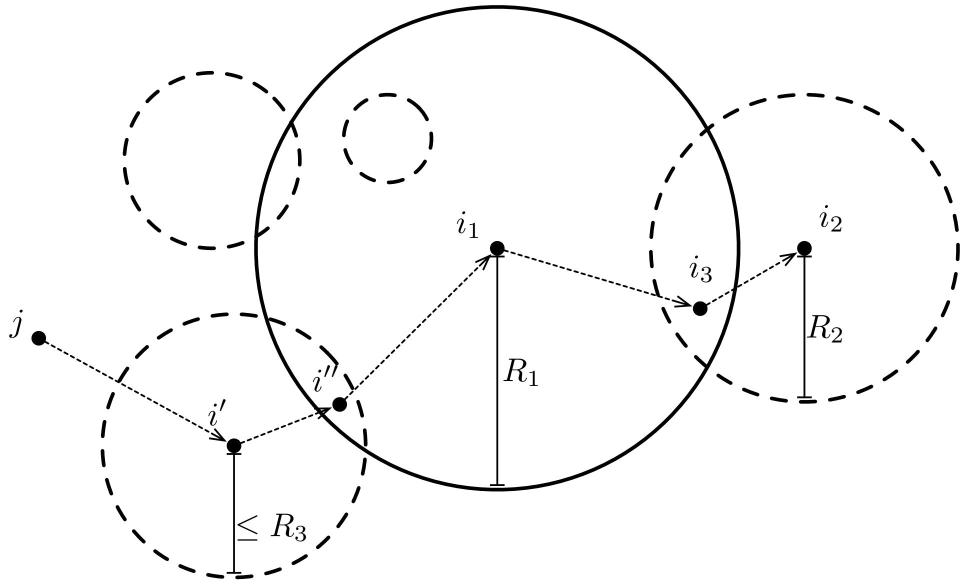

Figure 1 A depiction of a group G_{i_1,R_1}. The solid ball is $B(i_1, R_1)$ and the dashed balls are those in G_{i_1,R_1}. Point j is covered by tripling the ball centered at i'. The dashed path depicts the way we bound $d(j, i_2)$ in the second part of the case **Centering at** i_2.

Next we describe the radius of a ball that would be required if we centered it at one of i_1, i_2 or i_3. Consider any $j \in Y_{i,r}$ with, say, $j \in B(i', 3r')$ for some $(i', r') \in G_{i,r}$. Let i'' be any point in $B(i_1, r) \cap B(i', r')$. We bound the distance of j from each of i_1, i_2 and i_3 to see what radius would suffice for each of these three possible centers. Figure 1 depicts this group and one case of the analysis below.

- **Centering at** i_1. Simply put,

$$\begin{aligned} d(j, i_1) &\leq d(j, i') + d(i', i'') + d(i'', i_1) \\ &\leq 3 \cdot R_2 + R_2 + R_1 \\ &= R_1 + 4 \cdot R_2. \end{aligned}$$

  So radius $C^{(1)} := R_1 + 4 \cdot R_2$ suffices if we choose i_1 as the center.
- **Centering at** i_2. If $(i', r') = (i_2, R_2)$ then $d(j, i_2) \leq 3 \cdot R_2$. Otherwise, $r' \leq R_3$ and

$$\begin{aligned} d(j, i_2) &\leq d(j, i') + d(i', i'') + d(i'', i_1) + d(i_1, i_3) + d(i_3, i_2) \\ &\leq 3 \cdot R_3 + R_3 + R_1 + R_1 + R_2 \\ &= 2 \cdot R_1 + R_2 + 4 \cdot R_3. \end{aligned}$$

  So radius $C^{(2)} := \max\{3 \cdot R_2, 2 \cdot R_1 + R_2 + 4 \cdot R_3\}$ suffices if we choose i_2 as the center.
- **Centering at** i_3. If $(i', r') = (i_2, R_2)$ then $d(j, i_3) \leq d(j, i_2) + d(i_2, i_3) \leq 3 \cdot R_2 + R_2 = 4 \cdot R_2$. Otherwise, $r' \leq R_3$ and we see

$$\begin{aligned} d(j, i_3) &\leq d(j, i') + d(i', i'') + d(i'', i_1) + d(i_1, i_3) \\ &\leq 3 \cdot R_3 + R_3 + R_1 + R_1 \\ &= 2 \cdot R_1 + 4 \cdot R_3. \end{aligned}$$

  So radius $C^{(3)} := \max\{4 \cdot R_2, 2 \cdot R_1 + 4 \cdot R_3\}$ suffices if we choose i_3 as the center.

With these bounds, we now describe how to choose a single ball covering the points covered by tripled balls in $G_{i,r}$ in a way that gives a good bound on the minimum-radius ball covering these points. The following cases employ particular constants to decide which center should be used, these have been optimized for our approach. The final bounds are stated to be of the form $3 \cdot C_{i,r}$ plus some multiple of r. Let $C_{i,r} = \sum_{(i',r') \in G_{i,r}} r$ be the total radii of all balls in $G_{i,r}$. So $\sum_{G_{i,r} \in \mathcal{G}} C_{i,r} = cost(\mathcal{B}_1) = C_1$.

- **Case**: $R_3 > R_2/3$.
  Then the ball $B'_{i,r}$ is selected to be $B(i_1, C^{(1)})$. Note $4/3 \cdot R_2 < R_2 + R_3 \leq C_{i,r}$ so $C^{(1)} \leq r + 3 \cdot C_{i,r}$.
- **Case**: $R_3 \leq R_2/3$ and $R_2 \geq \frac{6}{5} \cdot R_1$.
  The ball $B'_{i,r}$ is selected to be $B(i_2, C^{(2)})$. Note $C^{(2)} \leq \frac{6}{5} \cdot \cdot R_1 + 3 \cdot R_2 \leq \frac{6}{5} \cdot \cdot r + 3 \cdot C_{i,r}$.
- **Case**: $R_3 \leq R_2/3$ and $\frac{6}{5} \cdot R_1 > R_2 \geq \frac{3}{8} \cdot R_1$.
  The ball $B'_{i,r}$ is selected to be $B(i_3, C^{(3)})$. Note $C^{(3)} \leq \frac{11}{8} \cdot R_1 + 3 \cdot R_2 \leq \frac{11}{8} \cdot r + 3 \cdot C_{i,r}$.
- **Case**: $R_3 \leq R_2/3$ and $\frac{3}{8} \cdot R_1 > R_2$.
  The ball $B'_{i,r}$ is selected to be $B(i_1, C^{(1)})$. Note $C^{(1)} \leq \frac{11}{8} \cdot R_1 + 3 \cdot R_2 \leq \frac{11}{8} \cdot r + 3 \cdot C_{i,r}$.

In any case, we see that by selecting $B'_{i,r}$ optimally, the radius is at most $\frac{11}{8} \cdot r + 3 \cdot C_{i,r}$. Also, since $R_1, R_2, R_3 \leq 3 \cdot R_m$ by Theorem 6, then the radius of $B'_{i,r}$ is also seen to be at most, say, $21 \cdot R_m$.

Choosing which groups to merge

For each group $G_{i,r} \in \mathcal{G}$, we consider two options. Either we select all balls in $G_{i,r}$ with triple their original radii (thus, with total cost $3 \cdot C_{i,r}$), or we select the single ball $B'_{i,r}$ described in the previous section. We want to do this to minimize the resulting cost while ensuring the number of centers open is at most k'. To help with this, we consider the following linear program. For each $G_{i,r} \in \mathcal{G}$ we have a variable $z_{i,r}$ where $z_{i,r} = 0$ corresponds to selecting the $|G_{i,r}|$ balls with triple their original radius and $z_{i,r} = 1$ corresponds to selecting the single ball $B'_{i,r}$. As noted in the previous section, the radius of $B'_{i,r}$ is at most $\frac{11}{8} \cdot r + 3 \cdot C_{i,r}$ and also at most $21 \cdot R_m$.

$$\begin{array}{llrcl} \textbf{minimize}: & \sum_{G_{i,r} \in \mathcal{G}} (1 - z_{i,r}) \cdot 3 \cdot C_{i,r} + z_{i,r} \cdot cost(\{B'_{i,r}\}) & & & \\ \textbf{subject to}: & & \sum_{G_{i,r} \in \mathcal{G}} ((1 - z_{i,r}) \cdot |G_{i,r}| + z_{i,r}) & \leq & k' \\ & & z_{i,r} & \in & [0,1] \quad \forall\, G_{i,r} \in \mathcal{G} \end{array} \qquad \textbf{(LP-Choose)}$$

To consolidate the groups, compute an optimal extreme point to **LP-Choose**. Since all but one constraint are $[0, 1]$ box constraints, there is at most one variable $z_{i,r}$ that does not take an integer value. Since $|G_{i,r}| \geq 1$, then setting $z_{i,r}$ to 1 yields a feasible solution whose cost increases by at most the radius of $B'_{i,r}$, which was observed to be at most $21 \cdot R_m \leq 21 \cdot \epsilon \cdot OPT$.

Recall that a, b are such that $a, b \geq 0, a + b = 1$ and $a \cdot k_1 + b \cdot k_2 = k'$. Setting $z_{i,r} = a$ for each $G_{i,r} = 1$ is feasible since $1 - z_{i,r} = b$, $\sum_{G_{i,r} \in \mathcal{G}} |G_{i,r}| = k_2$, and there are at most k'_1 terms in this sum. The value of this solution is

$$\sum_{G_{i,r} \in \mathcal{G}} (3 \cdot b + 3 \cdot a) \cdot C_{i,r} + \frac{11}{8} \cdot b \cdot r = 3 \cdot C_2 + \frac{11}{8} \cdot b \cdot C_1$$

so the optimum solution to **LP-Choose** has value at most this as well. Summarizing,

▶ **Lemma 7.** *In polynomial time, we can compute a set of at most k' balls with total radius at most $\frac{11}{8} \cdot b \cdot C_1' + 3 \cdot C_2 + 21 \cdot \epsilon \cdot OPT$ which cover all points in X'.*

Finally, we can complete our analysis. Recall our simple solution of tripling the balls in $\mathcal{B}_1'$ has cost at most $3 \cdot C_1'$ and the more involved solution jut described has cost at most $3 \cdot C_1 + \frac{11}{8} \cdot a \cdot C_2 + 21 \cdot \epsilon \cdot OPT$. Now,

$$\min\left\{3 \cdot C_2, 3 \cdot C_1 + \frac{11}{8} \cdot b \cdot C_2\right\} \leq (1-d) \cdot 3 \cdot C_2 + d \cdot \left(b \cdot \frac{11}{8} \cdot C_2 + 3 \cdot C_1\right)$$

holds for any $0 \leq d \leq 1$. To maximize the latter, we set $d = \frac{3(1-b)}{\frac{11}{8} \cdot b^2 - \frac{11}{8} \cdot b + 3}$ and see the minimum of these two terms is at most

$$\left(\frac{9}{\frac{11}{8} \cdot b^2 - \frac{11}{8} \cdot b + 3}\right) \cdot (aC_1 + bC_2) \leq \left(\frac{9}{\frac{11}{8} \cdot b^2 - \frac{11}{8} \cdot b + 3}\right) \cdot OPT'$$

where we have used bound 1 for the last step.

The worst case occurs at $b = \frac{1}{2}$, at which the bound becomes $85/288 \cdot OPT'$. Thus, the cost of the solution is at most $\frac{288}{85} \cdot OPT' + 21 \cdot \epsilon \cdot OPT$. Adding the balls $\mathcal{B}'$ we guessed to also cover the points in $X - X'$, we get get a solution covering all of X with total radii at most

$$cost(\mathcal{B}') + \frac{288}{85} \cdot OPT' + 21 \cdot \epsilon = OPT - OPT' + \frac{288}{85} \cdot OPT' + 21 \cdot \epsilon \cdot OPT \leq 3.389 \cdot OPT$$

for sufficiently small ϵ.

Algorithm Summary

The entire algorithm for MSR that we have just presented is summarized in Algorithm 1.

Algorithm 1 MSR Approximation.

$\mathcal{S} \leftarrow \emptyset$ {The set of all solutions seen over all guesses}
for each subset $\mathcal{B}_j'$ of $1/\epsilon$ balls **do**
  let X', R_m be as described in Section 2.2
  $(\mathcal{A}, R) \leftarrow k$-MEDIAN 2-approximation on X'
  if $R > 2 \cdot R_m$ **then**
    reject this guess $\mathcal{B}'$ and continue with the next
  let $\mathcal{B}_1, \mathcal{B}_2, \lambda$ be the bipoint solution described in Theorem 6
  let $\mathcal{G}$ be the groups (a partitioning of $\mathcal{B}_1$) described in Section 2.4
  for each $G_{i,r} \in \mathcal{G}$, let $B_{i,r}'$ be the cheapest ball covering $\cup_{(i',r') \in G_{i,r}} B(i', 3 \cdot r')$
  let z' be an optimal extreme point to **LP-Choose**
  $\mathcal{B}^{(1)} \leftarrow \{B_{i,r}' : z_{i,r}' > 0\} \cup \bigcup_{z_{i,r}'=0} \{(i', 3 \cdot r') : (i', r') \in G_{i,r}\}$
  $\mathcal{B}^{(2)} \leftarrow \{(i, 3 \cdot r) : (i, r) \in \mathcal{B}_2\}$
  let $\mathcal{B}$ be $\{(i, 3 \cdot r) : (i, r) \in \mathcal{B}'\}$ plus the cheaper of the two sets $\mathcal{B}^{(1)}$ and $\mathcal{B}^{(2)}$
  $\mathcal{S} \leftarrow \mathcal{S} \cup \{\mathcal{B}\}$
return the cheapest solution from $\mathcal{S}$

3 Minimum Sum of Diameters

Here, we observe that a slight modification to the MSR approximation in fact yields a 6.546-approximation for MSD. Note that for any $Y \subseteq X$ with diameter, say, $diam(Y)$, for any $i \in Y$ we have $Y \subseteq B(i, diam(Y))$ and $diam(B(i, diam(Y)) \leq 2 \cdot diam(Y)$. So while it is difficult to guess any single cluster from the optimum MSD solution, we can guess the $1/\epsilon$ largest diameters (the values) and guess balls $\mathcal{B}'$ with these radii that cover these largest-diameter clusters. Let OPT'_D denote the total diameter of the remaining clusters from the optimum solution, $k' = k - \frac{1}{\epsilon}$, X' be the remaining points to cluster, and $R_m = \min\{r : (i,r) \in \mathcal{B}'\} \leq \epsilon \cdot OPT_D$.

For any $\lambda \geq 0$, note $OPT_{\mathbf{LP}(\lambda)} + \lambda \cdot k' \leq OPT'_D$ as picking any single center from each cluster in optimum solution on X' yields an MSR solution with cost at most OPT'_D. We then use Theorem 6 to get a bipoint solution $\mathcal{B}_1, \mathcal{B}_2, \lambda$.

If we triple the balls in $\mathcal{B}_2$ and output those clusters, we get a solution with total diameter $\leq 6 \cdot cost(\mathcal{B}_2)$. For the other case, we again form groups $\mathcal{G}$. Instead of picking a ball $B'_{i,r}$ for each group $G_{i,r} \in \mathcal{G}$, we simply let $B'_{i,r}$ be the set of points covered by the tripled balls in $G_{i,r}$. We claim $diam(B'_{i,r}) \leq 2 \cdot r + 6 \cdot C_{i,r}$.

To see this, consider any two points j', j'' covered by $\cup_{(i',r') \in G_{i,r}} B(i', 3 \cdot r')$, say (i', r') and (i'', r'') are the balls in $G_{i,r}$ which, when tripled, cover j' and j'', respectively. If $(i', r') = (i'', r'')$ (i.e. it is the same tripled ball from $G_{i,r}$ that covers both j', j'') then $d(j', j'') \leq 6 \cdot r' \leq 6 \cdot C_{i,r}$. Otherwise, we have $r' + r'' \leq C_{i,r}$ and

$$d(j', j'') \leq d(j', i') + d(i', i) + d(i, i'') + d(i'', j'') \leq 4 \cdot r' + r + r + 4 \cdot r'' \leq 2 \cdot r + 4 \cdot C_{i,r}.$$

In either case, we can upper bound $d(j', j'') \leq 2 \cdot r + 6 \cdot C_{i,r}$, so $diam(B'_{i,r})$ is bounded by the same. We use an LP similar to **LP-Choose** except with the modified objective function to reflect the diameter costs of the corresponding choices.

$$\begin{array}{rrcl}
\textbf{minimize}: & \sum_{G_{i,r} \in \mathcal{G}} (1 - z_{i,r}) \cdot 6 \cdot C_{i,r} + z_{i,r} \cdot diam(B'_{i,r}) & & \\
\textbf{subject to}: & \sum_{G_{i,r} \in \mathcal{G}} ((1 - z_{i,r}) \cdot |G_{i,r}| + z_{i,r}) & \leq & k' \\
 & z_{i,r} & \in & [0,1] \quad \forall\, G_{i,r} \in \mathcal{G}
\end{array}$$

(LP-Choose MSD)

For $a, b \geq 0$, we let $a + b = 1$ and $a \cdot k_1 + b \cdot k_2 = k'$, similar to MSR. Setting $z_{i,r} = a$ shows the optimum LP solution value is at most

$$\sum_{G_{i,r} \in \mathcal{G}} (6 \cdot b + 6 \cdot a) \cdot C_{i,r} + 2 \cdot b \cdot r = 6 \cdot C_2 + 2 \cdot b \cdot C_1.$$

In an optimal extreme point, at most one variable in **LP-Choose MSD** that is fractional so we set it to 1 we pick to corresponding group to be covered by a single ball. The final cost is $\min\{6 \cdot C_2, 6 \cdot C_1 + 2 \cdot b \cdot C_2 + O(\epsilon) \cdot OPT_D\} \leq (1-d) \cdot 6 \cdot C_2 + d \cdot (b \cdot 2 \cdot C_2 + 6 \cdot C_1)$ for any $d \in [0,1]$. Let At $d = \frac{6(1-b)}{2 \cdot b^2 - 2 \cdot b + 6}$ the worst case for the final bound is at $b = 1/2$ at which we see the cost is at most $\frac{72}{11} \cdot OPT'_D + O(\epsilon) \cdot OPT_D$. Adding this to the $1/\epsilon$ balls we guessed (whose diameters are at most twice their radius) and choosing ϵ sufficiently small shows we get a solution with an approximation guarantee of 6.546 for MSD, which is better than two times the MSR guarantee.

4 Getting the Bipoint Solution: Proof of Theorem 6

We again emphasize that one can slightly adapt the algorithm and analysis in [7] to prove a slightly weaker version of Theorem 6 that would still suffice for our approximation guarantees. The main difference is that the averaging of the bipoint solution costs as given in bound (1) from Section 2.4 would be bounded by $(1 + \epsilon') \cdot OPT$ for some $\epsilon' > 0$ (the running time depends linearly on $\log 1/\epsilon$).

We give an alternative approach that uses simple LP rounding. This may be of independent interest since our method of getting a single λ rather than two "close" values λ_1, λ_2 is simple in principle and may apply to other Lagrangian multipler preserving (LMP) approximations that use direct LP rounding. That is, we give a recipe to find a single λ and two corresponding solutions $\mathcal{B}_1, \mathcal{B}_2$ that uses very generic properties of the rounding algorithm.

The proof of Theorem 6 proceeds through the usual approach of using a binary search using an LMP algorithm. We begin by describing our LMP algorithm followed by a simple consolidation step which is used in some parts of the binary search. Our direct LP rounding procedure is presented here as is an outline of the binary search routine. For the sake of space, full details of how our binary search works with the rounding procedure are deferred to full version.

4.1 A Simple LMP Algorithm via Direct LP Rounding

Algorithm 2 describes our rounding procedure. Note it only depends on x' and not on λ itself.

Algorithm 2 ROUND(x').

$\mathcal{B} \leftarrow \emptyset$
for (i, r) with $x'_{i,r} > 0$ in non-increasing order of r **do**
  if $B(i, r) \cap B(i', r') = \emptyset$ for each $(i', r') \in \mathcal{B}$ **then**
    $\mathcal{B} \leftarrow \mathcal{B} \cup \{(i, r)\}$
return $\mathcal{B}$

To analyze the performance of this algorithm, we also consider the dual of $\mathbf{LP}(\lambda)$.

$$\begin{array}{lrcll} \mathbf{max} & \sum_{j\in X'} y_j & & & \\ \text{s.t.} & \sum_{j\in B(i,r)\cap X'} y_j & \leq & r+\lambda & \forall\ (i,r), r\leq R_m \\ & y & \geq & 0 & \end{array} \qquad (\mathbf{DUAL}(\lambda))$$

▶ **Theorem 8.** *Let $\lambda \geq 0$. Let x' be an optimal solution for $\mathbf{LP}(\lambda)$ and y' be an optimal dual solution for $\mathbf{LP}(\lambda)$. Let $\mathcal{B}$ denote the set returned by ROUND(x'). The balls in $\mathcal{B}$ are pairwise-disjoint and for each $(i, r) \in \mathcal{B}$ we have $r \leq R_m$ and $r + \lambda = \sum_{j\in B(i,r)} y'_j$.*

Proof. Disjointedness follows by construction. Each ball $\mathcal{B}$ has radius at most R_m since each ball is from the support of x' and $\mathbf{LP}(\lambda)$ only has variable for balls with radius $\leq R_m$. Again, since each $(i, r) \in \mathcal{B}$ lies in the support of x' then complementary slackness shows $r + \lambda = \sum_{j\in B(i,r)} y'_j$. ◀

Note the last condition shows $cost(\mathcal{B}) + \lambda \cdot |\mathcal{B}| \leq \sum_{j\in X'} y'_j = OPT_{\mathbf{LP}(\lambda)}$. Thus, we call this a "Langrangian multipler preserving" algorithm because if $\mathcal{B}''$ is obtained by tripling the radii of the balls returned by ROUND(x'), then $cost(\mathcal{B}'') + 3 \cdot \lambda \cdot |\mathcal{B}''| \leq 3 \cdot OPT_{\mathbf{LP}(\lambda)}$.

4.2 Sketch of the Binary Search

While it is fairly easy to see that using $\lambda = 0$ will have an optimal LP solution be rounded to $|X'| > k'$ balls, we need to ensure that for large enough λ that our rounding procedure produces $\leq k'$ balls in order to begin our binary search. Thus, we consider another step CONSOLIDATE$(\mathcal{B}, \lambda, \mathcal{A}, R_m)$ that tries to consolidate some of the balls output by ROUND(x') using the balls from the k-Median approximation $\mathcal{A}$. Roughly speaking, if the single radius $3 \cdot R_m$-ball centered at some $i' \in \mathcal{A}'$ is cheaper than the balls of $\mathcal{B}$ it covers, we replace them with this single ball.

Algorithm 3 CONSOLIDATE$(\mathcal{B}, \lambda, \mathcal{A}, R_m)$.

$\mathcal{B}^c \leftarrow \emptyset$
for each $i' \in \mathcal{A}$ **do**
  Let $N_{i'} = \{(i,r) \in \mathcal{B} : i \in B(i', 2 \cdot R_m)\}$
  if $3 \cdot R_m + \lambda \leq \sum_{(i,r)\in N_{i'}} (r + \lambda)$ **then**
    $\mathcal{B}^c \leftarrow \mathcal{B}^c \cup \{(i', 3 \cdot R_m)\}$
    $\mathcal{B} \leftarrow \mathcal{B} - N_{i'}$
$\mathcal{B}^c \leftarrow \mathcal{B}^c \cup \mathcal{B}$
return $\mathcal{B}^c$

We the following show in the full version.

▶ **Lemma 9.** *Let $\mathcal{B} = ROUND(x')$ for some optimal solution x' for* $\mathbf{LP}(\lambda)$ *and $\mathcal{B}^c = CONSOLIDATE(\mathcal{B}, \lambda, \mathcal{A}, R_m)$.*
1. *If $\lambda \geq 4 \cdot R_m$, then $|\mathcal{B}^c| \leq k'$,*
2. *$r \leq 3 \cdot R_m$ for each $(i,r) \in \mathcal{B}^c$*
3. *for each $(i,r) \in \mathcal{B}^c$ there is some $X_{i,r} \subseteq B(i,r)$ such that $r + \lambda \leq \sum_{j \in X_{i,r}} \cdot y'_j$ where y' is an optimal solution to* $\mathbf{DUAL}(\lambda)$*,*
4. *for different $(i,r), (i',r') \in \mathcal{B}^c$ we have $X_{i,r} \cap X_{i',r'} = \emptyset$, and*
5. *for each $j \in \mathcal{X}'$ we have $j \in B(i, 3 \cdot r)$ for some $(i,r) \in \mathcal{B}^c$.*

In particular, the total $(r + \lambda)$-cost of $\mathcal{B}^c$ is still at most $OPT_{\mathbf{LP}(\lambda)}$ since properties 3 and 4 show each ball can be paid for some variables in the optimal dual solution and no variable in the dual is charged more than once this way.

In this way, we can start the binary search with $\lambda_1 = 0$ (for which $ROUND$ will return exactly k' balls) and $\lambda_2 = 4 \cdot R_m$ (for which consolidating the rounded solution will produce $\leq k'$ balls). During the binary search, if at any step the optimal LP solution x_1 to $\mathbf{LP}(\lambda_1)$ is also an optimal LP solution to $\mathbf{LP}(\lambda_2)$ (here $[\lambda_1, \lambda_2]$ is the current interval enclosed by the binary search) or if $|\mathcal{B}| \geq k' \geq |\mathcal{B}^c|$ where $\mathcal{B}$ is obtained by rounding an optimal solution to $\mathbf{LP}(\lambda)$ and $\mathcal{B}^c$ is obtained by consolidating it, it is easy to find the bipoint solution satisfying Theorem 6.

So we focus on *break points* λ which, intuitively, are values λ where the set of optimal extreme points to $\mathbf{LP}(\lambda)$ changes. In the full version, we prove there is sufficiently-large distance between distinct breakpoints. So after a polynomial number of iterations, if the search did not terminate for one of the reasons mentioned above, then the window $[\lambda_1, \lambda_2]$ encloses exactly one break point.

We also show how to compute the largest λ such that x_1 remains optimal for $\mathbf{LP}(\lambda)$ by solving yet another a linear program that exploits complementary slackness. After computing the only breakpoint in our final binary search window, it is easy to construct the bipoint solution satisfying the requirements of Theorem 6. We note that whenever we return a bipoint solution from this binary search, we consider a post-processing routine to ensure each ball in $\mathcal{B}_1$ will intersect at least one ball in $\mathcal{B}_2$ to fulfill the requirements of Theorem 6.

5 Concluding Remarks

It may be possible to improve our analysis further by considering an even more involved approach to analyzing how to optimally cover a group using a single ball, though such an approach seems likely to produce approximations that are still a constant-factor worse than 3. What is more interesting is an observation about our new approach to finding the bipoint solution. If we ever encounter a λ such that $|\mathcal{B}| \geq k' \geq |\mathcal{B}^c|$ where $\mathcal{B}$ is the output of ROUND and $\mathcal{B}^c$ is the output of CONSOLIDATE (using $\mathcal{B}$), then Check 1 will terminate the search with bipoint solution $\mathcal{B}, \mathcal{B}^c$. If we refine the CONSOLIDATE step to perform the consolidations for $\mathcal{B}$ one at a time and stop when the number of clusters first becomes $\leq k'$, one can show that tripling the radii of these $\leq k'$ balls is a solution with cost at most $(3 + O(\epsilon)) \cdot OPT'$. But this is just for one case in our binary search. In general, is there a refinement of our binary search routine (or some other approach) that would always produce a $(3 + \epsilon)$-approximation?

References

1. N. Bansal, A. Chakrabarti, A. Epstein, and B. Schieber. A quasi-ptas for unsplittable flow on line graphs. In *Proceedings of 38th Annual Symposium on Theory of Computing (STOC)*, STOC '06, pages 721–729, New York, NY, USA, 2006. Association for Computing Machinery.
2. J. Batra, N. Garg, A. Kumar, T. Mömke, and A. Wiese. New approximation schemes for unsplittable flow on a path. In *Proceedings of the 2015 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 47–58, 2015.
3. B. Behsaz and M. R. Salavatipour. On minimum sum of radii and diameters clustering. *Algorithmica*, 73(1):143–165, September 2015.
4. V. Capoyleas, G. Rote, and G. Woeginger. Geometric clusterings, 1990.
5. M. Charikar and S. Guha. Improved combinatorial algorithms for the facility location and k-median problems. In *40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039)*, pages 378–388, 1999.
6. M. Charikar, S. Guha, É. Tardos, and D. B. Shmoys. A constant-factor approximation algorithm for the k-median problem. *Journal of Computer and System Sciences*, 65(1):129–149, 2002.
7. M. Charikar and R. Panigrahy. Clustering to minimize the sum of cluster diameters. *Journal of Computer and System Sciences*, 68(2):417–441, 2004. Special Issue on STOC 2001.
8. S. Doddi, M. V. Marathe, S. S. Ravi, D. S. Taylor, and P. Widmayer. Approximation algorithms for clustering to minimize the sum of diameters. In *In Proceedings of 7th Scandanavian Workshop on Algorithm Theory (SWAT)*, pages 237–250. Springer-Verlag, 2000.
9. M. E. Dyer and A. M. Frieze. A simple heuristic for the p-centre problem. *Operations Research Letters*, 3(6):285–288, 1985.
10. M. Gibson, G. Kanade, E. Krohn, I. A. Pirwani, and K. Varadarajan. On metric clustering to minimize the sum of radii. *Algorithmica*, 57(3):484–498, February 2009.
11. M. Gibson, G. Kanade, E. Krohn, I. A. Pirwani, and K. Varadarajan. On clustering to minimize the sum of radii. *SIAM Journal on Computing*, 41(1):47–60, 2012.
12. F. Grandoni, T. Mömke, and A. Wiese. A ptas for unsplittable flow on a path. In *To appear in proceedings of 54th ACM Symposium on Theory of Computing (STOC)*, 2022.
13. F. Grandoni, T. Mömke, and A. Wiese. Unsplittable flow on a path: The game! In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 906–926, 2022.
14. F. Grandoni, T. Mömke, A. Wiese, and A. Zhou. To augment or not to augment: Solving unsplittable flow on a path by creating slack. In *Proceedings of the 2017 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2411–2422, 2017.

15 F. Grandoni, T. Mömke, A. Wiese, and H. Zhou. A $(5/3 + \epsilon)$-approximation for unsplittable flow on a path: Placing small tasks into boxes. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, pages 607–619, New York, NY, USA, 2018. Association for Computing Machinery.

16 P. Hansen and B. Jaumard. Minimum sum of diameters clustering. *Journal of Classification*, 4(2):215–226, September 1987.

17 P. Heggernes and D. Lokshtanov. Optimal broadcast domination in polynomial time. *Discrete Mathematics*, 306(24):3267–3280, 2006.

18 D. S. Hochbaum and D. B. Shmoys. A best possible heuristic for the k-center problem. *Mathematics of Operations Research*, 10(2):180–184, 1985.

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

20 O. Kariv and S. L. Hakimi. An algorithmic approach to network location problems. ii: The p-medians. *SIAM Journal on Applied Mathematics*, 37(3):539–560, 1979.

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