**Approximation Algorithms for Clustering with Minimum Sum of Radii, Diameters, and Squared Radii**

by

Mahya Jamshidian

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science
University of Alberta

# Abstract

In this study, we present an improved approximation algorithm for three related problems. In the Minimum Sum of Radii clustering problem (MSR), we aim to select $k$ balls in a metric space to cover all points while minimizing the sum of the radii. In the Minimum Sum of Diameters clustering problem (MSD), we are to pick $k$ clusters to cover all the points such that sum of diameters of all the clusters is minimized. At last, in the Minimum Sum of Squared Radii problem (MSSR), the goal is to choose $k$ balls, similar to MSR. However in MSSR, the goal is to minimize the sum of squares of radii of the balls. We present a 3.389-approximation for MSR and a 6.546-approximation for MSD, improving over respective 3.504 and 7.008 developed by Charkar and Panigrahy (2001). In particular, our guarantee for MSD is better than twice our guarantee for MSR. In the case of MSSR, the best known approximation guarantee is $4 \cdot (540)^2$ based on the work of Bhowmick, Inamdar, and Varadarajan in their general analysis of the $t$-Metric Multicover Problem. With our analysis, we get a 11.078-approximation algorithm for Minimum Sum of Squared Radii.

*"An algorithm must be seen to be believed."*

*- Donald E. Knuth*

# Acknowledgments

I find it difficult to put into words how many ways I am grateful to my supervisor and mentor, Zachary Friggstad. Your guidance, not only in the academic area but also in life, has taught me more than I could ever imagine. This work and my studies would have not been possible without your patience and brilliance. I am forever thankful for your teaching on how to be a researcher, a colleague, and a human.

I would like to thank Mohammad R. Salavatipour and Xiaoqi Tan for being on my thesis committee and reading through my thesis as well as providing insightful suggestions that make this work a better one.

At last, I dedicate this thesis to my parents, Abbas and Hamideh. Without your sacrifices throughout my life, I could not have achieved a shred of what I have today. There are no words that can describe how grateful I am to be your daughter. I am also thankful to my sister, Dina, as her encouragement have been and will be a light in my path.

# Table of Contents

# List of Tables

# List of Figures

# Abbreviations & Acronyms

**FPTAS** Fully Polynomial Time Approximation Scheme.

**LMP** Lagrangian Multiplier Preserving.

**LP** Linear Programming.

**MSD** Minimum Sum of Diameters.

**MSR** Minimum Sum of Radii.

**MSSR** Minimum Sum of Squared Radii.

**PTAS** Polynomial Time Approximation Scheme.

**QPTAS** Quasi-Polynomial Time Approximation Scheme.

# Chapter 1

# Introduction

Clustering is one of the most well-studied problems in computing science and information technology, due to its numerous applications in Data Science and Unsupervised Learning. In many applications, the goal is to cluster the data points into some given number of clusters. Although Clustering refers to this partitioning process itself regardless of the number of clusters, many times we focus our attention to the the problems where we aim to partition the data into at most $k$ clusters. Sometimes, such a problem involves finding $k$ cluster centers and a mapping $\sigma$ from data points to the centers to minimize some objective function. One of the most studied such objective functions is the $k$-MEANS problem. In this problem, we are given a metric $(\mathcal{X}, d)$ and the input value $k$ and we wish to find $k$ partitions of data points in $\mathcal{X}$ such that the sum of squared distances from data points to their centres is minimized. In the $k$-MEANS case, a cluster prototype is the mean of the intra-cluster distances. As shown in [1], the problem in general metrics is $NP$-hard. It is also proved that the problem is $NP$-hard in Euclidean Space ($d$ dimension) even when $k = 2$ [2]. The problem is also $NP$-hard for a fixed dimension $d = 2$ for the Euclidean Space as proved in [3].

Another well-studied such clustering problem is $k$-CENTER which aims to minimize the maximum radius of among balls [4, 5]. It has been also shown how the best approximation factor one can get for $k$-CENTRE is 2 unless $P = NP$.

Yet another important example is the $k$-Median problem which aims to minimize sum of distances from data points to their centers, as studied comprehensively in [6–10]. The best current known approximation for $k$-Median is $2.675 + \frac{2}{p}$ for a fixed integer $p \geq 1$, in running time $n^{O((1/\epsilon)log(1/\epsilon))}$ based on work of [11].

Throughout, we put our focus on a different objective function for a clustering task. In this approach, we still need to pick $k$ centers, however, we also pick a radius for each of these picked centers. The constraint is to, of course, pick the centres and radii such that all the data points are present in at least one of the constructed balls, while the objective is to minimize the sum of all the radii. Specifically, we study the following problem.

**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 $C \subseteq X$, $|C| \leq k$ 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. $d(j, i) \leq r_i$). The goal is to minimize the total radii, i.e. $\sum_{i \in C} r_i$.*

That is, we want to cover $X$ using at most $k$ balls with minimum total radius. For example, perhaps we want to broadcast messages to all points by selecting $k$ sources with minimum total broadcast radius.

We also consider a related problem to minimize sum of diameters of the clusters. More percisely, we define the MSD problem as follows.

**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 $\alpha$-approximation algorithm for MSR yields an $2\alpha$-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 $OPT_R \leq OPT_D$ 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 $\leq 2 \cdot r_i$ and the sum of diameters would then be at most $2\alpha \cdot OPT_R \leq 2\alpha \cdot OPT_D$.

As an intuition to the MSR problem, centres can be thought of as prospective mobile tower locations, whereas the points in $\mathcal{X}$ can be thought of as client locations. A tower can be set up in such a way that it can service consumers within a given radius. However, the cost of service rises with the broadcast distance travelled. The goal is to serve all clients at the lowest possible cost.

When calculating the amount of energy required for wireless transmission, it is typical to think about the cost function to be Minimum Sum of Squared Radii. In reality, it requires power proportionate to $r^2$ to broadcast up to a certain radius $r$. This inspires a form of MSR in which we want to reduce the sum of the radii's squares. The Minimum Sum of Squared Radii (MSSR) issue is what we refer to as.

## 1.1 Preliminaries

This section presents the required prerequisites for understanding the methods and approaches discussed in the rest of the thesis.

### 1.1.1 Metric

This work is primarily concerned with metric spaces which can be naively described as a space including points with distances between them. Let $\mathcal{X}$ and $d : \mathcal{X} \to \mathcal{X}$ be a set and a function, known as *set of points* and *distance function*, respectively. If the distance function satisfies all the following property for points $x, y, z \in \mathcal{X}$, then the pair $(X, d)$ is a metric space.

1. $d(x, y) \geq 0$

2. $d(x, x) = 0$

3. $d(x, y) \neq 0$ if $x \neq y$

4. $d(x, y) = d(y, x)$, and

5. $d(x, z) \leq d(x, y) + d(y, z)$.

Property (5) is known as the triangle inequality which has many algorithmic and analytic purposes for our studies.

A well-studied metric is the *Euclidean metric* where elements of $X$ are present in a Euclidean space and their distance satisfies Euclidean distance function. Another example of metric spaces is the *Doubling Dimension*, generalizing Euclidean metrics is some sense. For any $u \in \mathcal{X}$, $B(x, r)$ is the ball of radius $r$ centered at $u$, i.e. $B(u, r) = \{v : c \in X, d(u, v) \leq r\}$. The smallest value $k > 0$ such that each ball in space can be covered by at most $2^k$ balls of half radius is known as the *doubling dimension* of the finite space $X$. Note that $d$-dimensional Euclidean metric has doubling dimension $\Theta(d)$.

Also, the following notation are necessary for the rest of our study.

**Definition 3** *A ball $B$ in a metric space $(\mathcal{X}, d)$, identified by two values $i \in \mathcal{X}, r \in \mathbb{R}_{\geq 0}$ is the following set $\{j \in \mathcal{X} : d(i, j) \leq r\}$. Then, $i$ is the centre of $B$ and $r$ is known as the radius of $B$.*

**Definition 4** *For a set $C \subseteq \mathcal{X}$, the diameter of the induced cluster on $C$ is equal to $\max_{i,j \in C} d(i, j)$.*

### 1.1.2 Approximation Algorithms

Before describing the motivation behind approximation algorithms, we need to define an optimization problem. An optimization problem is the task of finding a *feasible* solution that optimizes some objective value. It then can be shown as the pair $(\mathcal{S}, o)$ where $S$ and $o$ are the set of feasible solutions and objective function $o : \mathcal{S} \rightarrow \mathrm{R}$, respectively. Depending on the problem, the task of optimizing could be minimizing or maximizing the objective function $o$, in that case we call the minimization or maximization problems.

An example of a optimization instance $(S, o)$ is the problem of finding the minimum cost route in a weighted graph (which passes through all the nodes). In this minimization problem, the elements of $\mathcal{S}$ are routes $R$ that passes through each node of the input graph and $o$ is the sum of all the weights on the edges of $R$.

Let us define $\delta : \mathbf{Z}^+ \to \mathbf{R}^+$ a function that gets the input size of an optimization problem $\Gamma$ as input and maps it to a real number. Then, the algorithm $A$ is a $\delta$-approximation algorithm for $\Gamma$ if any input $I$ to $A$, the outputted feasible solution $S(I)$ holds $f(I) \leq \delta(I) \cdot OPT$ and runs in polynomial time in the input size. Note that the approximation factor may or may not depend on the input size. There are problems for which we have approximation factors that will not grow with respect to the input size.

A vertex cover of a graph is a set of vertices that includes at least one endpoint of every edge of the graph. The problem of finding a Minimum Vertex Cover is a classical optimization problem. It is NP-hard and has a 2-approximation algorithm. As another example, given a set of elements $\{1, 2, \cdots, n\}$ (called the universe) and a collection $\mathcal{S}$ of $m$ sets whose union equals the universe, the set cover problem is to identify the smallest sub-collection of $\mathcal{S}$ whose union equals the universe. There is a $O(\log n)$-approximation algorithm for this problem. In most scenarios, we hope to get $(1+\epsilon)$-approximation for arbitrarily small, positive and fixed $\epsilon$.

An algorithm for an optimization problem $\Gamma$ is called *Polynomial time approximation scheme (PTAS)* if for every instance and fixed positive $\epsilon$ input, it runs in $poly(|I|)$ and the output $\mathbf{S}$ admits $f(\mathbf{S}) \leq (1+\epsilon) \cdot OPT$. We can require $A$ to run in polynomial in the size of input and $\frac{1}{\epsilon}$ and then $A$ is a *fully polynomial time approximation scheme (FPTAS)*. Another approximation scheme is the quasi-polynomial-time approximation scheme or QPTAS. A QPTAS has time complexity $n^{polylog(n)}$ for each fixed $\epsilon > 0$.

In computational complexity theory, the class APX (approximable) is the set of optimization problems that allow polynomial-time approximation algorithms with approximation ratio bounded by a constant; i.e. they have a $O(1)$-approximation algorithm.

### 1.1.3 Linear Programming

As one of the fundamental optimization problems, *Linear Programming (LP)* is a problem expressed as follows.

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & \begin{aligned} Ax &\geq b \\ x &\geq 0 \end{aligned} \end{aligned} \tag{1.1}$$

where $c$ is a vector in $\mathbb{R}^n$, $A$ is a $m \times n$ matrix with real value entries, and $b$ is in $\mathbb{R}^m$. Note that the operator $\geq$ mentioned in the body of the program is the pair-wise $\geq$ operator between two vectors. In other words, $Ax \geq b$ is a set of $n$ inequalities in the form of $\sum_j A_{i,j} x_j \geq b_j$. If $x$ satisfies all the constraints of the LP, then it is called a feasible solution. An LP is feasible if it has at least one feasible solution. It could be that an LP is feasible but the objective function is unbounded, which then makes the LP unbounded. A linear program is unbounded if it is feasible but its objective function can be made arbitrarily "good".

As of a history on using and solving linear programs, The work of [12] described an algorithm that could solve the program efficiently in most cases. At the time, being mid to late 40s, Dantzig used LP for planning problems in US Air Force. Later, many industries applied the work for daily planning. Later it was shown that LP is polynomial-time solvable by [13] using an algorithm called "Ellipsoid Method". In practice, however, the algorithm is slow and of little practical value, though it did serve as inspiration for later work that proved to be far more useful. Later, Narendra Karmarkar [14] proposed a novel interior-point method for addressing linear-programming problems, which was a bigger theoretical and practical development in the discipline.

In (1.1), the program described is called the primal LP. For a primal LP, there is a dual LP defined as follows:

$$\begin{aligned} \text{maximize} \quad & b^T y \\ \text{subject to} \quad & A^T y \leq c \\ & y \geq 0 \end{aligned} \tag{1.2}$$

The following is useful for many applications of LP:

**Theorem 5 (Weak Duality (Theorem 3.20 [15]))** *If $x$ and $y$ are feasible solutions to primal and dual programs respectively, then*

$$c^T x \geq b^T y$$

We can conclude from the weak duality theorem that every feasible solution to a dual program is a lower-bound to the optimal value of the primal program. The inequality mentioned in the theorem holds tightly when both the primal and dual have an optimal value, as stated in Theorem 6.

**Theorem 6 (Strong Duality(Theorem 3.20 [15]))** *The primal program has an optimal solution if and only if the dual program has an optimal solution. Furthermore, their optimal value coincide:*

$$c^T x = b^T y.$$

For our discussion later in this study, we need the following definition.

**Definition 7** *Let $P = \{x : Ax = b, x \geq 0\} \subseteq \mathbf{R}^n$. Then $x \in \mathbf{R}^n$ is an Extreme Point solution of $P$ if there is no non-zero vector $y \in \mathbf{R}^n$ such that $x + y$ and $x - y \in P$.*

The corner points of the set of possible solutions are depicted graphically as extreme point solutions. We can show that for bounded linear programs, there is always an optimal extreme point solution.

**Lemma 8** *Let $P = \{x : Ax \geq b, x \geq 0\}$ and assume that $\min\{c^T x : x \in P\}$ is finite. Then there is always an extreme point optimal solution that can be computed in polynomial time.*

At last, using linear algebraic properties and the given lemma, one can show the following, also known as Rank Lemma.

**Lemma 9 (Lemma 2.1.4 [16])** *Let $P = \{x : Ax \geq b, x \geq 0\}$ and let $x$ be an extreme point solution of $P$ such that all entries of $x$ is non-zero. Then any maximal number of linearly independent tight constraints of the form $A_i x = b_i$ for some row $i$ is equal to the number of variables.*

*Integer Programming* is an optimization problem represented similarly to an LP except that each variable $x$ can be either 1 or 0. Many of the combinatorial problems can be modeled using an integer program, and be approximated using a relaxation from the integer program to an LP since Integer Programming is NP-hard. Although we have concrete way of solving an LP, there are many methods that one can use to try to find a feasible solution to the main integer program, and also bound the objective function.

Another important property associated with Primal and Dual programs is the *Complementary Slackness.* Note how if we have an optimal solution to a program, whether primal or dual, it does not necessary mean that all the constraints are tight. This leads to the following useful fact:

**Theorem 10 (Complementary Slackness (Theorem 3.24 [15]))** *Let $\bar{x}$ be a feasible solution for the primal program, and $\bar{y}$ a feasible solution to the corresponding dual. $\bar{x}$ and $\bar{y}$ are both optimal if and only if:*

1. $x_j \bar{>} 0 \longrightarrow \sum_{i=1}^{m} \bar{y}_i A_{i,j} = c_j$ *for any* $1 \leq j \leq n$
2. $y_i \bar{>} 0 \longrightarrow \sum_{j=1}^{n} \bar{x}_j A_{i,j} = b_i$ *for any* $1 \leq i \leq m$

We can cast many combinatorial problems as integer programs and then relax to linear programs.

### 1.1.4 $k$-Centre Problem

The metric $k$-Centre problem is extensively studied in theoretical computer science. The problem is considered a clustering task for which we are given a metric, $(\mathcal{X}, d)$ and the parameter $k$. We aim to pick $k$ centres from the set $\mathcal{X}$ such that the maximum distance between a data point in $\mathcal{X}$ and its closest centre is minimized. There is a simple greedy 2-approximation algorithm for this problem running in $O(nk)$ and it is shown there are no $(2-\epsilon)$-approximation algorithms for $k$-centre for any $\epsilon > 0$ unless $\mathbf{P} = \mathbf{NP}$ [5].

### 1.1.5 Lagrangian Relaxation

Let us describe this method with an example. Consider the following Linear Programming Relaxation of MSR. Here, $x_{(i,r)} = 1$ corresponds to us choosing to open centre $i \in \mathcal{X}$ with radius $r$.

$$\begin{aligned} \text{minimize :} \quad & \sum_{(i,r)} r \cdot x_{(i,r)} \\ \text{subject to :} \quad & \sum_{(i,r): j \in B(i,r)} x_{(i,r)} \geq 1 \quad \text{for each } j \in \mathcal{X} \\ & \textstyle\sum_{(i,r)} x_{(i,r)} \leq k \\ & \mathbf{x} \geq 0 \end{aligned}$$

Note that the pair $(i, r)$ are from the set $\mathcal{X} \times [D]$ where $D$ is the maximum distance between any pair in $\mathcal{X}$ which we can scale to be an integer value, and hence, is polynomial. We will use this linear program to design a constant-factor approximation. To do this, we "Lagrangify" the cardinality constraint. In particular, for any "given" $\lambda \geq 0$ we consider the following LP where we call $\lambda$ the Lagrangian multiplier.

$$\begin{aligned} \text{minimize :} \quad & \sum_{(i,r)} r \cdot x_{(i,r)} + \lambda(x_{(i,r)} - k) \\ \text{subject to :} \quad & \sum_{(i,r): j \in B(i,r)} x_{(i,r)} \geq 1 \quad \text{for each } j \in \mathcal{X} \\ & \sum_{(i,r)} x_{(i,r)} \leq k \\ & \mathbf{x} \geq 0 \end{aligned}$$

Then, we will be considering the program where we drop the constant term in the objective function. In this way, we penalize picking too many centres in the objective function. But note how the solution must still be feasible for the first LP, which then requires another step in most solutions.

**Lemma 11** *For some $\lambda \geq 0$, if an optimum integer solution to the Lagrangified LP uses exactly $k$ balls, then it is an optimum solution to the MSR problem as well.*

**Proof.** If there are exactly $k$ centres in the found solution, the the term $\lambda(x_{(i,r)} - k)$ cancels out in the objective function. Then, the LP solution can be translated to a discrete solution easily with the value at most optimal value of LP. ■

Note that it is still challenging to determine the Lagrangified LP's optimum integer solution, because the optimum may not require precisely $k$ balls. In order to obtain an approximately optimal solution that employs precisely $k$ balls, our algorithm will approximate the search for the best integer solution and test several values of lambda.

Even this might not be feasible. In the end, our approach will discover some $\lambda \geq 0$ and two distinct nearly-optimal integer solutions for this identical Lagrangified LP: one uses at least $k$ balls and one uses at most $k$ balls, as we describe and demonstrate in Chapter 3. A "bi-point" solution is a pair of such solutions for a certain $\lambda \geq 0$. The name "bi-point" is chosen since the original LP for MSR may be solved realistically and inexpensively by appropriately averaging these two integer answers.

## 1.2 Prior 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 [17]. In polynomial time, the best approximation algorithm is the stated 3.504 approximation by Charikar and Panigrahy. [18]. Interestingly, [17] show that MSR can be solved exactly in $n^{O(\log n \cdot \log \Gamma)}$ where $\Gamma$ is the *aspect ratio* of the metric (maximum distance divided by minimum nonzero distance). Using Randomized Algorithm, this yields a quasi-PTAS for MSR: i.e. a $(1+\epsilon)$-approximation with running time $n^{O(\log 1/\epsilon + \log^2 n)}$. The main idea that underlies this result is that if we probabilistically partition the metric into sets with at most half the original diameter, then with high probability only $O(\log n)$ balls in the optimal MSR solution are "cut" by the partition.

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. 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 [19, 20] However, improved constant-factor approximations were subsequently produced [21–23], that is until a PTAS was finally found by Grandoni et al. [24].

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 [25]. Prior to the work studied in this dissertation, 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 [18].

MSR and MSD have been studied in special cases as well. In constant-dimensional Euclidean metrics, MSR can be solved exactly in polynomial time [26]. 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 [27]. 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 [28]. However, MSD is $\mathbf{NP}$-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 [29, 30].

Another related problem is lower-bounded min sum of radii with outliers (LBkSRO) as discussed in [31]. In this problem, we are given the set $\mathcal{X}$ in which for each element

$u$, there is an input $L_u$. Then, the goal is to pick $k$ centres and radii such that cluster with center $u$ must have at least $L_u$ data points and there has to be at most $m$ unclustered data points. Then, the objective is to minimize the sum of radii. They described a 12.365-approximation algorithm for LbkSRO. The version where $m = 0$, known as LbkSR has a 3.83-approximation factor, using similar methods. In [32], we see another fault tolerant version. They discuss the Metric Multi-Cover problem, in which we have a set of data points $\mathcal{X}$ and set of candidate facilities $\mathcal{Y}$ and a demand value $k$ for all the data points. The goal here is to pick pairs of $(i, r)$ as many as we wish to minimize the sum of radii to power of $\alpha$ while each data point has to be present in at least $k$ balls. This modification of the problem admits an approximation factor of $2(108)^\alpha$. If we make a constraint to have at most $t$ centres, then their work gives an approximation factor of $4(540)^\alpha$. On another note, if each data point has specific demand, the factor changes to $2 \cdot (144)^\alpha$. At last, by adding an opening cost to each centre candidate, the approximation factor would be $(216)^\alpha$.

For the Euclidean space for MMC, there is a $23.02 + 63.95(k - 1)$ approximation algorithm where $k$ is the coverage demand of the points [33]. They also consider a non-discrete version of MMC, where there are areas to cover rather than discrete data points. Their work proves a $63.94 + 177.64(k - 1)$ approximation factor. For a client specific version of MMC in the Euclidean, [34] proves a $4(27\sqrt{2})^\alpha$ factor. Also, [35] studies the MMC problem with a penalty added for each uncovered data point. Then the objective function is to minimize minimum sum of radii and sum of the penalties. They give a $3^\alpha + r_{max}^\alpha$ approximation algorithm in Euclidean space.

## 1.3 Our work and Organization

Prior to this work, Charikar and Panigrahy presented a 3.504-approximation for MSR [18]. Since an $\alpha$-approximation for MSR yields a $2\alpha$-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 12** *There is a polynomial-time 3.389-approximation for MSR.*

We obtain this primarily by refining a so-called bipoint rounding step from [18]. That is, our improvement for MSR mainly focuses in the last phase of the algorithm in [18] 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.

We also present an alternative method for obtaining these two subsets of balls by considering a simple rounding of a linear programming (LP) relaxation, the Lagrangian relaxation of the problem obtained by relaxing the constraint that at most $k$ centres are chosen, rather than the primal-dual technique used in [18].The rounding algorithm is incredibly simple and we employ fairly generic arguments to convert it to a bipoint solution for a single Lagrangian multiplier $\lambda$, this approach 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 $\lambda$, as long as the LMP approximation is from direct LP rounding. However, we emphasize that 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 13** *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.

Finally, we get to discuss the MSSR problem, using the same machinery. The algorithm makes use of the same bipoint rounding and placing of the centres. At last, we have the following result.

**Theorem 14** *There is a polynomial-time 11.078-approximation for MSSR.*

Note that this is best known result for the MSSR case. As mentioned, objective functions with powers greater than 1 has been studied in other works. Although, their corresponding problems are more strict and hence, they establish a larger approximation factor. One can also analyze Charikar's approach to get a constant approximation.

## Organization of the Dissertation

The MSR approximation is given in Chapter 2. Our algorithm follows the same general structure as the algorithm in [18] so we defer the details behind one significant step to chapter 3. Our MSD and MSSR approximation is also given in chapter 2.

Finally, our new approach to obtaining a clean bipoint solution is discussed in chapter 3. Brief concluding remarks are given in Section 4.

At last, the main part of the analysis of the improved approximation factor is about fixing a particular constant and picking the proper configuration for the combined solution. The result of such analysis is stated in the chapter 2, however tables summarizing how generic choices of constants would lead to approximation guarantees are in the appendix A and B, the purpose is to demonstrate that our choices are optimal.

# Chapter 2

# Improved approximation of MSR, MSD, and MSSR

In this chapter we describe the algorithm for MSR, MSD, and MSSR. In each of these cases, we will be going over the process of solving the LP for the sake of finding the proper Lagrangian value, then combining the two solutions for an approximation guarantee. Based on the objective function, one may have to change some parts of the algorithm, all of which are discussed in their respective sections.

## 2.1 Minimum Sum of Radii

At first, we put our focus on the MSR problem, and then apply the same procedure and ideas to the others.

From now on in this thesis, $n = |X|$. We know that $d(i, i') > 0$ for distinct $i, i' \in X$ because it is a metric space. We refer to a pair $(i, r)$ as a ball with the understanding it is referring to the set $B(i, r)$.

In some parts of 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 by the furthest point that is covered by that ball. So there are only $O(n^2)$ 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 the balls. For such a subset, we let $cost(\mathcal{B}) = \sum_{(i,r) \in \mathcal{B}} r$ be the total radii of these balls.

We will fix some small constant $\epsilon > 0$ such that $1/\epsilon$ is an integer. Note that a smaller $\epsilon$ leads to a better guarantee with increased (but still polynomial) running time. We will be able to pick a small enough $\epsilon$ such that it will hide in the approximation guarantee and that is to why it is not mentioned in the statement of Theorem 12. We assume $k > 1/\epsilon$, otherwise we can simply use brute force to find the optimum solution in $n^{O(1/\epsilon)}$ time.

Here we provide a detailed explanation for the MSR algorithm and all its subroutines. The main contribution happens at the last phase of algorithm, where we establish a better approximation factor. 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 "bi-point" solution that we describe in the Chapter 3. By bi-point, we simply mean two subsets of balls $\mathcal{B}_1, \mathcal{B}_2$ with $|\mathcal{B}_1| \geq k \geq |\mathcal{B}_2|$ so some averaging of these balls looks like a feasible fractional solution using exactly $k$ balls.

### 2.1.1 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. In other words, if our optimal solution has the fewest number of balls, then the centre of each ball must be outside of all the other balls, otherwise we can merge the balls and not increase the objective function.

Similar to [18], 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 discussion. That is, let $\mathcal{B}' \subseteq \mathcal{B}^*$ be such that $|\mathcal{B}'| = 1/\epsilon$ and $r \leq r'$ for each $(i,r) \in \mathcal{B}^* - \mathcal{B}'$ and $(i',r') \in \mathcal{B}'$.

Let $R_m$ be the minimum radius of a ball in $\mathcal{B}'$. Remember that since we already have guessed the largest balls and the sum of those guessed balls cannot be greater than $OPT$, then $R_m \leq \epsilon \cdot OPT$. 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 $OPT' = OPT - \sum_{(i,r)\in\mathcal{B}'} r$ 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 $'$.

Our guessing step must perform a "precheck" for this guess as follows before proceeding. By the work of [5], there is a 2-approximation algorithm for the $k$-CENTRE problem. Then, we run this standard 2-approximation for the $k'$-CENTRE instance on the metric restricted to $X'$. 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 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 $i \in \mathcal{A}$.

**Theorem 15** *Given an optimal solution set $\mathcal{B}^*$ with value $OPT$ and fewest number of balls, let $\mathcal{B}'$ be the set of $\frac{1}{\epsilon}$ largest balls in $\mathcal{B}^*$. Then, if $R_m$ is the minimum radius in $\mathcal{B}'$, we have $R_m \leq \epsilon \cdot OPT$.*

*Furthermore, let $X' = X - \cup_{(i,r)\in\mathcal{B}'} B(i,r)$, the set of data points not covered by $\mathcal{B}'$. Then, if $\mathcal{A}$ is the set of centres in the output of 2-approximation $(k - \frac{1}{\epsilon})$-CENTRE algorithm, then value of $A$ is at most $2 \cdot R_m$.*

When analyzing the rest of the algorithm, we 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 (which are in $O(n^\epsilon)$), so it will be at most the cost of the solution found when $\mathcal{B}'$ was guessed correctly.

### 2.1.2 Step 2: Getting a Bi-point Solution

The output from this step is similar to [18]. We remark that their approach would suffice for our purposes, except there would be yet another "$\epsilon$" introduced with their technique. We will be following a different approach primarily to show there is a simple and direct LP rounding routine and, more importantly, to give a generic procedure that is likely to apply to most LP-rounding LMP approximations to get a bi-point solution where both points can be compared with the optimal LP solution for a single value $\lambda$. At first, we will be describing the LP program relaxation for the problem of MSR, as follow:

$$\begin{array}{lrcl} \textbf{min} & \sum_{(i,r)} r \cdot x_{(i,r)} & & \\ \textbf{s.t.} & \sum_{(i,r):j \in B(i,r)} x_{(i,r)} & \geq & 1 \quad \forall\, j \in X' \\ & \sum x_{(i,r)} & \leq & k \\ & x & \geq & 0 \end{array} \qquad (\textbf{LP})$$

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 $\lambda$ 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 $O(n^2)$ variables.

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

The following is standard and follows by considering the natural $\{0,1\}$-integer solution corresponding to the balls in $\mathcal{B}^* - \mathcal{B}'$.

**Lemma 16** *For any* $\lambda \geq 0$*, let* $OPT_{\mathbf{LP}(\lambda)}$ *denote the optimum value of* $\mathbf{LP}(\lambda)$*. Then* $OPT_{LP(\lambda)} - \lambda \cdot k' \leq OPT'$*.*

**Proof.** Note that $OPT_{LP(\lambda)} - \lambda \cdot k'$ is the optimal value for the Lagrangified original LP for the problem. Then, note how in the original LP, we must have $\sum_{(i,r)} x_{(i,r)} \leq k^{'}$.

Then, the statement follows from the fact that the optimal value for the original LP is a lower-bound for $OPT^{'}$. ■

The following theorem is required for the analysis later studied in Chapter 3. We will state it here for the sake of understanding the next step of the bi-point solution construction.

**Theorem 17** *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* $r \leq 3 \cdot R_m$ *(*$R_m$ *is the smallest radius in the set of guessed balls),*
- *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*
- $cost(\mathcal{B}_\ell) + \lambda \cdot k_\ell \leq OPT_{\mathbf{LP}(\lambda)}$

It could be that $\mathcal{B}_1 = \mathcal{B}_2$, in which case $k_1 = k' = k_2$ must hold. Notice that if $k_1 = k'$ then $cost(\mathcal{B}_1) \leq OPT'$ then by Lemma 16 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 $3 \cdot OPT$ (metric property 5, triangle inequality). A similar approximation follows if $k_2 = k$. We do not distinguish these cases in our full analysis below.

As a warm-up, we consider the case when one of $k_1$ and $k_2$ is equal to $k^{'}$.

**Theorem 18** *If the bi-point step yields k balls (i.e.* $k_1 = k^{'}$*), then we have a 3-approximation.*

**Proof.** For this proof, we simply consider the LP itself. Note that if we have exactly $k'$ balls in the solution, then we have $\sum_{(i,r)} x_{(i,r)} = k'$, which means in the objective function of the Lagrangified LP, the term $\lambda(k' - \sum_{(i,r)} x_{(i,r)})$ cancels out. At last, using the rounding argument we can simply show that this solution is at most 3 times the optimal solution, which is lower-bounded by $OPT'$ value of the LP. Note that this is a rare case, and the idea behind finding and combining two solutions is that this case may not happen. ■

### 2.1.3 Step 3: Combining Bi-point Solutions

Let $\lambda, \mathcal{B}_1, \mathcal{B}_2$ be the *bi-point solution* from Theorem 17. 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 17 and the second by Lemma 16.

$$a \cdot C_1 + b \cdot C_2 \leq a \cdot (OPT_{\mathbf{LP}(\lambda)} - \lambda \cdot k_1) + b \cdot (OPT_{\mathbf{LP}(\lambda)} - \lambda \cdot k_2) = OPT_{\mathbf{LP}(\lambda)} - \lambda \cdot k' \leq OPT' \tag{2.1}$$

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 is 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 [18]. 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 show 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 [18], 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, [18] 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 $\mathcal{B}_1$.

**How to cover a group with a single ball**

From here, the approach in [18] 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 $X'$.

**Theorem 19** *If one decide to replace a group $G_{(i,r)}$ by a single ball, the cost of the ball is at most $\frac{11}{8} \cdot r + 3 \cdot cost(G_{(i,r)})$.*

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 [18], 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.

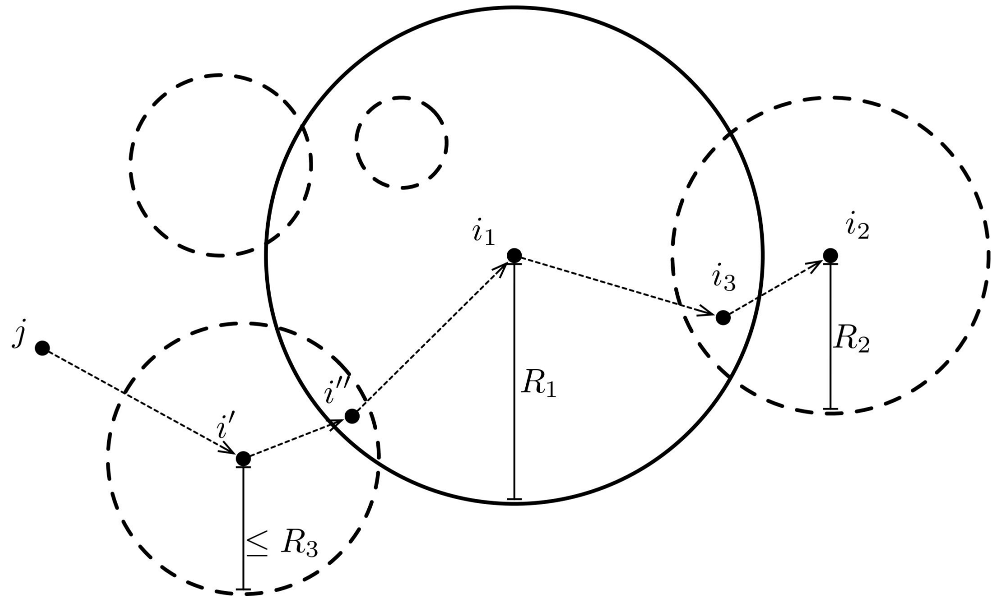


Figure 2.1: A depiction of a group $G_{(i_1,R_1)}$. The solid ball is $B(i_1, R_1)$ and the dashed balls are those in $G_{(i_1,R_1)}$. Point $j$ is covered by tripling the ball centered at $i'$. The dashed path depicts the way we bound $d(j, i_2)$ in the second part of the case **Centering at** $i_2$.

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.

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 $j \in B(i', 3r')$ 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 2.1 depicts this group and one case of the analysis below.

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

$$d(j, i_1) \leq d(j, i') + d(i', i'') + d(i'', i_1) \leq 3 \cdot R_2 + R_2 + R_1 = R_1 + 4 \cdot R_2.$$

  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

$$d(j, i_2) \leq d(j, i') + d(i', i'') + d(i'', i_1) + d(i_1, i_3) + d(i_3, i_2) \leq 3 \cdot R_3 + R_3 + R_1 + R_1 + R_2$$
$$= 2 \cdot R_1 + R_2 + 4 \cdot R_3.$$

  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

$$d(j, i_3) \leq d(j, i') + d(i', i'') + d(i'', i_1) + d(i_1, i_3) \leq 3 \cdot R_3 + R_3 + R_1 + R_1 = 2 \cdot R_1 + 4 \cdot R_3.$$

  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 optimization steps have been deferred to the Appendix A for the interested reader. 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 $C^{(2)} \leq \frac{6}{5} \cdot R_1 + 3 \cdot R_2 \leq \frac{6}{5} \cdot r + 3 \cdot C_{(i,r)}$.

- **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 17, then the radius of $B'_{(i,r)}$ is also seen to be at most, say, $21 \cdot R_m$. That is because of two facts

1. With additional preprocessing described in next chapter, a ball in the output of rounding has radius at most $3 \cdot R_m$, and

2. If we pick our centre to be $i_2$, we might have the case where the radius is $2 \cdot R_1 + R_2 + 4 \cdot R_3$.

Then, by replacing $R_1$, $R_2$, and $R_3$ by $3 \cdot R_m$, the radius of the fractional group is bounded by $21 \cdot R_m$.

### 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 $B'_{(i,r)}$. 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 $21 \cdot R_m$.

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

**(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, according to lemma 9. 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 $a\cdot k_1 + b\cdot k_2 = k'$. Thus, setting $z_{(i,r)} = a$ for each $G_{(i,r)} = 1$ is feasible as $1 - z_{(i,r)} = b$, $\sum_{G_{(i,r)}\in\mathcal{G}} |G_{(i,r)}| = k_2$, and there are at most $k'_1$ terms in this sum. The value of this solution is

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

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

**Lemma 20** *In polynomial time, we can compute a set of at most $k'$ balls with total radius at most $\frac{11}{8}\cdot b\cdot C_2 + 3\cdot C_1 + 21\cdot\epsilon\cdot OPT$ which cover all points in $X'$.*

Finally, we can complete our analysis. Recall our simple solution of tripling the balls in $\mathcal{B}_2$ has cost at most $3\cdot C_2$ 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 2.1 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 $\epsilon$.

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

**Algorithm 1** MSR Approximation

```
S ← ∅        {The set of all solutions seen over all guesses}
for each subset B'_j of 1/ε balls do
    let X', R_m be as described in Section 2.1.1
    (A, R) ← k-CENTRE 2-approximation on X'
    if R > 2 · R_m then
        reject this guess B' and continue with the next
    let B_1, B_2, λ be the bi-point solution from Algorithm 5 {see Theorem 17}
    let G be the groups (a partitioning of B_1) described in Section 2.1.3
    for each G_(i,r) ∈ G, let B'_(i,r) be the cheapest ball covering ∪_{(i',r')∈G_(i,r)} B(i', 3·r')

    let z' be an optimal extreme point to LP-Choose
    B^(1) ← {B'_(i,r) : z'_(i,r) > 0} ∪ ⋃_{z'_(i,r)=0} {(i', 3 · r') : (i', r') ∈ G_(i,r)}
    B^(2) ← {(i, 3 · r) : (i, r) ∈ B_2}
    let B be {(i, 3 · r) : (i, r) ∈ B'} plus the cheaper of the two sets B^(1) and B^(2)
    S ← S ∪ {B}
return the cheapest solution from S
```

## 2.2 Minimum Sum of Diameters

Here, we observe that a slight modification to the MSR approximation in fact yields a 6.546-approximation for MSD.

**Lemma 21** *If there is an $\alpha$-approximation for MSR, then there is a $2\alpha$-approximation for MSD.*

**Proof.** Consider the ball $B$ in the solution for MSR using the $\alpha$-approximation. Note that each pair of data points in $B$ are at most $2\alpha$ apart. Hence, there exists a ball with diameter at most $2\alpha$ covering the same set of data points. ■

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 17 to get a bi-point 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.

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

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

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

.

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 By setting $d = \frac{6(1-b)}{2 \cdot b^2 - 2 \cdot b + 6}$, the worst case analysis for the final bound happens when $b = 1/2$, 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 $\epsilon$ 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.

## 2.3 Minimum Sum of Squared Radii

The algorithm for MSSR follows the exact same procedure as MSR, the only modification happens at the phase where we analyze the cost of a group based on the centre picked for it. The case by case analysis is similar to the MSR Case. However, note that the objective function value in the lagrangified LP is in the following form:

$$\sum_{(i,r)} x_{(i,r)}(r^2 + \lambda)$$

Remember the upper-bounds for a single ball that is centred in eitehr of $i_1, i_2$, and $i_3$, as depicted in 2.1. 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 optimization steps have been deferred to the Appendix B.

We then find ourselves in two cases,

1. $R_2 \geq \frac{1}{2} \cdot R_1$, then the cost is $9 \cdot R_2 \leq 9 \cdot C(G_{(i_1,R_1)})$, and

2. $R_2 \leq \frac{1}{2} \cdot R_1$, the cost is upper-bounded by $(\frac{27}{4} \cdot R_1^2 + 9 \cdot R_2^2)$.

In any case, we see that by selecting $B'_{(i,r)}$ optimally, the radius is at most $\frac{27}{4} \cdot R_1^2 + 9 \cdot R_2^2$. This is the solution where we aim to pick a group and merge it. Similar to before, we will compare the outcome of the corresponding *Choose LP* with $C_2$, the solution in which we just tripled the radii of the output balls of Rounding.

Now we must again choose which groups to merge similar to both MSR and MSD case. Hence, consider the following LP-choose.

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

To consolidate the groups, compute an optimal extreme point to **LP-Choose (MMSR)**. 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 9. 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 $(21 \cdot R_m)^2 \leq 21^2 \cdot \epsilon \cdot OPT$.

The final cost is $\min\left\{9 \cdot C_2, 9 \cdot C_1 + \frac{27}{4} \cdot b \cdot C_2 + O(\epsilon) \cdot OPT_{MSSR}\right\} \leq (1-d) \cdot 9 \cdot C_2 + d \cdot \left(b \cdot \frac{27}{4} \cdot C_2 + 9 \cdot C_1\right)$ for any $d \in [0,1]$.

Then, we need to find $d$ such that we can upper-bound $\min\left\{9 \cdot C_2, 9 \cdot C_1 + \frac{27}{4} \cdot b\right\}$ with an expression $\beta \cdot (aC_1 + bC_2)$. Note that the parametric equation for $\beta$ in the maximized point is

$$\beta = \frac{9d}{1-b}$$

which we set equal to

$$\frac{9(1-d)+\frac{27}{4}bd}{b}.$$

Then, we can write $d = \frac{9(1-b)}{\frac{27}{4}b^2 - \frac{27}{4}b + 9}$. Then, is maximized at $b = \frac{1}{2}$ and then $d$ is equal to $\frac{32}{55}$.

Then we proceed to construct the choose LP and at last, we get a 11.078-approximation algorithm.

# Chapter 3

# Obtaining the solutions

In this chapter, we go over details for the obtaining two solutions for the bi-point combination step.

## 3.1 Overview

We again emphasize that one can slightly adapt the algorithm and analysis in [18] to prove a slightly weaker version of Theorem 17 that would still suffice for our approximation guarantees. The main difference is that the averaging of the bipoint solution costs as given in bound (2.1) from Section 2.1.3 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 $\lambda$ rather than two "close" values $\lambda_1, \lambda_2$ 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 $\lambda$ and two corresponding solutions $\mathcal{B}_1, \mathcal{B}_2$ that uses very generic properties of the rounding algorithm.

## 3.2 A Simple LMP Algorithm via Direct LP Rounding

Algorithm 2 describes our rounding procedure. Note it only depends on $x'$, the solution to $LP(\lambda)$ and not on $\lambda$ itself. However, the role of $\lambda$ is implicitly present in the solution $x^{'}$.

---

**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 $\mathbf{LP(\lambda)}$.

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

Let $x^{'}$ be the optimal solution for $\mathbf{LP(\lambda)}$ and $y^{'}$ be the optimal solution to the corresponding dual $\mathbf{DUAL(\lambda)}$.

**Theorem 22** *Let $\lambda \geq 0$. 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 $r + \lambda = \sum_{j\in B(i,r)} y'_j$. Furthermore, by tripling the radii in the returned set of ROUND($x'$), one can cover all the data points.*

**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 $cost(\mathcal{B}'') + 3 \cdot \lambda \cdot |\mathcal{B}''| \leq 3 \cdot OPT_{\mathbf{LP}(\lambda)}$.

### 3.2.1 Consolidating Solutions from ROUND

To begin our binary search, we need to ensure taking a large value of $\lambda$ will produce $\leq k'$ balls. It is not necessarily clear from ROUND($x'$) that this will happen, so we also employ a consolidation step described in Algorithm 3.

Roughly speaking, it does the following. If some ball from the $k$-CENTER solution $\mathcal{A}$ contains the center of more than one ball from $\mathcal{B}$, then replace all such balls of $\mathcal{B}$ with a single ball that covers all these balls but only if this results in a cheaper solution (in terms of the $r+\lambda$ cost per ball). For a large $\lambda$, this will always be cheaper for any $i \in \mathcal{A}$ that captures the center of more than one ball in $\mathcal{B}$, so at most $k'$ balls will remain.

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

```
B'' ← ∅
for each i' ∈ A do
    Let N_{i'} = {(i, r) ∈ B : i ∈ B(i', 2 · R_m)}
    if 3 · R_m + λ ≤ Σ_{(i,r)∈N_{i'}} (r + λ) then
        B'' ← B'' ∪ {(i', 3 · R_m)}
        B ← B − N_{i'}
B'' ← B'' ∪ B    {Include the balls that were not consolidated in our returned set}
return B''
```

Relevant properties of the algorithm are summarized in the following Lemma.

**Lemma 23** *Let $\lambda \geq 0$. Let $\mathcal{B}$ be the output from ROUND($x'$) where $x'$ is an optimal solution for* $\mathbf{LP}(\lambda)$ *and $y'$ an optimal solution for* $\mathbf{DUAL}(\lambda)$. *Let $\mathcal{B}''$ be the output of CONSOLIDATE($\mathcal{B}, \lambda, \mathcal{A}, R_m$). Then for each $(i, r) \in \mathcal{B}''$, $r \leq 3 \cdot R_m$ and there is some $X_{(i,r)} \subseteq B(i, r)$ such that $r + \lambda \leq \sum_{j \in X_{(i,r)}} y'_j$. Furthermore, for different $(i, r), (i', r') \in \mathcal{B}''$ we have $X_{(i,r)} \cap X_{i',r'} = \emptyset$. Finally, for each $j \in X'$ we have $j \in B(i, 3 \cdot r)$ for some $(i, r) \in \mathcal{B}''$.*

**Proof.** The existence of $X_{(i,r)}$ is simply by setting $X_{(i,r)} = B(i,r)$ if $(i,r) \in \mathcal{B}$ at the end of the algorithm (i.e. if $(i,r)$ is not consolidated into a larger group) or by setting $X_{(i,r)} = \cup_{(i',r') \in N(i)} B_{(i',r')}$ if $(i,r)$ was formed when $i \in \mathcal{A}$ was considered. Since the balls in $\mathcal{B}$ are disjoint at the start of the algorithm and since we remove a ball when it is consolidated, the $X_{(i,r)}$ sets will be disjoint. That $r \leq 3 \cdot R_m$ either follows because $(i,r)$ was in the input (so it is in the support of $x'$), or it is the radius of a new ball which was set to be exactly $3 \cdot R_m$.

We only consolidated balls if $3 \cdot R_M + \lambda \leq \sum_{(i,r) \in N_{i'}} (r + \lambda)$. Since the latter sum is exactly the sum of dual variables for points covered in the balls from $N_{i'}$, the new $r + \lambda$ cost remains bounded by the sum of the dual values of points in $X_{(i,r)}$.

Finally, for each $j \in X'$ if $j \in B(i, 3 \cdot r)$ for a ball that is not consolidated it remains covered by tripling the radii of balls in $\mathcal{B}''$. Otherwise, suppose $(i,r) \in N_{i'}$. Then $d(i',j) \leq d(i',i) + d(i,r) \leq 3 \cdot R_M + 3 \cdot r \leq 3 \cdot R_M + 3 \cdot R_M \leq 6 \cdot R_m$ so certainly $j \in B(i', 3 \cdot (3 \cdot R_m))$. ■

The following shows that extreme values of $\lambda$ will yield solutions with $\geq k'$ and $\leq k'$ balls, which is required at the start of our binary search algorithm.

**Lemma 24** *Consider $\lambda \geq 0$ and an optimal solution $x'$ for* **LP($\lambda$)**. *If $\lambda = 0$ then calling ROUND($x'$) will produce a solution with $|X'| > k'$ balls. If $\lambda = 4 \cdot R_m$ then calling CONSOLIDATE(ROUND($x'$), $\lambda, \mathcal{A}, R_m$) will produce a solution with at most $k'$ balls.*

**Proof.** First consider the case $\lambda = 0$. The LP solution that sets $x_{(i,0)} = 1$ for each $i \in X'$ and all other variables to 0 has value 0. It is also the only optimal solution as supporting any variable corresponding to a ball with positive radius yields an LP solution that has strictly positive cost. So $x'$ only supports balls with radius 0. Since $i' \notin B(i, 0)$ for distinct $i, i' \in X'$ (as we assumed $d(i,i') > 0$), then ROUND($x'$) will return $|X'|$ balls, one per point.

Now consider $\lambda = 4 \cdot R_m$. We claim for any $i' \in \mathcal{A}$, when $i'$ was considered in an iteration we have consolidation would happen if $|N_{i'}| \geq 2$. Note the resulting ball has $r+\lambda$ cost being $3 \cdot R_m + 4 \cdot R_m = 7 \cdot R_m$. On the other hand, the right-hand side of the inequality determining if we perform the consolidation involves at least two separate $\lambda$ values, so the right hand side is at least $8 \cdot R_m$.

So at the end of the loop, each $i' \in \mathcal{A}$ for which consolidation was not performed covers at most one center from $\mathcal{B}$ that was not consolidated. Since each $\mathcal{B}$ that was not consolidated is counted this way (as $\mathcal{A}$ is a $k'$-CENTER solution), we have $|\mathcal{B}''| \leq k'$ at the end of the algorithm. ■

Finally, we require the following routine FILL($\mathcal{B}_1, \mathcal{B}_2$) which will be used to ensure the property from Theorem 17 that all balls in the first solution intersect at least one ball from the second solution.

**Algorithm 4** FILL($\mathcal{B}_1, \mathcal{B}_2$)

**while** Some ball $(i, r) \in \mathcal{B}_1$ is disjoint from balls in $\mathcal{B}_2$ and $|\mathcal{B}_2| < k'$ **do**
  $\mathcal{B}_2 \leftarrow \mathcal{B}_2 \cup \{(i, r)\}$
**if** $|\mathcal{B}_2| = k'$ **then**
  Do the replacement $\mathcal{B}_1 \leftarrow \mathcal{B}_2$
**return** The pair $\mathcal{B}_1, \mathcal{B}_2$

**Lemma 25** *Let $\mathcal{B}_1, \mathcal{B}_2$ be two sets obtained by rounding and, perhaps, consolidating two optimal solutions $x_1, x_2$ to* **LP($\lambda$)** *for some common value $\lambda$. For either of the final sets $\mathcal{B}$ returned by FILL, we have $cost(\mathcal{B}) + \lambda \cdot |\mathcal{B}| \leq OPT_{\mathbf{LP}(\lambda)}$. Furthermore, if $\mathcal{B}'_1, \mathcal{B}'_2$ denotes the pair of returned sets by FILL, then we have $|\mathcal{B}'_1| \geq k' \geq |\mathcal{B}'_2|$. Finally, each $(i, r) \in \mathcal{B}'_1$ intersects at least one ball in $\mathcal{B}'_2$.*

**Proof.** For each $(i, r) \in \mathcal{B}_1$, let $X^1_{(i,r)} \subseteq B(i, r)$ be the $X$-set from Lemma 23, i.e. $r + \lambda \leq \sum_{j \in X^1_{(i,r)}} y'_j$ (or $B(i, r)$ itself if $\mathcal{B}_1$ is just obtained from ROUND($x_1$)). Since $(i, r)$ is disjoint from all balls in $\mathcal{B}_2$ when it is added to $\mathcal{B}_2$, we can let $X^2_{(i,r)}$ be the same as $X^1_{(i,r)}$ and maintain the invariant that balls in $\mathcal{B}_2$ have their $r + \lambda$ value paid for by the dual values of disjoint subsets of points.

That $|\mathcal{B}'_1| \geq k' \geq |\mathcal{B}'_2|$ is immediate, given that the original sets $\mathcal{B}_1, \mathcal{B}_2$ satisfy this bound as well. The final condition that each ball in $\mathcal{B}'_1$ intersect at least one ball in $\mathcal{B}'_2$ is from the fact that either $\mathcal{B}'_1 = \mathcal{B}'_2$ (if $|\mathcal{B}'_2| = k'$) or that the procedure stopped before $|\mathcal{B}'_2|$ became $k'$ (i.e. there are no more balls in the original set $\mathcal{B}_1$ that are disjoint from all balls in $\mathcal{B}'_2$). ■

### 3.2.2 The Binary Search

Recall feasibility of some $x'$ for $\mathbf{LP}(\lambda)$ is not dependent on $\lambda$ itself since the constraints are independent of $\lambda$. We call a value $\lambda > 0$ **smooth** if for some $\epsilon > 0$ we have that the set of optimal extreme points to $\mathbf{LP}(\lambda)$ is the same as the set of optimal extreme points for $\mathbf{LP}(\lambda')$ for any $\lambda' \in [\lambda - \epsilon, \lambda + \epsilon]$. Otherwise, we call $\lambda$ a **break point** (note $\lambda = 0$ is a break point).

We will prove that there is sufficiently large distance between consecutive break points. Our binary search algorithm will proceed until the window is small enough to enclose at most one break point, unless an earlier stopping criteria is met. At this point we can compute the break point itself and then return the required bi-point solution.

**Invariant**: The binary search will maintain values $0 \leq \lambda_1 < \lambda_2$. For each $\ell = 1, 2$, let $x_\ell$ be an optimal solution to $\mathbf{LP}(\lambda_\ell)$ We let $\mathcal{B}_1 = \text{CONSOLIDATE}(\text{ROUND}(x_1), \lambda_1, \mathcal{A}, R_m)$ and $\mathcal{B}_2 = \text{ROUND}(x_2)$. The invariant will also maintain that $|\mathcal{B}_1| \geq k' \geq |\mathcal{B}_2|$ and that $x_1$ is *not* an optimal solution for $\mathbf{LP}(\lambda_2)$.

There are two reasons our binary search may terminate early:

- **Check 1**: For a value $\lambda$, if $x'$ is the corresponding optimal solution to $\mathbf{LP}(\lambda)$ and if $\mathcal{B} := \text{ROUND}(x')$ produces $\geq k'$ balls yet $\mathcal{B}^c := \text{CONSOLIDATE}(\mathcal{B}, \lambda, \mathcal{A}, R_m)$ produces $\leq k'$ balls then we terminate the search and return the two solutions $\mathcal{B}, \mathcal{B}'$ plus $\lambda$. Clearly every ball in $\mathcal{B}$ intersects a ball in $\mathcal{B}^c$ (either it is also in $\mathcal{B}^c$ or it will intersect the ball it was consolidated into). The other properties required by Theorem 17 follow directly from Lemma 23.

- **Check 2**: For a pair $\lambda_1, \lambda_2$, let $x_1, x_2$ be corresponding optimal LP solutions. If $x_1$ is optimal for $\mathbf{LP}(\lambda_2)$ we perform procedure in FILL on ROUND($x_1$) and $\mathcal{B}_2$ (Lemma 25) and return the resulting sets along with $\lambda_2$. Note ROUND($x_1$) will also have size $\geq k'$ since it is the unconsolidated version of $\mathcal{B}_1$. By properties of the balls from Lemma 23 and the 25 (noting $x_1$ is optimal for $\mathbf{LP}(\lambda_2)$), the returned quantities satisfy the properties stated in Theorem 17.

Let $\Delta = 8 \cdot n \cdot n^{4 \cdot n^2}$ be a lower bound on the gap between break points (cf. Lemma 27 belows). Note once $\lambda_2 - \lambda_1 \leq 1/\Delta$, then if Check 2 fails we have that the largest $\lambda$ for which $x_1$ is an optimal solution to $\mathbf{LP}(\lambda)$ satisfies $\lambda \in [\lambda_1, \lambda_2]$. We show how to compute this value $\lambda$ exactly in Lemma 26. We can now state our binary search.

**Algorithm 5** Binary Search to Find the Bipoint Solution

```
λ1 ← 0, λ2 ← 4 · R_m and corresponding x1, B1, x2, B2 as in the invariant
if Check 1 on either λ1 or λ2 or Check 2 on (λ1, λ2) passes then
    return the corresponding solution
while λ1 + 1/Δ < λ2 do
    Let λ ← (λ1 + λ2)/2, x' is optimal to LP(λ), and B, B^c as in Check 1.
    if Check 1 on λ passes, return the corresponding solution
    if |B^c| ≥ k' then
        λ1, x1, B1 ← λ, x', B^c
    else
        λ2, x2, B2 ← λ, x', B
    if Check 2 on (λ1, λ2) passes, return the corresponding solution
Compute the only breakpoint λ ∈ [λ1, λ2], let x', B, B^c be as in Check 1 for this λ
if Check 1 on λ passes, return the corresponding solution
if |B^c| ≥ k' then
    perform FILL(B^c, B2) and return the resulting sets along with λ
else
    perform FILL(ROUND(x1), B) and return the resulting sets along with λ
```

After the initial values $\lambda_1 = 0$ and $\lambda_2 = 4 \cdot R_m$ are set, if the checks all fail then the invariant is initially true. In each step of the search, if the checks fail then it is easy to see the invariant continues to hold.

If the loop terminates without returning, the final the returned sets and $\lambda$-value have the properties stated in Theorem 17. This immediately follows from Lemmas

23, 25, and the fact that $x_1$ and $x_2$ are both optimal for $\mathbf{LP}(\lambda)$ as $\lambda$ is the only break point in $[\lambda_1, \lambda_2]$. Though we should point out that if final **else** was reached, then $|\mathcal{B}| < k'$ or else Check 1 on $\lambda$ would have passed. Also, $|\text{ROUND}(x_1)| \geq k'$ since $\mathcal{B}_1$ does (i.e. the consolidated version is larger than $k'$ so the unconsolidated must be as well). The reason we use ROUND($x_1$) instead of the consolidated version $\mathcal{B}_1$ is because ROUND does not depend on $\lambda$ whereas $\mathcal{B}_1$ does and we might end up with different consolidations if we did the consolidation with respect to $\lambda$ instead of $\lambda_1$. So, $|\text{ROUND}(x_1)| \geq k' \geq |\mathcal{B}|$, as required. So in $O(\log(4 \cdot R_m \cdot \Delta)) = O(\log R_m + n^2 \log n)$ iterations, which is polynomial in the input size, the binary search will return a bipoint solution satisfying the properties stated in Theorem 17.

### Supporting Results for the Binary Search

**Lemma 26** *Let $x'$ be an optimal solution for $\mathbf{LP}(\lambda_1)$. In polynomial time, we can compute the greatest $\lambda$ such that $x'$ remains optimal for $\mathbf{LP}(\lambda)$.*

**Proof.** Consider the following LP for this fixed value of $x'$ but having $\lambda$ as a variable and variables $y_j, j \in X'$ as in $\mathbf{DUAL}(\lambda)$.

$$\begin{array}{rrcll} \textbf{maximize}: & \lambda & & & \\ \textbf{subject to}: & \sum_{j \in B(i,r) \cap X'} y_j & \geq & r + \lambda & \forall\ (i,r) \text{ s.t. } r \leq R_m \\ & \sum_{j \in X'} y_j & = & \sum_{(i,r): r \leq R_m} x' \cdot (r + \lambda) & \\ & y, \lambda & \geq & 0 & \end{array}$$

We emphasize that $x'$ is a fixed value in this setting, so the second constraint is linear in the variables $y$ and $\lambda$.

The first and third constraints assert $y$ is a feasible dual solution for the particular $\lambda$. The second asserts its value in $\mathbf{DUAL}(\lambda)$ is equal to the the value of $x_1$ in $\mathbf{LP}(\lambda)$. Thus, $x'$ is optimal for $\mathbf{LP}(\lambda)$ exactly if there is some corresponding $y$ that causes all of the constraints of the above LP to hold. So solving this LP will yield the maximum $\lambda$ such that $x'$ is an optimal solution for $\mathbf{LP}(\lambda)$. ■

Before we begin the next proof, we recall Hadamard's bound on the determinant of a matrix in terms of the lengths of its row vectors. It is a bound on the determinant of a matrix whose entries are complex numbers. In that case, if $v_i$ notes the $i$-th column of the real-valued $n \times n$ matrix $N$, then we will have:

$$|det(N)| \leq \prod_{i=1}^{n} ||v_i||_2$$

**Lemma 27** *For two different break points $\lambda < \lambda'$, we have $\lambda + 1/\Delta < \lambda'$ where $\Delta = 8 \cdot n^2 \cdot n^{4 \cdot n^2}$.*

**Proof.** Let $x$ be any extreme point solution of the polytope defining $\mathbf{LP(\lambda)}$. So $x$ is the unique solution to a $M \cdot x = b$ where $M$ is an $n \times n$ non-singular submatrix of the constraint matrix (here, $n = |X'|$). From Cramer's rule, the denominator of each variable is bounded by $|\det(M)|$. Since the constraint matrix only has entries 0 and 1, each row $M_j$ satisfies $||M_j||_2 \leq \sqrt{n}$. By Hadamard's determinant bound, $|\det(M)| \leq \prod_j ||M_j||_2 \leq n^{n/2}$. Thus, every denominator in $x$ is an integer at most $n^{n/2}$.

Note $\lambda' > 0$. Let $\lambda''$ be very close to $\lambda'$ such that some extreme point $x'$ that is optimal for $\mathbf{LP}(\lambda')$ is not optimal for $\mathbf{LP}(\lambda'')$. This must be the case, it could not be that there is an extreme point that is optimal for $\lambda''$ arbitrarily close to $\lambda'$ but not for $\lambda'$ itself since the set of $\lambda''$ for which a particular $x$ is an optimal solution is a closed set. Let $x''$ be an optimal extreme point for $\mathbf{LP}(\lambda'')$, which then must be optimal for $\mathbf{LP}(\lambda')$ as well.

Define a linear function $f'(z) = \sum_{(i,r)}(z + r) \cdot x'_{(i,r)}$ and similarly define $f''(z) = \sum_{(i,r)}(z + r) \cdot x''_{(i,r)}$. Then $f'(\lambda') = f''(\lambda')$ but $f'(\lambda'') \neq f''(\lambda'')$ so they have different slopes. That is, $f'(z) = f''(z)$ has a unique solution, namely at $z = \lambda' = \frac{\sum_{(i,r)} r \cdot (x''_{(i,r)} - x'_{(i,r)})}{\sum_{(i,r)} x'_{(i,r)} - x''_{(i,r)}}$. Note each of $x'$ and $x''$ supports at most $n$ values since they are extreme points of a polytope with only $n$ constraints apart from nonnegativity. So the top term in the ratio above expressing $\lambda'$ is a fraction of the form $N/D$ where $D \leq n^{n/2 \cdot 2n} = n^{n^2}$. Similarly, the bottom term of the ratio for $\lambda'$ is a fraction of the form $N'/D'$ where $N' \leq 2n \cdot n^{n^2}$ (using the fact that all $x'$ and $x''$ values are $\leq 1$). Thus, $\lambda'$ itself is a fraction whose denominator is at most $2n \cdot n^{2 \cdot n^2}$.

Finally, since $\lambda$ and $\lambda'$ are different break points, then $\lambda' - \lambda$ is a fraction whose denominator is at most $4n^2 \cdot n^{4 \cdot n^2} = \Delta/2$. Thus, $\lambda + 1/\Delta < \lambda'$. ■

# Chapter 4

# Conclusion

## 4.1 Conclusion and Future Considerations

Foremost, it may be possible to further refine our study by taking into account a more involved strategy for determining how to best cover a group with a single ball. But it appears likely that such a method will result in approximations that are still a constant-factor worse than 3. On the other hand, observation regarding our novel method of locating the bipoint solution is very intriguing. If we ever encounter a $\lambda$ 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?

As mentioned before, there exists a QPTAS for MSR, and the question begs to be asked, *Is there a PTAS for the MSR problem?* Note that we already know that one cannot approximate $MSD$ with a factor better than 2, so even though we showed a separation between approximating the two related problem, where is the line that MSR and MSD also differ in hardness, if they do? Closing the gap between 2 and our result for MSD appears to require methodology better than using a rounding that imposes a 3-approximation factor on the problem itself.

Other paths that we discuss during our work, involve getting the rounding step approximation factor to better than 3. That is to construct a better than 3, likely 2, approximation for MSR's LMP. This is specifically motivated given the current 2-approximation for $k$-centre [5].

Considering the same problem in the Doubling Dimension and Euclidean space is the next step in our study. One can use the same arguments of choosing a centre or fixating a centre to tackle both MSR and MSD in these metrics. It is likely that getting a PTAS for MSR in doubling metrics would be easier than getting a PTAS in general. It is already known there is an exact algorithm in low-dimensional Euclidean metrics [17], so doubling properties come in helpful. Also, in high-dimensional Euclidean spaces we do have that the diameter of a set is at most $1/\sqrt{2}$ times its radius which may lead to better approximations.

# Bibliography


[1] S. Dasgupta, "The hardness of k-means clustering," 2008.

[2] D. Aloise, A. Deshpande, P. Hansen, and P. Popat, "Np-hardness of euclidean sum-of-squares clustering," *Mach. Learn.*, vol. 75, no. 2, pp. 245–248, 2009.

[3] M. Mahajan, P. Nimbhorkar, and K. Varadarajan, "The planar k-means problem is np-hard," in *WALCOM: Algorithms and Computation*, Springer Berlin Heidelberg, 2009.

[4] M. E. Dyer and A. M. Frieze, "A simple heuristic for the p-centre problem," *Operations Research Letters*, vol. 3, no. 6, pp. 285–288, 1985.

[5] D. S. Hochbaum and D. B. Shmoys, "A best possible heuristic for the k-center problem," *Mathematics of Operations Research*, vol. 10, no. 2, pp. 180–184, 1985.

[6] 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*, vol. 48, no. 2, 274–296, 2001.

[7] 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)*, 1999, pp. 378–388.

[8] M. Charikar, S. Guha, Tardos, and D. B. Shmoys, "A constant-factor approximation algorithm for the k-median problem," *Journal of Computer and System Sciences*, vol. 65, no. 1, pp. 129–149, 2002.

[9] O. Kariv and S. L. Hakimi, "An algorithmic approach to network location problems. ii: The p-medians," *SIAM Journal on Applied Mathematics*, vol. 37, no. 3, pp. 539–560, 1979.

[10] J. H. Lin and J. S. Vitter, "Approximation algorithms for geometric median problems," *Information Processing Letters*, vol. 44, no. 5, pp. 245–249, 1992.

[11] J. Byrka, T. Pensyl, B. Rybicki, A. Srinivasan, and K. Trinh, "An improved approximation for k-median and positive correlation in budgeted optimization," *ACM Trans. Algorithms*, vol. 13, no. 2, 2017.

[12] G. B. Dantzig and M. N. Thapa, "Linear programming," New York, NY, USA: Springer, 1997.

[13] L. Khachiyan, “Polynomial algorithms in linear programming,” *USSR Computational Mathematics and Mathematical Physics*, vol. 20, no. 1, pp. 53–72, 1980.

[14] N. Karmarkar, “A new polynomial-time algorithm for linear programming,” ser. STOC ’84, New York, NY, USA: Association for Computing Machinery, 1984, 302–311.

[15] B. Korte and J. Vygen, “Linear programming,” in *Combinatorial Optimization: Theory and Algorithms*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 51–71.

[16] L. C. Lau, R. Ravi, and M. Singh, *Iterative Methods in Combinatorial Optimization* (Cambridge Texts in Applied Mathematics). Cambridge University Press, 2011.

[17] M. Gibson, G. Kanade, E. Krohn, I. A. Pirwani, and K. Varadarajan, “On metric clustering to minimize the sum of radii,” *Algorithmica*, vol. 57, no. 3, pp. 484–498, 2009.

[18] M. Charikar and R. Panigrahy, “Clustering to minimize the sum of cluster diameters,” *Journal of Computer and System Sciences*, vol. 68, no. 2, pp. 417–441, 2004, Special Issue on STOC 2001.

[19] 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)*, ser. STOC ’06, Seattle, WA, USA: Association for Computing Machinery, 2006, 721–729.

[20] J. Batra, N. Garg, A. Kumar, T. Momke, 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)*, 2015, pp. 47–58.

[21] F. Grandoni, T. Momke, 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*, ser. STOC 2018, New York, NY, USA: Association for Computing Machinery, 2018, 607–619.

[22] F. Grandoni, T. Momke, and A. Wiese, “Unsplittable flow on a path: The game!” In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2022, pp. 906–926.

[23] F. Grandoni, T. Momke, 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)*, 2017, pp. 2411–2422.

[24] F. Grandoni, T. Momke, 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.

[25] 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)*, Springer-Verlag, 2000, 237–250.

[26] M. Gibson, G. Kanade, E. Krohn, I. A. Pirwani, and K. Varadarajan, "On clustering to minimize the sum of radii," *SIAM Journal on Computing*, vol. 41, no. 1, pp. 47–60, 2012.

[27] V. Capoyleas, G. Rote, and G. Woeginger, *Geometric clusterings*, 1990.

[28] P. Hansen and B. Jaumard, "Minimum sum of diameters clustering," *Journal of Classification*, vol. 4, no. 2, pp. 215–226, Sep. 1987.

[29] B. Behsaz and M. R. Salavatipour, "On minimum sum of radii and diameters clustering," *Algorithmica*, vol. 73, no. 1, 143–165, 2015.

[30] P. Heggernes and D. Lokshtanov, "Optimal broadcast domination in polynomial time," *Discrete Mathematics*, vol. 306, no. 24, pp. 3267–3280, 2006.

[31] S. Ahmadian and C. Swamy, "Approximation Algorithms for Clustering Problems with Lower Bounds and Outliers," in *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, I. Chatzigiannakis, M. Mitzenmacher, Y. Rabani, and D. Sangiorgi, Eds., ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 55, Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016, 69:1–69:15.

[32] S. Bhowmick, T. Inamdar, and K. R. Varadarajan, "Fault-tolerant covering problems in metric spaces," *Algorithmica*, vol. 83, pp. 413–446, 2021.

[33] A. K. Abu-Affash, P. Carmi, M. J. Katz, and G. Morgenstern, "Multi cover of a polygon minimizing the sum of areas," in *WALCOM: Algorithms and Computation*, N. Katoh and A. Kumar, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 134–145.

[34] S. Bhowmick, K. Varadarajan, and S.-K. Xue, "A constant-factor approximation for multi-covering with disks," ser. SoCG '13, Rio de Janeiro, Brazil: Association for Computing Machinery, 2013, 243–248.

[35] X. Liu, W. Li, and R. Xie, "A primal-dual approximation algorithm for the k-prize-collecting minimum power cover problem," *Optimization Letters*, Nov. 2021.

[36] F. Chudak and D. B. Shmoys, "Improved approximation algorithms for the uncapacitated facility location problem," *SIAM Journal on Computing*, vol. 33, no. 1, pp. 1–25, 2003.

[37] F. Chudak. and D. P. Williamson, "Improved approximation algorithms for capacitated facility location problems," in *Proceedings of the 7th International IPCO Conference on Integer Programming and Combinatorial Optimization*, Springer-Verlag, 1999, 99–113.

[38] S. Guha and S. Khuller, "Greedy strikes back: Improved facility location algorithms," *Journal of Algorithms*, vol. 31, no. 1, pp. 228–248, 1999.

[39] M. R. Korupolu, C. G. Plaxton, and R. Rajaraman, "Analysis of a local search heuristic for facility location problems," *Journal of Algorithms*, vol. 37, no. 1, pp. 146–188, 2000.

[40] C. L. Monma and S. Suri, "Partitioning points and graphs to minimize the maximum or the sum of diameters," *Graph Theory, Combinatorics, and Applications*, pp. 880–912, 1991.

[41] D. P. Williamson, "The primal-dual method for approximation algorithms," *Mathematical Programming*, vol. 91, no. 3, pp. 447–478, Feb. 2002.

[42] S. Bandyapadhyay and K. Varadarajan, "Approximate Clustering via Metric Partitioning," in *27th International Symposium on Algorithms and Computation (ISAAC 2016)*, S.-H. Hong, Ed., ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 64, Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016, 15:1–15:13.

[43] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit, "Local search heuristic for k-median and facility location problems," in *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*, ser. STOC '01, Hersonissos, Greece: Association for Computing Machinery, 2001, 21–29.

[44] J. HADAMARD, "Resolution d'une question relative aux determinants," *Bull. des Sciences Math.*, vol. 2, pp. 240–246, 1893.

# Appendix A: Optimizing our choice of parameter for the MSR analysis

The MSR analysis included a variety of cases and certain constants were chosen to define these cases. Here, we show that our choices of constants are optimal for our analysis techniques.

Let $\delta$ be the ratio of $R_3$ to $R_2$. We wish to transform radius of the cluster picked at each center to be similar to other cases. In other words, we wish to transform each radius into a term like $(1+\beta)\cdot R_1 + (4-\alpha)\cdot R_2$, a term similar to the radius of case of picking $i_1$, but with less weight on $R_2$ and more on $R_1$. After finding the values to $\alpha$ and $\beta$ in terms of $\delta$, we make a decision for the center to pick based on value of $\frac{R_3}{R_2}$:

1. soln*(1): Picking the center based on the value of $R_2$ according to table $A$.1 if $\frac{R_3}{R_2} \leq \delta$.

2. soln*(2): Picking the center at $i_1$ if $\frac{R_3}{R_2} \geq \delta$.

Based on value of $\alpha$ and $\beta$, we can then find the parametric approximation factor for soln*(1) parameterized by $\delta$, which we do in what follows. But first, let us calculate the approximation factor for soln*(2) using the following inequality:

$$R_2 + R_3 \leq \mathit{Cost}(G_{i_1,R_1}) \Longrightarrow (1+\delta)\cdot R_2 \leq \mathit{Cost}(G_{i_1,R_1}) \Longrightarrow R_2 \leq (\frac{1}{1+\delta})\mathit{Cost}(G_{i_1,R_1}).$$

Then, the cost of a group is $R_1 + (\frac{4}{1+\delta})\cdot \mathit{Cost}(G_{i_1,R_1}) = \mathit{Cost}(B(i_1,R_1)) + (\frac{4}{1+\delta})\cdot \mathit{Cost}(G_{i_1,R_1})$ as $\mathit{Cost}(B(i_1,R_1)) = R_1$. For the sake of brevity, let $\mathit{Cost}(G_{i_1,R_1}) = C(G_{i_1,R_1})$ and $\mathit{Cost}(B(i_1,R_1)) = C(B(i_1,R_1))$.

Now, let us restate the cost of each scenario in terms of $\alpha$ and $\delta$, so that we can minimize the cost after aggregating the costs over all groups.

*Case 1 ($R_2 \geq R_1 + 2 \cdot \delta \cdot R_2$):*

- Cost of picking $i_1$: $R_1 + 4 \cdot R_2$.
- Cost of picking $i_2$: $3 \cdot R_2$ since $R_2 \geq R_1 + 2 \cdot \delta \cdot R_2 \Longrightarrow 3R_2 \geq 2R_1 + (1 + 4 \cdot \delta)R_2$.
- Cost of picking $i_3$: $4R_2$ since $R_2 \geq R_1 + 2 \cdot \delta \cdot R_2 \Longrightarrow 2R_2 \geq 2R_1 + 4 \cdot \delta R_2 \Longrightarrow 4R_2 \geq 2R_1 + 4 \cdot \delta R_2$.

Hence, we pick $i_2$ with minimum cost.

*Case 2 ($R_1 + 2 \cdot \delta \cdot R_2 \geq R_2 \geq \frac{2}{3} \cdot R_1 + \frac{4}{3} \cdot \delta \cdot R_2$):*

- Cost of picking $i_1$: $R_1 + 4 \cdot R_2$.
- Cost of picking $i_2$: $2 \cdot R_1 + (1 + 4 \cdot \delta) \cdot R_2$ since $R_2 \leq R_1 + 2 \cdot \delta \cdot R_2 \Longrightarrow 3R_2 \leq 2R_1 + (1 + 4 \cdot \delta)R_2$.
- Cost of picking $i_3$: $4R_2$ since $R_2 \geq \frac{2}{3} \cdot R_1 + \frac{4}{3} \cdot \delta \cdot R_2 \Longrightarrow 3R_2 \geq 2 \cdot R_1 + 4 \cdot \delta R_2 \Longrightarrow 4R_2 \geq 2R_1 + 4 \cdot \delta R_2$.

Note that $\frac{2}{3} \cdot R_1 + \frac{4}{3} \cdot \delta \cdot R_2 \leq R_2 \Longrightarrow 2 \cdot R_1 + (1 + 4 \cdot \delta) \cdot R_2 \leq 4 \cdot R_2$. Also, we have $\frac{2}{3} \cdot R_1 + \frac{4}{3} \cdot \delta \cdot R_2 \leq R_2 \Longrightarrow \frac{1}{3} \cdot R_1 + \frac{4}{3} \cdot \delta \cdot R_2 \leq R_2 \longrightarrow R_1 + 4 \cdot \delta \cdot R_2 \leq 3 \cdot R_1 \Longrightarrow 2 \cdot R_1 + (1 + 4 \cdot \delta) \cdot R_2 \leq R_1 + 4 \cdot R_1$. Hence, we pick $i_2$ with minimum cost.

Since $R_2 \geq \frac{2}{3} \cdot R_1 + \frac{4 \cdot \delta}{3} \cdot R_2$, then

$$\frac{-4 \cdot \delta + 3 - \alpha}{1 - \frac{4}{3} \cdot \delta}(R_2 \cdot (1 - \frac{4}{3} \cdot \delta) - \frac{2}{3} \cdot R_1) \geq 0,$$

where $\delta < \frac{3}{4}$ and $3 \geq 4 \cdot \delta + \alpha$, for some value of $\alpha$. Since we pick the center $i_2$, the cost is $2 \cdot R_1 + (1 + 4 \cdot \delta) \cdot R_2$. At last,

$$\text{Cost } \leq 2 \cdot R_1 + (1 + 4 \cdot \delta)R_2 + \frac{-4 \cdot \delta + 3 - \alpha}{1 - \frac{4}{3} \cdot \delta}(R_2 \cdot (1 - \frac{4}{3} \cdot \delta) - \frac{2}{3} \cdot R_1) = (\frac{2 \cdot \alpha}{3 - 4 \cdot \delta})R_1 + (4 - \alpha) \cdot R_2.$$

*Case 3 ($\frac{2}{3} \cdot R_1 + \frac{4}{3} \cdot \delta \cdot R_2 \geq R_2 \geq \frac{1}{2} \cdot R_1 + \delta \cdot R_2$):*

- Cost of picking $i_1$: $R_1 + 4 \cdot R_2$.

- Cost of picking $i_2$: $2 \cdot R_1 + (1 + 4 \cdot \delta) \cdot R_2$ since $R_2 \leq R_1 + 2 \cdot \delta \cdot R_2 \Longrightarrow 3R_2 \leq 2R_1 + (1 + 4 \cdot \delta)R_2$.

- Cost of picking $i_3$: $4R_2$ since $R_2 \geq \frac{2}{3} \cdot R_1 + \frac{4}{3} \cdot \delta \cdot R_2 \Longrightarrow 3R_2 \geq 2 \cdot R_1 + 4 \cdot \delta R_2 \Longrightarrow 4R_2 \geq 2R_1 + 4 \cdot \delta R_2$.

Note that $\frac{2}{3} \cdot R_1 + \frac{4}{3} \cdot \delta \cdot R_2 \geq R_2 \Longrightarrow 2 \cdot R_1 + (1 + 4 \cdot \delta) \cdot R_2 \geq 4 \cdot R_2$ Hence, we pick $i_3$ with minimum cost.

Since $R_2 \leq \frac{2}{3} \cdot R_1 + \frac{4 \cdot \delta}{3} \cdot R_2$, then

$$\frac{\alpha}{1 - \frac{4}{3} \cdot \delta}(R_2 \cdot (\frac{4}{3} \cdot \delta - 1) + \frac{2}{3} \cdot R_1) \geq 0,$$

where $\delta < \frac{3}{4}$ and $\alpha \geq 0$, for some value of $\alpha$. Since we pick the center $i_2$, the cost is $4 \cdot R_2$. At last,

$$\text{Cost } \leq 4 \cdot R_2 + \frac{\alpha}{1 - \frac{4}{3} \cdot \delta}(R_2 \cdot (\frac{4}{3} \cdot \delta - 1) + \frac{2}{3} \cdot R_1) = (\frac{2 \cdot \alpha}{3 - 4 \cdot \delta})R_1 + (4 - \alpha) \cdot R_2.$$

*Case 4 ($\frac{1}{2} \cdot R_1 + \delta \cdot R_2 \geq R_2 \geq \frac{1}{4} \cdot R_1 + \delta \cdot R_2$):*

- Cost of picking $i_1$: $R_1 + 4 \cdot R_2$.

- Cost of picking $i_2$: $2 \cdot R_1 + (1 + 4 \cdot \delta) \cdot R_2$ since $R_2 \leq R_1 + 2 \cdot \delta \cdot R_2 \Longrightarrow 3R_2 \leq 2R_1 + (1 + 4 \cdot \delta)R_2$.

- Cost of picking $i_3$: $2R_1 + 4 \cdot \delta \cdot R_2$ since $R_2 \leq \frac{1}{2} \cdot R_1 + \delta \cdot R_2 \Longrightarrow 4R_2 \geq 2 \cdot R_1 + 4 \cdot \delta R_2$.

We have $\frac{1}{4} \cdot R_1 + \delta \cdot R_2 \leq R_2 \Longrightarrow R_1 + 4 \cdot \delta \cdot R_2 \leq 4 \cdot R_2 \longrightarrow 2 \cdot R_1 + 4 \cdot \delta \cdot R_2 \leq R_1 + 4 \cdot R_2$. Hence, we pick $i_3$ with minimum cost.

Since $R_2 \geq \frac{1}{4} \cdot R_1 + \delta \cdot R_2$, then

$$\frac{4 - 4 \cdot \delta - \alpha}{1 - \delta}(R_2 \cdot (1 - \delta) - \frac{1}{4} \cdot R_1) \geq 0,$$

where $4 \geq 4 \cdot \delta + \alpha$ and $\alpha \geq 0$, for some value of $\alpha$. Since we pick the center $i_3$, the cost is $2 \cdot R_1 + 4 \cdot \delta \cdot R_2$. At last,

$$\text{Cost } \leq 2 \cdot R_1 + 4 \cdot \delta \cdot R_2 + \frac{4 - 4 \cdot \delta - \alpha}{1 - \delta}(R_2 \cdot (1 - \delta) - \frac{1}{4} \cdot R_1) = (1 + \frac{\alpha}{4 - 4 \cdot \delta})R_1 + (4 - \alpha) \cdot R_2.$$

*Case 5 ($\frac{1}{4} \cdot R_1 + \delta \cdot R_2$):*

- Cost of picking $i_1$: $R_1 + 4 \cdot R_2$.
- Cost of picking $i_2$: $2 \cdot R_1 + (1 + 4 \cdot \delta) \cdot R_2$ since $R_2 \leq R_1 + 2 \cdot \delta \cdot R_2 \Longrightarrow 3R_2 \leq 2R_1 + (1 + 4 \cdot \delta)R_2$.
- Cost of picking $i_3$: $2R_1 + 4 \cdot \delta \cdot R_2$ since $R_2 \leq \frac{1}{2} \cdot R_1 + \delta \cdot R_2 \Longrightarrow 4R_2 \geq 2 \cdot R_1 + 4 \cdot \delta R_2$.

We have $\frac{1}{4} \cdot R_1 + \delta \cdot R_2 \geq R_2 \Longrightarrow R_1 + 4 \cdot \delta \cdot R_2 \geq 4 \cdot R_2 \longrightarrow 2 \cdot R_1 + 4 \cdot \delta \cdot R_2 \geq R_1 + 4 \cdot R_2$. Hence, we pick $i_1$ with minimum cost.

Since $R_2 \leq \frac{1}{4} \cdot R_1 + \delta \cdot R_2$, then

$$\frac{\alpha}{1 - \delta}(R_2 \cdot (\delta - 1) + \frac{1}{4} \cdot R_1) \geq 0,$$

where $\delta < 1$ and $\alpha \geq 0$, for some value of $\alpha$. Since we pick the center $i_1$, the cost is $R_1 + 4 \cdot R_2$. At last,

$$\text{Cost } \leq R_1 + 4 \cdot R_2 + \frac{\alpha}{1 - \delta}(R_2 \cdot (\delta - 1) + \frac{1}{4} \cdot R_1) = (1 + \frac{\alpha}{4 - 4 \cdot \delta})R_1 + (4 - \alpha) \cdot R_2.$$

| Case # | $\frac{R_3}{R_2} < \delta$ | $\frac{R_3}{R_2} \geq \delta$ |
|---|---|---|
| 1 | $3 \cdot C(G_{i_1,R_1})$ | $\frac{3}{1+\delta} \cdot C(G_{i_1,R_1})$ |
| 2 | $\frac{2 \cdot \alpha}{3-4 \cdot \delta} \cdot C(B(i_1, R_1)) + (4 - \alpha) \cdot C(G_{i_1,R_1})$ | $C(B(i_1, R_1)) + \frac{4}{1+\delta} \cdot C(G_{i_1,R_1})$ |
| 3 | $\frac{2 \cdot \alpha}{3-4 \cdot \delta} \cdot C(B(i_1, R_1)) + (4 - \alpha) \cdot C(G_{i_1,R_1})$ | $C(B(i_1, R_1)) + \frac{4}{1+\delta} \cdot C(G_{i_1,R_1})$ |
| 4 | $(1 + \frac{\alpha}{4-4 \cdot \delta}) \cdot C(B(i_1, R_1)) + (4 - \alpha) \cdot C(G_{i_1,R_1})$ | $C(B(i_1, R_1)) + \frac{4}{1+\delta} \cdot C(G_{i_1,R_1})$ |
| 5 | $(1 + \frac{\alpha}{4-4 \cdot \delta}) \cdot C_1^g + (4 - \alpha) \cdot C(G_{i_1,R_1})$ | $C(B(i_1, R_1)) + \frac{4}{1+\delta} \cdot C(G_{i_1,R_1})$ |

Table A.1: The cost of our choice for centers in every condition discussed.

As per Table A.1, and the conditions for $\alpha$ and $\delta$ in every case, set $\alpha = 1$ and $\delta = \frac{1}{3}$.

Then, the cost of each group is as in Table A.2 based on our decision: Note that each

| Case # | $\frac{R_3}{R_2} < \frac{1}{3}$ | $\frac{R_3}{R_2} \geq \frac{1}{3}$ |
|---|---|---|
| 1 | $3 \cdot C(G_{i_1,R_1})$ | $\frac{9}{4} \cdot C(G_{i_1,R_1})$ |
| 2 | $\frac{6}{5} \cdot C(B(i_1, R_1)) + (3) \cdot C(G_{i_1,R_1})$ | $C(B(i_1, R_1)) + 3 \cdot C(G_{i_1,R_1})$ |
| 3 | $\frac{6}{5} \cdot C(B(i_1, R_1)) + (3) \cdot C(G_{i_1,R_1})$ | $C(B(i_1, R_1)) + 3 \cdot C(G_{i_1,R_1})$ |
| 4 | $(\frac{11}{8}) \cdot C(B(i_1, R_1)) + (3) \cdot C(G_{i_1,R_1})$ | $C(B(i_1, R_1)) + 3 \cdot C(G_{i_1,R_1})$ |
| 5 | $(\frac{11}{8}) \cdot C(B(i_1, R_1)) + (3) \cdot C(G_{i_1,R_1})$ | $C(B(i_1, R_1)) + 3 \cdot C(G_{i_1,R_1})$ |

Table A.2: The cost of our choice for centers in every condition discussed.

group's cost is in form of $A \cdot C(B(i_1, R_1)) + B \cdot C(G_{i_1,R_1})$ where the maximum values for $A$ and $B$ happens when $\frac{R_3}{R_2} < \frac{1}{3}$ and $R_2 \leq \frac{R_1}{2} + \frac{R_2}{3}$ with cost $\frac{11}{8} \cdot C(B(i_1, R_1)) + 3 \cdot C(G_{i_1,R_1})$.

# Appendix B: Optimizing our choice of parameter for the MSSR analysis

We have summarized the case, the condition, and the centre picked in Table . Then,

| Case # | Condition | Centre picked |
|---|---|---|
| 1 | $R_2 \geq R_1 + 2\delta R_2$ | $i_2$ |
| 2 | $R_1 + 2\delta R_2 \geq R_2 \geq \frac{2}{3} \cdot R_1 + \frac{4}{3}\delta \cdot R_2$ | $i_2$ |
| 3 | $\frac{2}{3} \cdot R_1 + \frac{4}{3}\delta \cdot R_2 \geq R_2 \geq \frac{1}{2} \cdot R_1 + \delta \cdot R_2$ | $i_3$ |
| 4 | $\frac{1}{2} \cdot R_1 + \delta \cdot R_2 \geq R_2 \geq \frac{1}{4} \cdot R_1 + \delta \cdot R_2$ | $i_3$ |
| 5 | $\frac{1}{2} \cdot R_1 + \delta \cdot R_2 \geq R_2 \geq \frac{1}{4} \cdot R_1 + \delta \cdot R_2$ | $i_1$ |

Table B.1: The centres that will be picked in the Minimum Sum of Squared Radii based on values of $R_1$ and $R_2$ as depicted in 2.1

Table describes the upper-bound on the cost of each choice, using the upper-bound value for $R_2$. The main difficulty in this scenario is getting rid of $R_1R_2$ terms in the costs for each case.

*Case 1, $R_2 \geq R_1 + 2\delta R_2$:* Cost of picking $i_2$ will be $(3 \cdot R_2)^2 = 9 \cdot R_2^2$. We aim to balance the cost of other cases such that the upper-bound of each has one term $9 \cdot R_2^2$ and a term with some coefficient for $R_1^2$.

*Case 2, $R_1 + 2\delta R_2 \geq R_2$:* We will be using the bound $R_2 \leq \frac{1}{1-2\delta} R_1$. Note that $\delta$ is the ratio $\frac{R_3}{R_2}$. Then we must have that $\delta < \frac{1}{2}$. Then the cost is bounded by $4 + \frac{32 \cdot \delta^2 - 16 \cdot \delta - 5}{(1-2\delta)^2} R_1^2 + 9R_2^2$.

*Case 3,* $\frac{2}{3} \cdot R_1 + \frac{4}{3}\delta \cdot R_2$*:* We will be using the bound $R_2 \leq \frac{2}{3-4\delta} R_1$. Then we must have that $\delta < \frac{3}{4}$. At last, the cost is bounded by $7 \cdot \frac{2}{3-4\cdot\delta} R_1^2 + 9R_2^2$.

*Case 4,* $R_1 + 2\delta R_2 \geq R_2$*:* We will be using the bound $R_2 \leq \frac{1}{2-2\delta} R_1$. Then we must have that $\delta < 1$. Then the cost is bounded by $4 + \frac{-16\cdot\delta^2+32\delta-9}{(2-2\delta)^2} R_1^2 + 9R_2^2$.

*Case 5,* $\frac{1}{2} \cdot R_1 + \delta \cdot R_2 \geq R_2$*:* We will be using the bound $R_2 \leq \frac{1}{4-4\delta} R_1$. Then we must have that $\delta < 1$. At last, the cost is bounded by $\frac{39-32\cdot\delta}{(4-4\cdot\delta)^2} R_1^2 + 9R_2^2$.

| Case # | Condition | Centre picked |
|---|---|---|
| 1 | $R_2 \geq R_1 + 2\delta R_2$ | $i_2$ |
| 2 | $R_1 + 2\delta R_2 \geq R_2 \geq \frac{2}{3} \cdot R_1 + \frac{4}{3}\delta \cdot R_2$ | $i_2$ |
| 3 | $\frac{2}{3} \cdot R_1 + \frac{4}{3}\delta \cdot R_2 \geq R_2 \geq \frac{1}{2} \cdot R_1 + \delta \cdot R_2$ | $i_3$ |
| 4 | $\frac{1}{2} \cdot R_1 + \delta \cdot R_2 \geq R_2 \geq \frac{1}{4} \cdot R_1 + \delta \cdot R_2$ | $i_3$ |
| 5 | $\frac{1}{2} \cdot R_1 + \delta \cdot R_2 \geq R_2 \geq \frac{1}{4} \cdot R_1 + \delta \cdot R_2$ | $i_1$ |

Table B.2: The centres that will be picked in the Minimum Sum of Squared Radii based on values of $R_1$ and $R_2$ as depicted in 2.1