Parameterized Approximation Algorithms for Sum of Radii Clustering and Variants

Xianrun Chen[1, 2], **Dachuan Xu**[3], **Yicheng Xu**[1, 2, *], **Yong Zhang**[1, 2]

[1] Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China
[2] University of Chinese Academy of Sciences, Beijing, China
[3] Beijing University of Technology, Beijing, China

Abstract

Clustering is one of the most fundamental tools in artificial intelligence, machine learning, and data mining. In this paper, we follow one of the recent mainstream topics of clustering, Sum of Radii (SoR), which naturally arises as a balance between the folklore k-center and k-median. SoR aims to determine a set of k balls, each centered at a point in a given dataset, such that their union covers the entire dataset while minimizing the sum of radii of the k balls. We propose a general technical framework to overcome the challenge posed by varying radii in SoR, which yields fixed-parameter tractable (fpt) algorithms with respect to k (i.e., whose running time is $f(k)\text{ploy}(n)$ for some f). Our framework is versatile and obtains fpt approximation algorithms with constant approximation ratios for SoR as well as its variants in general metrics, such as Fair SoR and Matroid SoR, which significantly improve the previous results.

Introduction

Clustering is a fundamental tool in information technology and has been widely implemented in various fields, including artificial intelligence, data mining, bioinformatics, pattern recognition, computer vision, and more. Clustering aims to partition a set of data points into partitions, called clusters. Many clustering problems involve finding k cluster centers to minimize some objective functions. One of the most studied such objective functions is k-center which minimizes the maximum radius of clusters. Another example is the k-median, which aims to minimize the sum of distances from data points to their closest centers. Both problems have proved invaluable in various real-world applications, such as facility layout planning, healthcare resource allocation, and transportation network design, etc.

In this paper, we focus on a rather different objective function for clustering, called Sum of Radii (SoR). This problem asks to find k clusters to cover all data points, such that the sum of these cluster radii is minimized. This objective is more useful as it reduces the so-called dissection effect (Hansen and Jaumard 1997; Monma and Suri 1989), which seamlessly incorporates the strengths of both k-center and k-median. The dissection effect occurs as multiple balls (clusters) in the k-center have the same maximum radius, leading to significant overlap between clusters. Consequently, data points that should ideally belong to the same cluster may end up being assigned to different clusters due to this overlap. This phenomenon is called the dissection effect, which can be reduced by using SoR as the objective instead.

Sum of Radii is a well-known clustering problem first introduced in a seminal work of Monma and Suri (1989). This problem is NP-hard to solve, as a straightforward reduction from the Set Cover (to the non-metric SoR) can be identified. On the positive side, Charikar and Panigrahy (2004) presents the first constant approximation algorithm for SoR, achieving a 3.504-approximation through primal-dual technique. This result has held the state-of-the-art in polynomial time for nearly two decades until recently when Friggstad and Jamshidian (2022) surpasses it. They introduce a novel approach, achieving a superior 3.389-approximation using an improved LP rounding procedure. Meanwhile, for the inapproximability results, it is intriguing to note that the lower bound is unknown for SoR despite extensive research. Gibson et al. (2010) show that SoR can be solved exactly in $n^{O(\log n \cdot \log \Gamma)}$ where Γ is the diameter of the metric (i.e. maximum distance between two points from the dataset). By standard techniques, this yields a QPTAS (Quasi Polynomial Time Approximation Scheme) for SoR with running time $n^{O(\log 1/\varepsilon + \log^2 n)}$. They also show that SoR is NP-hard even in metrics with constant doubling dimensions or shortest-path metrics of edge-weighted planar graphs. Surprisingly, it is polynomial-solvable for SoR in constant-dimensional Euclidean metrics (Gibson et al. 2012).

Indeed, devising polynomial time constant approximations for clustering problems poses a significant challenge in the realm of computer science. As a result, the approach of developing fixed-parameter tractable (fpt) approximations has emerged as a successful alternative for tackling clustering problems. Though admitting an exponential factor on the parameter k, fpt algorithm for clustering is practical as the value of cluster number k is usually less than 10 in real scenarios (Pedregosa et al. 2011; Steinbach, Karypis, and Kumar 2000). Moreover, this approach has yielded interesting and valuable results that may not be achievable in polynomial time in different clustering objectives.

For k-center, Hochbaum and Shmoys (1986) proved that unless P = NP, it is NP-hard to approximate the optimal so-

*Corresponding author. Email: yc.xu@siat.ac.cn

lution within a factor less than 2. Similarly, for k-supplier, the optimal solution cannot be approximated within a factor less than 3 unless P = NP. Moreover, introducing constraints to these clustering problems can significantly increase their complexity and make achieving good approximation ratios more challenging. However, Goyal and Jaiswal (2023) have presented tight fixed-parameter tractable (fpt) approximations for many constrained variants of k-center and k-supplier clustering, contributing to more efficient and accurate clustering solutions in specific settings. A variant called non-uniform k-center is introduced by Chakrabarty, Goyal, and Krishnaswamy (2020), which allows balls with different radii to cover the dataset, resembling the concept of SoR, except that the objective is to minimize a scalable ratio for the input radii. Chakrabarty, Goyal, and Krishnaswamy (2020) shows that no constant approximation can be found for this problem in polynomial time. However, in fpt(k) time, Bandyapadhyay, Friggstad, and Mousavi (2022) showcase that the non-uniform k-center admits a simple 2-approximation algorithm, offering a promising approach for this variant.

As for k-median and k-means, it is well-known that unless P = NP, it is even NP-hard to approximate the optimal value within a factor less than $1+2/e$ for the k-median and $1+8/e$ for the k-means (Guha and Khuller 1999; Jain, Mahdian, and Saberi 2002). In recent research, significant progress has been made in developing efficient approximation algorithms for both k-median and k-means. To the best of our knowledge, for k-median problem, the best-known polynomial result can be found in Cohen-Addad et al. (2023) and Gowda et al. (2023), which presents a 2.6712-approximation achieved by two exciting improvements with the bi-point rounding procedure. Similarly, for k-means problem, Ahmadian et al. (2020) achieve a 6.357-approximation in general metrics. Both problems admit a huge gap on the lower bound. Remarkably, Cohen-Addad et al. (2019) demonstrate that tight approximation can be found in fpt time for both k-median and k-means. Furthermore, Goyal, Jaiswal, and Kumar (2020) shows that fpt algorithm can achieve valuable approximation even on some constrained variants for k-median and k-means clustering. Agrawal et al. (2023) presents a general reduction framework on solving the outliers version of k-median and k-means as well as some variants in fpt(k) time with almost tight approximation ratio.

Fpt algorithms also show their strength in dealing with SoR. Inamdar and Varadarajan (2020) presents a 28-approximation algorithm for the uniform capacitated SoR under the fpt time complexity. Another significant advancement is showcased in Bandyapadhyay, Lochet, and Saurabh (2023), where the authors propose a 15-approximation algorithm for the non-uniform capacitated SoR. This result demonstrates that even in situations where clusters have non-uniform capacity constraints, fpt algorithm can provide a solution with a relatively low approximation ratio. Moreover, the same research introduces a 4-approximation algorithm for the uniform capacitated SoR, reinforcing the effectiveness of their approach in handling uniform capacity constraints.

Related Work

Fair clustering has emerged as a highly debated and critical topic in the field of machine learning and combinatorial optimization. Fairness in clustering aims to ensure that the clustering process does not perpetuate or exacerbate existing biases and inequalities in the data and that the resulting clusters are representative and equitable across various groups or demographics. The work by Chierichetti et al. (2017) mark a significant milestone in fair clustering. They introduce the first notion of fair clustering named balance fairness, which seeks to achieve balance in the clustering output for two protected groups. They propose the "fairlet" decomposition method to achieve this balance in the context of the k-center and k-median, and this work lays the foundation for considering fairness in clustering algorithms, opening up new avenues for research in this area. Various fairness notations have been proposed for different applications, reflecting the diverse ways fairness can be defined and measured in clustering problems. One key aspect of fairness concerns data summarization, where the selected clustering centers should accurately represent the entire dataset. Kleindessner, Awasthi, and Morgenstern (2019) introduce this kind of fairness in data summarization and presente a linear time algorithm with an exponential approximation factor for the fair k-center problem. Building upon the work of Kleindessner, Awasthi, and Morgenstern (2019), Jones, Nguyen, and Nguyen (2020) further improve the results by presenting a 3-approximation algorithm that runs in linear time with respect to the input size of the dataset. Angelidakis et al. (2022) combine this model along with a private-fair concept and propose a fast 15-approximation algorithm.

Several notable results have been achieved for various matroid clustering problems as well. In the context of the matroid center on general metrics, Chen et al. (2016) leverage the pure combinatorial structure of the problem to efficiently produce a solution with an approximation ratio of 3. For the matroid median problem, Swamy (2016) introduces a simple 8-approximation algorithm based on LP-rounding techniques. However, no polynomial time algorithm has been found for the Matroid SoR problem, the state of the art on this problem is (Friggstad and Jamshidian 2022), which presents a (9+ε)-approximation in $k^{O(k)}n^{O(1)}$ time, where k denotes the number of elements in the basis of given matroid. It is worth mentioning that this result also implies a (9+ε)-approximation for fair SoR problem due to the connection between fair constraints and matroid constraints, which will be discussed in detail in the section for matroid SoR.

Problems and Our Contributions

Given a set X of n points and a metric space d on domain $X \times X$, we define the ball $B = B(c, r)$ centered at $c \in X$ with radius $r \geq 0$ to be the set $\{p \in X : d(c, p) \leq r\}$. Note that all balls throughout this paper are centered at some points from X. The cost of a set $\mathcal{B} = \{B_1, B_2, ...\}$ of balls, denoted as cost($\mathcal{B}$), is sum of radii of those balls.

Definition 1. *(Sum of Radii) SoR involves choosing a set $\mathcal{B}$ of k balls to cover all the points in X such that the sum*

Problem	State-of-the-art	Our results
SoR	3.389-approx (Friggstad and Jamshidian ESA'22)	fpt (2+ε)-approx
FairSoR	fpt (9+ε)-approx (Inamdar and Varadarajan ESA'20)	fpt (3+ε)-approx
MatSoR	fpt (9+ε)-approx (Inamdar and Varadarajan ESA'20)	fpt (3+ε)-approx

Table 1: A summary of our results with the current state-of-the-art results.

of radii on these balls is minimized. Formally, SoR can be defined as

$$\begin{aligned} \min \quad & \text{cost}(\mathcal{B}) \\ s.t. \quad & \textstyle\bigcup_{B\in\mathcal{B}} B \quad = X, \\ & |\mathcal{B}| \quad = k. \end{aligned}$$

Equivalently, SoR asks to find a subset $C = \{c_1, c_2, \dots c_k\}$ and k corresponding radius r_i to generate k balls covering the entire set X while minimizing the total radius $\sum_{i=1}^{k} r_i$. As former mentioned, SoR is NP-hard to solve in general metrics.

In addition, fairness can be introduced to the SoR in the case that there are demographic groups that compose X. Suppose that X is divided into m demographic groups, such that X_i is the set of points in X from demographic group i, where $\bigcup_{i=1}^{m} X_i = X$. To introduce fairness to the SoR, each of the m demographic groups is givens a value k_i, where k_i is the number of center which can be chosen from demographic group i.

Definition 2. *(Fair Sum of Radii) FairSoR involves choosing a set $\mathcal{B}$ of k balls which satisfies the fairness constraint to cover X such that the sum of radii on these balls is minimized. Formally, FairSoR can be defined as*

$$\begin{aligned} \min \quad & \text{cost}(\mathcal{B}) \\ s.t. \quad & \textstyle\bigcup_{B\in\mathcal{B}} B \quad = X, \\ & \textstyle\sum_{i=1}^{m} k_i \quad = k, \\ & |C \cap X_i| \quad = k_i, 1 \le i \le m. \end{aligned}$$

Analogously, matroid constraint can be naturally introduced to SoR. Consider a finite set E as the ground set and $\mathcal{I}$ be a family of independent sets of E (i.e. non-empty collection of subsets of E), the pair $\mathcal{M} = (E, \mathcal{I})$ is a matroid if 1) $A \in \mathcal{I}$ implies that all subsets of A is also included in $\mathcal{I}$, including $\emptyset$; 2) if $A, B \in \mathcal{I}$, and $|A| < |B|$, then there exists an element $e \in B \setminus A$ such that $A \cup \{e\} \in \mathcal{I}$. It is worth mentioning that all inclusion-wise maximal independent sets of a matroid $\mathcal{M}$ have equal size, and they are called the basis of $\mathcal{M}$. Consider a matroid whose group set is X as the input, $\mathcal{M} = (X, \mathcal{I})$, MatSoR asks the following quesetion:

Definition 3. *(Matroid Sum of Radii) MatSoR involves choosing a set $\mathcal{B}$ of k balls centered at an independent set $C \in \mathcal{I}$ to cover X while minimizing the sum of radii on these balls. Formally, MatSoR can be defined as*

$$\begin{aligned} \min \quad & \text{cost}(\mathcal{B}) \\ s.t. \quad & \textstyle\bigcup_{B\in\mathcal{B}} B \quad = X, \\ & C \quad \in \mathcal{I}. \end{aligned}$$

Our Results In this work, we are interested in designing fpt algorithms for SoR, FairSoR, and MatSoR. Our main result is a unified framework that returns (2+ε)-approximation for SoR and (3+ε)-approximation for both FairSoR and MatSoR respectively that all run in times of $2^{k\log(k/\varepsilon)} n^{O(1)}$. The basis of our framework lies in the ability that we can enumerate all possible assignments for any points to approximate optimal solution in fpt(k) time, which allows us to overcome the challenge posed by different cluster radii occurring in SoR. Moreover, our framework can serve as a versatile tool that one can extend to obtain approximation-preserving FPT reductions for related SoRs, such as FairSoR and MatSoR, and find competitive approximate solutions in an efficient manner, building upon the inner connection of fair constraint and matroid constraint. We conclude by giving our results and the state-of-the-art results in table 1 above.

Sum of Radii

In this section, we propose a general approximation algorithm for SoR. While straightforward, this approach is effective in handling the varying radii that occur in SoR. By allowing an extra factor of parameter k on the running time, we can guess all the optimal radii within a factor of ε and guess the assignment for each point correctly. By enumerating enough rounds, we can achieve both with a constant probability in fpt(k) time. After having a radius profile for the optimal solution, our problem can degenerate into k-center, which allows us to solve in a greedy manner.

From now on, we denote $\mathcal{B}^* = \{B_1^*, B_2^*, \dots, B_k^*\}$ as the set of balls of a hypothetical optimal solution, and let r_i^* and c_i^* denote the corresponding radius and center of B_i^*, respectively. By Lemma 1, we can assume that we know an ε-approximate radius r_i for each r_i^* in polynomial time.

Lemma 1. *For any fixed $\varepsilon \ge 0$, there exists a randomized algorithm that finds a radii profile $\{r_1, \dots, r_k\}$ such that*

$$r_i^* \le r_i, 1 \le i \le k,$$

$$\sum_{1\le i\le k} r_i \le (1+\varepsilon) \sum_{1\le i\le k} r_i^*$$

with probability at least $\frac{\varepsilon^{k-1}}{(k-1)^{k-1} n^2}$, running in linear time.

Proof. First, we can guess the value of the maximum radius r_{max}^* in the optimal solution. Since r_{max}^* corresponds to the distance between two points in X, there are at most n^2 possible choices for r_{max}^*. Therefore, the algorithm can choose this value uniformly at random with a probability of $1/n^2$. Since the largest radius is guessed precisely, for the other $k-1$ radius, choose the value in the interval

$[r_i^*, r_i^* + \frac{\varepsilon r_{max}}{k-1}]$ would yield the desired property of radii profile. Moreover, since r_{max}^* is the largest radius, the probability of selecting the other radius in the desired interval is at least $(\frac{\frac{\varepsilon r_{max}^*}{k-1}}{r_{max}^*}) = \frac{\varepsilon}{k-1}$, which is $\frac{\varepsilon^{k-1}}{(k-1)^{k-1}n^2}$ in total. This concludes the proof. □

For ease of discussion, we can now assume that an approximate value r_i for each r_i^* is given. The underlying idea of our algorithm hinges on guessing the assignments for any point in fpt(k) time since there are only k clusters in the optimal solution, which enables us to effectively handle the varying radii in the SoR. Then we can treat this problem as a k-center instance, using an iterative covering idea to yield a good approximation, as shown in Algorithm 1.

Algorithm 1: Iterative covering

Require: A metric space (X, d), an integer k.
Ensure: A center set C, a ball set $\mathcal{B}$.

```
for every radius profile {r_1, r_2, ... r_k} do
    while X ≠ ∅ do
        Select an point c ∈ X arbitrarily, guess the optimal ball B_i^* which contains c, select the approximate radiu r_i for r_i^* from radius profile R.
        C ← {c}, B ← {B(c, 2r_i)}, X ← X \ B(c, 2r_i).
    end while
    if |C| > k then
        return
    end if
    if cost(B) > 2 Σ_{i=1}^{k} r_i then
        return
    end if
end for
return C, B.
```

Algorithm 1 builds the solution by choosing the remaining points c from X arbitrarily as centers. In each iteration, the algorithm will remove the point in $B(c, 2r_i)$, as they are covered by center c within a distance of $2r_i$.

Observation 1. *Algorithm 1 returns a valid solution when the while loop of Algorithm 1 terminates within k iterations, and the cost is less than $2\sum_{i=1}^{k} r_i$.*

Proof. As the iteration goes, if we guess the assignment of each point c in the optimal solution correctly, i.e., $c \in B_i^*$, we have $d(c, c_i^*) \leq r_i^*$. Meanwhile, our radius is a good approximation for r_i^* based on Lemma 1, $r_i \geq r_i^*$, then for any point x in $B(c_i^*, r_i^*)$, we have $d(c, x) \leq d(c, c_i^*) + d(c_i^*, x) \leq 2r_i^* \leq 2r_i$. Therefore, ball $B(c, 2r_i)$ would cover the entire optimal ball $B(c_i^*, r_i^*)$, $B(c_i^*, r_i^*) \subseteq B(c, 2r_i)$. Since the optimal solution $\mathcal{B}^*$ covers all points in X using exactly k balls, our algorithm will execute with no more than k iterations. Moreover, cost($\mathcal{B}$) is no more than $2\sum_{i=1}^{k} r_i$ in that case, which would allow the solution to pass the cost check in the second if statement. □

Lemma 2. *Algorithm 1 returns a (2+ε)-approximation for SoR.*

Proof. Recall that cost($\mathcal{B}$) is the radius of all balls in $\mathcal{B}$.

$$\text{cost}(\mathcal{B}) \leq 2\sum_{i=1}^{k} r_i \leq 2(1+\varepsilon')\sum_{i=1}^{k} r_i^* = (2+\varepsilon)\text{cost}(\mathcal{B}^*).$$

The first inequality holds because of Observation 1, and the second inequality holds followed by Lemma 1. As parameter ε' is a scalable small value in our procedure, we can fix $\varepsilon = 2\varepsilon' > 0$ to obtain the desired solution. Therefore, Algorithm 1 returns a (2+ε)-approximation for SoR, which completes the proof. □

Theorem 1. *For any fixed parameter $\varepsilon > 0$, there is a (2 + ε)-approximation algorithm for SoR running in $2^{O(k\log k/\varepsilon)}n^3$ time with constant probability.*

Proof. Consider the probability of outputting a valid solution in Algorithm 1. First, we guess the optimal radius profile with probability at least $\frac{\varepsilon^{k-1}}{(k-1)^{k-1}n^2}$, as shown in Lemma 1. In each iteration of the while loop in Algorithm 1, we guess the belonging for point c with probability $\frac{1}{k}$. Since the while loop runs no more k iterations to output a valid solution, the whole algorithm succeeds with probability at least $\frac{\varepsilon^{k-1}}{(k-1)^{k-1}n^2} \cdot \frac{1}{k^k}$. This means that by repeating the whole algorithm $2^{O(k\log k/\varepsilon)} \cdot n^2$ times, we can obtain a $(2+\varepsilon)$-approximation algorithm with constant probability. Furthermore, the worse case for the while loop takes n iterations, as there are at most n points to be covered. Since the guessing phase for a radii profile takes linear time, the overall time complexity is bounded by $2^{O(k\log k/\varepsilon)}n^3$, which completes the proof. □

While it is rare for a solution to be produced in a single execution, we have demonstrated that the probability of achieving this outcome can be elevated to a constant value by conducting multiple repetitions, leveraging the concept of union bounds. This idea has proved to be useful in practice as we can scale the probability even to a high probability by running enough rounds. Furthermore, it is worth noting that the process of derandomizing this algorithm is also feasible by replacing random decisions with exhaustive searches in each iteration. The time complexity still remains the same.

Fair Sum of Radii

As we have the (2+ε)-approximation for SoR in the last section, we can use it as a subroutine to solve the fair variant. Recall that in a FairSoR instance, each point $c \in X$ belongs to a demographic group X_i, and the number of centers that can be chosen from each demographic group value k_i is given.

To deal with fair constraints, our main idea is first to estimate the radius profile for the optimal solution of FairSoR. Employing Algorithm 1, we generate an initial solution that may not conform to the fairness constraints. Then we modify the unfair solution by exchanging centers with some nearby candidates to satisfy the fairness constraint. The modification procedure, outlined in Algorithm 2, is designed to simultaneously exchange centers in order to uphold fairness constraints, within a low extra cost on the objective.

We carefully construct a bipartite graph to showcase all possible exchanges to meet the fairness constraint. Vertices on the left side have represented the centers chosen in the initial solution, while the vertices on the right side belong to each demographic group. Without loss of generality, suppose vertex a belongs to $\mathcal{B}_i^*$, while vertex b represents one of k_j centers that can be selected from demographic groups X_j, then an edge (a, b) exists if and only if there is a candidate center to be exchanged in the demographic group X_j for unfair center a within radius r_i. Therefore, any matching would present a feasible fair solution with a low extra cost from the unfair solution output by Algorithm 1, and we can find the maximum matching effectively.

Algorithm 2: Fair exchange via matching

Require: A metric space (X, d), an integer k
Require: Demographic group $X_1, X_2, \cdots, X_m$, integer $k_1, k_2, \cdots, k_m$.
Ensure: A fair center set C, a ball set $\mathcal{B}$

for every radius profile $\{r_1, r_2, \ldots r_k\}$ **do**
  Find an unfair solution C using Algorithm 1
  Create a vertex set V_1 of size $|C|$, representing the center points in unfair Set C.
  Create a vertex set V_2 of size k for m demographic groups, where group i have k_i vertices.
  Update $V = V_1 \cup V_2$
  for $j = 1$ to m **do**
    for $i = 1$ to $|C|$ **do**
      if $d(c_i, X_j) \leq r_i$ **then**
        Update E with an edge from vertex c_i to every vertex in group X_j.
        label it with $arg\min_{x \in X_j} d(c_i, x)$.
      end if
    end for
  end for
  Run Hopcroft-Karp Maximum Matching algorithm on the bipartite graph G.
  if maximum matching $= |C|$ **then**
    if $(c, c') \in$ maximum matching **then**
      $C \leftarrow C \setminus \{c\} \cup \{c'\}$,
      $\mathcal{B} \leftarrow \mathcal{B} \setminus \{B(c, 2r_i)\} \cup \{B(c', 3r_i)\}$
    end if
    return $C, \mathcal{B}$
  else
    return $\emptyset$
  end if
end for

Lemma 3. *There is an iteration that the maximum matching of the graph G is equal to $|C|$.*

Proof. On one hand, it is apparently that maximum matching is no more than $|C|$ as there are only $|C|$ vertices on the left side. On the other hand, suppose we assign the radius correctly to every point based on its distribution in the optimal solution. Under this circumstance, consider any two centers chosen in different rounds, and we can assume that with loss of generality, c_i, c_j, $c_i \in B_i^*, c_j \in B_j^*$,$i < j$. First, $d(c_i, c_i^*) \leq r_i^* \leq r_i$, assume that c_i^* belongs to demographic group X_l, so there will be at least one edge connecting c_i and every k_l vertex representing the group of X_l in the bipartite graph. Similarly, we update edges for c_j and c_j^* in the bipartite graph. Meanwhile, as we remove $B(c_i, 2r_i)$ in each iteration of the while loop during Algorithm 1, which includes the entire B_i^*, we will not choose two centers from the same optimal ball. Therefore, the optimal center that can be exchanged for different centers is unique. In conclusion, there will be $|C|$ available optimal centers to be exchanged, so the maximum matching output by Algorithm 2 is equal to $|C|$, which implies the lemma. □

After finding the maximum matching of size $|C|$, we update the center set with a fair center set shown on the right side of the bipartite graph G, and construct a new solution by replacing each ball $B(c, 2r_i)$ with a new ball $B(c', 3r_i)$ if edge (c, c') is in the maximum matching.

Lemma 4. *Algorithm 2 outputs a feasible solution for FairSoR when maximum matching is equal to $|C|$.*

Proof. When maximum matching is equal to $|C|$, we can exchange all the centers on the left side with a point represented by the vertex of the right side. Since matching is a set of edges in which no two edges share an endpoint, it means that no fairness constraint is violated by the solution output in this exchange manner. Recalling that for any point x in optimal B_i^*, we use a ball $B(c, 2r_i)$ centered at c to cover it in the unfair situation, which means $d(c, x) \leq 2r_i$, and now we can find a new center c' such that $d(c', c) \leq r_i$ via matching. $d(c', x) \leq d(c', c) + d(c, x) \leq 3r_i$, then we cover all points from B_i^* by placing a ball with a radius of $3r_i$ centered at c'. Therefore, the solution is a feasible solution for FairSoR. □

Lemma 5. *Algorithm 2 returns a (3+ε)-approximation for FairSoR.*

Proof. First, as we carefully construct the graph and find the maximum matching, we can guarantee that the solution output by Algorithm 2 satisfies the fairness constraint. Meanwhile, as we find all exchange centers within a ball of r_i, the total extra cost incurred by the exchange is less than $\sum_{i=1}^{k} r_i$. Moreover, we have proved that cost of the initial unfair solution is less than $2\sum_{i=1}^{k} r_i$ in SoR, and it still holds in the case of FairSoR. In conclusion, it holds that

$$\text{cost}(\mathcal{B}) \leq 3\sum_{i=1}^{k} r_i \leq 3(1+\varepsilon')\sum_{i=1}^{k} r_i^* = (3+\varepsilon)\text{cost}(\mathcal{B}^*).$$

Analogously, we can fix $\varepsilon = 3\varepsilon' > 0$ to obtain the desired approximation ratio. Additionally, if the solution uses less than k centers, we can pick up the remaining centers arbitrarily without violating the fairness constraint, which will not increase the cost of the solution. That implies the lemma. □

Theorem 2. *For any fixed parameter $\varepsilon > 0$, there is a (3 + ε)-approximation algorithm for FairSoR running in $2^{O(k \log k/\varepsilon)} n^3$ time with constant probability.*

Proof. Recalling that in Theorem 1, we can achieve $(2+\varepsilon)$-approximation for SoR in $O(2^{O(k\log k/\varepsilon)}n^3)$ time with constant probability by repeating Algorithm 1 enough rounds. Furthermore, we modify the initial solution to meet the fairness constraint with a constant loss of the objective, returns $(3+\varepsilon)$-approximation for FairSoR, as shown in Lemma 5. As the maximum matching can be effectively solved in $O(|E|\sqrt{|V|}) = O(nk\sqrt{k})$ (Hopcroft and Karp 1973), the overall time complexity of Algorithm 2 is still bounded by $O(2^{O(k\log k/\varepsilon)}n^3)$, which completes the proof. □

Matroid Sum of Radii

In this problem, we are given a finite metric space (X, d), and a matroid $\mathcal{M} = (X, \mathcal{I})$ of ground set X, whose size of basis is k. We are required to place a set of balls whose centers at $C \in \mathcal{I}$ (i.e. an independent set in the given matroid) to cover the points in X while minimizing the total sum of radii on these balls. The following observations about MatSoR and FairSoR inspire us the inner relation between FairSoR and MatSoR.

Observation 2. *The optimal center set C^* for MatSoR is one of the bases (i.e., independent set with maximum cardinality) of the matroid.*

Proof. Suppose that C^* is not the basis of the matroid, then consider the definition of a matroid, for any basis A of matroid $\mathcal{M}$, there exists an element $a \in A \setminus C^*$, such that $C^* \cup \{a\} \in \mathcal{I}$. To cover the same set X, the cost of $\mathcal{B}$ centered at $C^* \cup \{a\}$ is no more than the cost associated with C^* because $C^* \cup \{a\}$ uses more centers than C^*, which contradicts the optimality of C^*. In other words, we can only consider $|C^*| = k$ in MatSoR. □

Observation 3. *All feasible solutions C of FairSoR and their subsets form a matroid $\mathcal{M} = (X, \mathcal{I})$.*

Proof. We prove this by verifying the basic properties of matroid. It is easy to see that $\emptyset$ does not violate the fairness constraint, $\emptyset \in \mathcal{I}$. Furthermore, if $C \in \mathcal{I}$, then all subsets of C will not violate the fairness constant as they do not use more centers in each demographic group. Lastly, if $C_1, C_2 \in \mathcal{I}$, and $|C_1| < |C_2|$, then there exists an element $c \in C_2 \setminus C_1$ such that $C_1 \cup \{c\} \in \mathcal{I}$. As C_2 satisfies the fairness constraint using more centers than C_1, there exists some demographic group that remains available to choose a center in C_1, which can be found in C_2. In fact, the matroid constructed in this manner is called partition matroid as demographic groups X_i form a partition for dataset X. □

Based on the above observations, the algorithm for the FairSoR has provided valuable insights, revealing that the fair constraint can be viewed as a special case of a matroid constraint, specifically a partition matroid. Building upon this observation, we can adopt a similar approach for FairSoR to approximate MatSoR, which is outlined in Algorithm 3.

A useful subroutine of our algorithms is the maximum matroid intersection problem defined as follows.

Definition 4. *(Maximum Matroid Intersection) Given two matroids $\mathcal{M} = (X, \mathcal{I}_1)$, $\mathcal{M}' = (X, \mathcal{I}_2)$ defined on the same ground set X, and the goal is to find a common independent set C with maximum elements in the two matroids, i.e.,* $\max |C|, C \in \mathcal{I}_1 \cap \mathcal{I}_2$.

It is well-known that this problem can be solved in polynomial time (Schrijver 2003). By utilizing the solution output by Algorithm 1, we can generate a matroid based on the clustering result implied by Algorithm 1. Then by solving the maximum matroid intersection, we obtain a good approximation for MatSoR.

Algorithm 3: Exchange via matroid intersetcion

Require: A metric space (X, d), matroid $\mathcal{M} = (X, \mathcal{I})$ with base of size k
Ensure: A center set $C \in \mathcal{I}$, a ball set $\mathcal{B}$
1: **for** every radius profile $\{r_1, r_2, \dots r_k\}$ **do**
2: Find a center set C using Algorithm 1;
3: Construct a ball set $\mathcal{B}' = \{B(c_1, r_1), B(c_2, r_2), \dots\}$ based on C;
4: Define a new matroid $\mathcal{M}' = (X, \mathcal{I}')$, such that $\forall C' \in \mathcal{I}'$, $|C' \cap B(c_i, r_i)| \le 1, 1 \le i \le k$;
5: Solve the maximum matroid intersection problem for matroids $\mathcal{M}$ and $\mathcal{M}'$ to find a new center set $C' \in \mathcal{I}'$;
6: $\mathcal{B} \leftarrow \emptyset$;
7: **for** every $c' \in C' \cap B(c_i, r_i)$ **do**
8: $\mathcal{B} \leftarrow \mathcal{B} \cup \{B(c', 3r_i)\}$;
9: **end for**
10: **return** $C', \mathcal{B}$.
11: **end for**

Firstly, we guess a radius profile of the optimal solution for MatSoR and obtain a $(2+\varepsilon)$-approximation for SoR by Algorithm 1. After that, we define a new ball set $\mathcal{B}' = \{B(c_1, r_1), B(c_2, r_2) \dots\}$. Based on this ball set, we can define a new matroid such that every independent set contains no more than one point from each ball $B(c_i, r_i)$.

Then we solve the maximum matroid intersection problem for matroids $\mathcal{M}$ and $\mathcal{M}'$ to find a new center set $C' \in \mathcal{I}'$, and place a ball of radius $3r_i$ on c' if c' is covered in ball $B(c_i, r_i)$. We show that the new set of balls $\mathcal{B}$ found by this algorithm covers X with good approximation.

Lemma 6. *$\mathcal{B}$ computed by Algorithm 3 covers X, whose cost is upper bounded by $(3+\varepsilon)$cost($\mathcal{B}^*$).*

Proof. For every ball $B(c_i, r_i)$, we can pick an optimal center c_i^* as $d(c_i, c_i^*) \le r_i$. We let the resulting set be $\hat{C}^*$. Apparently, $\hat{C}^* \subseteq C^*$, which means $\hat{C}^*$ is an independent set of matroid $\mathcal{M}$. Furthermore, it contains exactly one point from each ball $B(c_i, r_i)$, so $\hat{C}^*$ is also an independent set of matroid $\mathcal{M}'$, which means $\hat{C}^*$ is an intersection of two matroids. Let C' denote the maximum intersection set computed in line 5 of Algorithm 3, it holds that $|C'| \ge |\hat{C}^*|$. As C' is independent in both two matroids, it implies that C' contains exactly one point from each ball $B(c_i, r_i)$.

Consider an arbitrary point $x \in X$, and suppose it is covered by ball $B(c_i, 2r_i)$ in Algorithm 1, $d(x, c_i) \le 2r_i$.

Since C' contains excatly one point from each ball $B(c_i, r_i)$, there exists a new center $c' \in C' \cap B(c_i, r_i)$. Meanwhile, $d(c', x) \leq d(c', c_i) + d(c_i, x) \leq r_i + 2r_i \leq 3r_i$. In other words, any point x is covered by the ball of radius $3r_i$ centered at c'. In other words, $\mathcal{B}$ cover the entire X using several balls with radius $3r_i$. Therefore, it holds that

$$\text{cost}(\mathcal{B}) \leq 3\sum_{i=1}^{k} r_i \leq 3(1+\varepsilon')\sum_{i=1}^{k} r_i^* = (3+\varepsilon)\text{cost}(\mathcal{B}^*).$$

The first inequality holds since any point x is covered by the ball of radius $3r_i$ centered at c', and the second inequality holds because of Lemma 1. As we can fix parameter ε such that $\varepsilon = 3\varepsilon' > 0$, we can conduct that cost($\mathcal{B}$) is bounded by (3+ε) times the cost of $\mathcal{B}^*$, which completes the proof. □

Theorem 3. *For any fixed parameter $\varepsilon > 0$, there exists a (3 + ε)-approximation algorithm for MatSoR running in $2^{O(k \log k/\varepsilon)} n^{O(1)}$ time with constant probability.*

Proof. Similar to the approach of dealing with FairSoR, we repeat Algorithm 1 enough rounds to achieve a (2 + ε)-approximation for SoR as an initial solution in $O(2^{O(k \log k/\varepsilon)} n^3)$ time with constant probability. Furthermore, we modify the initial solution to meet the matroid constraint with a constant loss of the objective, returns (3 + ε)-approximation for MatSoR, as shown in Lemma 6. Considering that the maximum matroid intersection is solvable in polynomial time, the time complexity is bounded by $2^{O(k \log k/\varepsilon)} n^{O(1)}$ in total, which completes the proof. □

Conclusion

To summarize, this paper presents a comprehensive study of SoR and its variants. By proposing a general approach, we have devised fixed-parameter tractable algorithms with competitive approximation ratios. Furthermore, we extend our approach to address multiple variants of SoR, incorporating additional constraints that arise in real-world clustering scenarios. By successfully handling fairness and matroid constraints, we demonstrate the versatility of our approach in tackling complex clustering challenges with diverse constraints.

For future work, it remains open whether SoR admits a QPTAS in fpt(k) time or not under some complexity hypothesis. On the other hand, obtaining a constant approximation for the constraint version of SoR in polynomial time is also an interesting direction.

Acknowledgements

Xianrun Chen and Yicheng Xu are supported by Natural Science Foundation of China 12371321, Fundamental Research Project of Shenzhen City JCYJ20210324102012033, Shenzhen key Laboratory of Intelligent Bioinformatics ZDSYS20220422103800001, and Guangxi Key Laboratory of Cryptography and Information Security GCIS202116. Dachuan Xu is supported by Natural Science Foundation of China 12371320. Yong Zhang is supported by Natural Science Foundation of China 12071460.

References

Agrawal, A.; Inamdar, T.; Saurabh, S.; and Xue, J. 2023. Clustering What Matters: Optimal Approximation for Clustering with Outliers. In *AAAI*, 6666–6674.

Ahmadian, S.; Norouzi-Fard, A.; Svensson, O.; and Ward, J. 2020. Better Guarantees for k-means and Euclidean k-Median by Primal-Dual Algorithms. *SIAM J. Comput.*, 49(4).

Angelidakis, H.; Kurpisz, A.; Sering, L.; and Zenklusen, R. 2022. Fair and Fast k-center Clustering for Data Summarization. In *ICML*, volume 162 of *Proceedings of Machine Learning Research*, 669–702.

Bandyapadhyay, S.; Friggstad, Z.; and Mousavi, R. 2022. Parameterized Approximation Algorithms for k-center Clustering and Variants. In *AAAI*, 3895–3903.

Bandyapadhyay, S.; Lochet, W.; and Saurabh, S. 2023. FPT Constant-Approximations for Capacitated Clustering to Minimize the Sum of Cluster Radii. In *SoCG*, volume 258 of *LIPIcs*, 12:1–12:14.

Chakrabarty, D.; Goyal, P.; and Krishnaswamy, R. 2020. The Non-Uniform k-Center Problem. *ACM Trans. Algorithms*, 16(4): 46:1–46:19.

Charikar, M.; and Panigrahy, R. 2004. Clustering to minimize the sum of cluster diameters. *J. Comput. Syst. Sci.*, 68(2): 417–441.

Chen, D. Z.; Li, J.; Liang, H.; and Wang, H. 2016. Matroid and Knapsack Center Problems. *Algorithmica*, 75(1): 27–52.

Chierichetti, F.; Kumar, R.; Lattanzi, S.; and Vassilvitskii, S. 2017. Fair clustering through fairlets. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 5036–5044.

Cohen-Addad, V.; Grandoni, F.; Lee, E.; and Schwiegelshohn, C. 2023. Breaching the 2 LMP Approximation Barrier for Facility Location with Applications to k-Median. In *SODA*, 940–986.

Cohen-Addad, V.; Gupta, A.; Kumar, A.; Lee, E.; and Li, J. 2019. Tight FPT Approximations for k-median and k-means. In *ICALP*, volume 132 of *LIPIcs*, 42:1–42:14.

Friggstad, Z.; and Jamshidian, M. 2022. Improved Polynomial-Time Approximations for Clustering with Minimum Sum of Radii or Diameters. In *ESA*, volume 244 of *LIPIcs*, 56:1–56:14.

Gibson, M.; Kanade, G.; Krohn, E.; Pirwani, I. A.; and Varadarajan, K. R. 2010. On Metric Clustering to Minimize the Sum of Radii. *Algorithmica*, 57(3): 484–498.

Gibson, M.; Kanade, G.; Krohn, E.; Pirwani, I. A.; and Varadarajan, K. R. 2012. On Clustering to Minimize the Sum of Radii. *SIAM J. Comput.*, 41(1): 47–60.

Gowda, K. N.; Pensyl, T.; Srinivasan, A.; and Trinh, K. 2023. Improved Bi-point Rounding Algorithms and a Golden Barrier for k-Median. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 987–1011. SIAM.

Goyal, D.; and Jaiswal, R. 2023. Tight FPT approximation for constrained *k*-center and *k*-supplier. *Theor. Comput. Sci.*, 940(Part): 190–208.

Goyal, D.; Jaiswal, R.; and Kumar, A. 2020. FPT Approximation for Constrained Metric k-median/means. In *IPEC*, volume 180 of *LIPIcs*, 14:1–14:19.

Guha, S.; and Khuller, S. 1999. Greedy Strikes Back: Improved Facility Location Algorithms. *Journal of Algorithms*, 31(1): 228–248.

Hansen, P.; and Jaumard, B. 1997. Cluster analysis and mathematical programming. *Mathematical Programming*, 79: 191–215.

Hochbaum, D. S.; and Shmoys, D. B. 1986. A Unified Approach to Approximation Algorithms for Bottleneck Problems. *Journal of the ACM*, 33(3): 533–550.

Hopcroft, J. E.; and Karp, R. M. 1973. An $n^{5/2}$ Algorithm for Maximum Matchings in Bipartite Graphs. *SIAM Journal on Computing*, 2(4): 225–231.

Inamdar, T.; and Varadarajan, K. R. 2020. Capacitated Sum-Of-Radii Clustering: An FPT Approximation. In *ESA*, volume 173 of *LIPIcs*, 62:1–62:17.

Jain, K.; Mahdian, M.; and Saberi, A. 2002. A new greedy approach for facility location problems. In *STOC*, 731–740.

Jones, M.; Nguyen, H.; and Nguyen, T. 2020. Fair k-centers via maximum matching. In *37th International Conference on Machine Learning*, 4940–4949. PMLR.

Kleindessner, M.; Awasthi, P.; and Morgenstern, J. 2019. Fair k-center clustering for data summarization. In *36th International Conference on Machine Learning*, 3448–3457. PMLR.

Monma, C.; and Suri, S. 1989. Partitioning points and graphs to minimize the maximum or the sum of diameters. In *Graph Theory, Combinatorics and Applications (Proc. 6th Internat. Conf. Theory Appl. Graphs)*, volume 2, 899–912.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; VanderPlas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.*, 12: 2825–2830.

Schrijver, A. 2003. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer.

Steinbach, M. S.; Karypis, G.; and Kumar, V. 2000. A Comparison of Document Clustering Techniques.

Swamy, C. 2016. Improved Approximation Algorithms for Matroid and Knapsack Median Problems and Applications. *ACM Trans. Algorithms*, 12(4): 49:1–49:22.