r-Gatherings on a Star

Shareef Ahmed[1], Shin-ichi Nakano[2], and Md. Saidur Rahman[1]

[1] Graph Drawing and Information Visualization Laboratory,
Department of Computer Science and Engineering,
Bangladesh University of Engineering and Technology, Dhaka, Bangladesh
{shareefahmed,saidurrahman}@cse.buet.ac.bd
[2] Gunma University, Kiryu 376-8515, Japan
nakano@cs.gunma-u.ac.jp

Abstract. Let C be a set of n customers and F be a set of m facilities. An r-gather clustering of C is a partition of the points in clusters such that each cluster contains at least r points. The r-gather clustering problem asks to find an r-gather clustering which minimizes the maximum distance between any two points in a cluster. An r-gathering of C is an assignment of each customer $c \in C$ to a facility $f \in F$ such that each open facility has zero or at least r customers. The r-gathering problem asks to find an r-gathering that minimizes the maximum distance between a customer and its facility. In this work we consider the r-gather clustering and r-gathering problems when the customers and the facilities are lying on a "star". We show that the r-gather clustering problem and the r-gathering problem with points on a star with d rays can be solved in $O(rn+(r+1)^d dr)$ and $O(n+r^2m+d^2r^2(d+\log m)+(r+1)^d 2^d (r+d)d)$ time respectively.

Keywords: r-Gathering, Clustering, Facility location problem

1 Introduction

Let C be a set of n points. An r-gather clustering of C is a partition of the points of C in clusters such that each cluster contains at least r points. The cost of a cluster is the maximum distance between a pair of points in the cluster. The cost of an r-gather clustering is the maximum cost among the costs of the clusters. The r-gather clustering problem asks to find an r-gather clustering of C with minimum cost [2].

Let C be a set of n customers and F be a set of m facilities, $d(c, f)$ be the distance between $c \in C$ and $f \in F$. An r-gathering of C to F is an assignment A of C to F such that each facility has at least r or zero customers assigned to it. The cost of an r-gathering is $\max_{c \in C}\{d(c, A(c))\}$ which is the maximum distance between a customer and its facility. The r-gathering problem asks to find an assignment of C to F having the minimum cost [4]. This problem is also known as the min-max r-gathering problem. The other version of the problem is known as the min-sum r-gathering problem which asks to find an assignment which minimizes $\sum_{c \in C} d(c, A(c))$ [9, 7]. In this paper we consider the min-max

r-gathering problem and we use the term r-gathering problem to refer the min-max version unless specified otherwise.

Both the r-gather clustering and r-gathering problems are NP-complete in general [2, 4]. For r-gather clustering problem a 2-approximation algorithm is known [2]. For the r-gathering problem a 3-approximation algorithm is known and it is proved that the problem cannot be approximated within a factor less than 3 for $r > 3$ unless $P = NP$ [4]. Recently, both problems are considered in a setting where all the points are lying on a line. An $O(n \log n)$ time algorithm [3] based on the matrix search method [5, 1], and an $O(rn)$ time algorithm [10] by reduction to the min-max path problem in a weighted directed graph [6] are known for the r-gather clustering problem when all the points are on a line. For the r-gathering problem an $O((n + m) \log(n + m))$ time algorithm [3] based on the matrix search method [5, 1], an $O(n + m \log^2 r + m \log m)$ time algorithm [8], and an $O(n + r^2 m)$ time algorithm [10] by reduction to the min-max path problem in a weighted directed graph [6] are known when all the customers and facilities are on a line. Recently the r-gather clustering problem is studied on mobile setting and a 4-approximation distributed algorithm is known [11].

In this paper, we consider both the r-gathering clustering and r-gathering problem when the points are on a star. Consider a scenario where a number of streets meet in a junction, and residents live by the streets. We wish to set up emergency shelters on the streets so that each shelter can serve at least r residents. The distance between two points are measured along the lines. We also wish to locate shelters so that evacuation time span can be minimized. This scenario can be modeled by the r-gather clustering problem where all input points C are located on a star. In an r-gather clustering of C having the minimum cost, each emergency shelters is located at the center of each cluster. On the other hand, if the set F of possible locations of shelters on the star is also given with the set C of residents and we wish to find an assignment of C to F with minimizing the evacuation time so that each shelter serve at least r residents, then the scenario can be modeled by the r-gathering problem where the points of C and F are located on a star. In this case, an r-gathering corresponds to an assignment of residents to shelters such that each "open" shelter serves at least r residents and the r-gathering problem finds the r-gathering minimizing the evacuation time.

When the points are on a line, each cluster of an optimal r-gather clustering consists of consecutive points on the line [10]. However, when the points are on a star, some clusters may not consists of consecutive points in the optimal r-gather clustering. For example, see Figure 1. We can observe that at least one cluster consists of non-consecutive points in any optimal solution. Figure 1 demonstrates an optimal solution for this scenario.

In this paper we give an $O(rn + (r + 1)^d dr)$ time algorithm for r-gather clustering problem on a star, and an $O(n+r^2 m+d^2 r^2 (d+\log m)+(r+1)^d 2^d (r+d)d)$ time algorithm for the r-gathering problem on a star, where d is the number of rays that form the star.

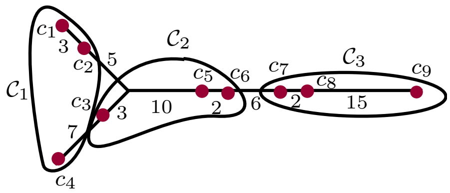

Fig. 1. An optimal 3-gather clustering on a star.

The rest of the paper is organized as follows. In Section 2 we define the problems and define terms used in the paper. In Section 3 we give an algorithm for the r-gather clustering problem on a star. In Section 4 we give an algorithm for the r-gathering problem on a star. Finally Section 5 is a conclusion.

2 Preliminaries

In this section we define two problems and some terms used in this paper.

Let $\mathcal{L} = \{l_1, l_2, \cdots, l_d\}$ be a set of d rays where all the rays of $\mathcal{L}$ share a common source point o. We call the set of rays $\mathcal{L}$ a *star* and the common source point o the *center* of the star. The *degree* of a star is the number d of rays which form the star. The Euclidean distance between two points p, q is denoted by $d_E(p, q)$. We denote by $d(p, q)$ the distance between two points p, q which is measured along the rays. If p and q are both located on the same ray, then $d(p, q) = d_E(p, q)$. On the other hand, if p and q are located on different rays, then $d(p, q) = d_E(p, o) + d_E(o, q)$. A cluster consists of points from two or more rays is a *multi-ray cluster*, otherwise a *single-ray cluster*. Two points p and q are the *end-points* of a cluster $\mathcal{C}$ if $d(p, q) = cost(\mathcal{C})$. A point p in a cluster $\mathcal{C}$ is a *far point* of $\mathcal{C}$, denoted by $e(C)$, if $d(o, p) \geq d(o, q)$ for each $q \in \mathcal{C}$.

We now define the first problem. Let $C = \{c_1, c_2, \cdots, c_n\}$ be n points located on a star. An *r-gather clustering* of C is a partition of the points of C into clusters such that each cluster contains at least r points. The cost of a cluster $\mathcal{C}$, denoted by $cost(\mathcal{C})$, is $\max_{p,q \in \mathcal{C}} d(p, q)$. The *$r$-gather clustering problem* asks to find an r-gather clustering such that the maximum cost among the costs of clusters is minimized, and such a clustering is called an optimal r-gather clustering. The following result is known [10]. Note that any cluster with $2r$ or more points can be divided into clusters so that each of which has at most $2r - 1$ points and at least r points.

Lemma 1 ([10]). *There is an optimal r-gather clustering in which each cluster has at most $2r - 1$ points.*

Let $C = \{c_1, c_2, \cdots, c_n\}$ be n customers and $F = \{f_1, f_2, \cdots, f_m\}$ be m possible locations for facilities located on a star. An *r-gathering* of C to F is an assignment A of C to F such that each facility has zero or at least r customers. A facility having one or more customers is called an *open facility*. We denote by F' the set of open facilities. $A(c)$ denotes the facility to which a customer c is assigned in an assignment A. The cost of a facility f, denoted by $cost(f)$, is $\max\{d(f,c_i)|A(c_i)=f\}$ if f has one or more customers, and is 0 if f has no customer. *The r-gathering problem* asks to find an r-gathering such that the maximum cost among the costs of facilities is minimized.

3 r-Gather Clustering on a Star

In this section we give an algorithm for r-gather clustering problem on a star. Let C be a set of points on a star $\mathcal{L}=\{l_1,l_2,\cdots,l_d\}$ of d rays with center o. We consider the set C as a union of d sets $C_1,C_2,\cdots,C_d$ where C_i is the set of customers on ray l_i. We have the following lemma.

Lemma 2. *There is an optimal r-gather clustering such that, for each C_i, the set of points in C_i assigned to multi-ray clusters is consecutive points on l_i including the nearest point to o.*

Proof. A pair c_m,c_s in C_i is called a reverse pair if c_m is assigned to a multi-ray cluster, c_s is assigned to a single-ray cluster, and $d(o,c_s)<d(o,c_m)$. Assume for a contradiction that A is an optimal r-gather clustering with the minimum number of reverse pairs but the number is not zero. Let c_s and c_m be a reverse pair in C_i with maximum $d(o,c_m)$. Let $\mathcal{C}_s$ and $\mathcal{C}_m$ be the clusters containing c_s and c_m, respectively. We have two cases.
Case 1: $\mathcal{C}_s$ has a point c in C_i with $d(o,c_m)<d(o,c)$.

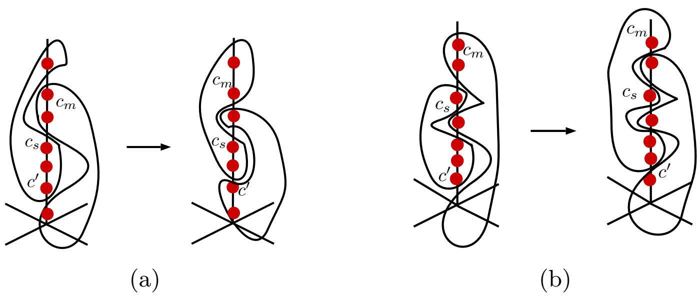

Fig. 2. (a) Illustration of Case 1 and (b) illustration of Case 2 of proof of Lemma 2.

Let c' be the nearest point to o in $\mathcal{C}_s$. Replacing $\mathcal{C}_s$ and $\mathcal{C}_m$ in the clustering by $\mathcal{C}_s\setminus\{c'\}\cup\{c_m\}$ and $\mathcal{C}_m\setminus\{c_m\}\cup\{c'\}$ generates a new r-gather clustering with less reverse pairs. A contradiction. Note that $cost(\mathcal{C}_s\setminus\{c'\}\cup\{c_m\})\le cost(\mathcal{C}_s)$ and $cost(\mathcal{C}_m\setminus\{c_m\}\cup\{c'\})\le cost(\mathcal{C}_m)$ hold.
Case 2: Otherwise. (Thus $d(o,c)<d(o,c_m)$ for every point c in $\mathcal{C}_s$.)
The same replacing results in a new r-gather clustering with less reverse pairs. A contradiction. Note that $cost(\mathcal{C}_s\setminus\{c'\}\cup\{c_m\})\le cost(\mathcal{C}_m)$ and $cost(\mathcal{C}_m\setminus\{c_m\}\cup\{c'\})\le cost(\mathcal{C}_m)$ hold. □

Lemma 3. *If an optimal r-gather clustering has multi-ray clusters, then at most one multi-ray cluster contains more than r points.*

Proof. Assume for a contradiction that every optimal r-gather clustering has two or more multi-ray clusters having more than r points. Let A be an r-gather clustering with the minimum number of multi-ray clusters having more than r points. Let $\mathcal{C}_i$ and $\mathcal{C}_j$ be two multi-ray clusters having more than r points. Let s_i, t_i be the two endpoints of $\mathcal{C}_i$ and s_j, t_j be the two endpoints of $\mathcal{C}_j$. Without loss of generality, assume that t_j is the closest point to o among the four end-points. Let $\mathcal{C}'_j \subset \mathcal{C}_j$ be $\{c \in \mathcal{C}_j | d(o,c) > d(o,t_j)\}$. Any point $c \in \mathcal{C}'_j$ must be on the same ray as s_j, otherwise t_j would not be an end-point of $\mathcal{C}_j$. We have two cases.

Case 1: $|\mathcal{C}'_j| < r$.

Let $\mathcal{C}''_j$ be a set of $|\mathcal{C}_j| - r$ arbitrary points from $\mathcal{C}_j \setminus \mathcal{C}'_j$. We now derive a new r-gather clustering A' by replacing $\mathcal{C}_i$ and $\mathcal{C}_j$ by $\mathcal{C}_i \cup \mathcal{C}''_j$ and $\mathcal{C}_j \setminus \mathcal{C}''_j$. Since t_j is the closest point to o among the four end-points s_i, t_i, s_j, t_j and $d(o,c) \leq d(o,t_j)$ for any point $c \in \mathcal{C}''_j$, we have $d(o,c) \leq d(o,s_i)$ and $d(o,c) \leq d(o,t_i)$. Thus the cost of $\mathcal{C}_i \cup \mathcal{C}''_j$ does not exceed the cost of $\mathcal{C}_i$. Hence the cost of A' is not greater than the cost of A. Thus A' has less multi-ray clusters with more than r points, a contradiction.

Case 2: Otherwise. Thus $|\mathcal{C}'_j| \geq r$.

In this case we derive a new r-gather clustering A' by replacing $\mathcal{C}_i$ and $\mathcal{C}_j$ by $\mathcal{C}_i \cup (\mathcal{C}_j \setminus \mathcal{C}'_j)$ and $\mathcal{C}'_j$. In this case, $\mathcal{C}'_j$ is a single-ray cluster. By a similar argument of Case 1, the cost of A' does not exceed the cost of A. Thus A' has less multi-ray clusters having more than r points than A, a contradiction. □

We now give the following lemma, which is used in the proof of Lemma 5 and Lemma 6.

Lemma 4. *If $|C| \geq 2r$ and there is an optimal r-gather clustering consisting of only multi-ray clusters, then there is an optimal r-gather clustering with the multi-ray cluster consisting of the farthest point from o and its $r-1$ nearest points.*

Proof. Let p be the farthest point from o and let N be the $r-1$ nearest points of p. Assume for a contradiction that in every optimal solution $N \cup \{p\}$ is not a cluster. We first prove that $N \cup \{p\}$ is contained in the same cluster. Let A be an optimal solution with cluster $\mathcal{C}_p$ containing p having the maximum number of points in N. Let q be a point in N assigned to a cluster $\mathcal{C}_q \neq \mathcal{C}_p$. Since the number of points in $\mathcal{C}_p$ is at least r, there is a point $p' \in \mathcal{C}_p$ not in N. Let q' be the farthest point from o in $\mathcal{C}_q \setminus \{q\}$. We now derive a new r-gather clustering by replacing $\mathcal{C}_p$ and $\mathcal{C}_q$ by $\mathcal{C}_p \setminus \{p'\} \cup \{q\}$ and $\mathcal{C}_q \setminus \{q\} \cup \{p'\}$. Thus a contradiction. Note that, $cost(\mathcal{C}_p \setminus \{p'\} \cup \{q\}) \leq cost(\mathcal{C}_p)$ and $cost(\mathcal{C}_q \setminus \{q\} \cup \{p'\}) \leq \max\{cost(\mathcal{C}_p), cost(\mathcal{C}_q)\}$, since $d(o,p) \geq d(o,q')$.

We now prove that $N \cup \{p\}$ form a multi-ray cluster. Assume for a contradiction that in any optimal r-gather clustering $N \cup \{p\}$ is not a cluster. Let A' be an optimal r-gather clustering with cluster $\mathcal{C}_p$ containing p having the minimum number of points not in N. Let p'' be the farthest point in $\mathcal{C}_p$ not in the ray l_p containing p, and $\mathcal{C}_s$ be a cluster in A' other than $\mathcal{C}_p$. Let s be the farthest point from o in $\mathcal{C}_s$. We now derive a new r-gathering by replacing $\mathcal{C}_p$ and $\mathcal{C}_s$ with $\mathcal{C}_p \setminus \{p''\}$ and $\mathcal{C}_s \cup \{p''\}$ without increasing cost, a contradiction. Since $d(o,s) \leq d(o,p)$, we have $d(s,p'') \leq d(p,p'')$ and thus $cost(\mathcal{C}_s \cup \{p''\}) \leq \max\{cost(\mathcal{C}_p), cost(\mathcal{C}_s)\}$. □

We now have the following lemma.

Lemma 5. *If an optimal r-gather clustering consists of only multi-ray clusters, then there is an optimal r-gather clustering with at most $d-1$ multi-ray clusters.*

Proof. We give a proof by induction on the number d of rays in the star. Clearly, the claim holds for $d = 2$, since in such case only one multi-ray cluster can exist. Assume that the claim holds for any star with less than d rays. We now prove that the claim also holds for any star of d rays. Assume for a contradiction that every optimal solution has at least d multi-ray clusters. Let A be an optimal r-gather clustering with the minimum number of multi-ray clusters. Let p be the farthest point from o. By Lemma 4, there is an optimal r-gather clustering with the cluster $\mathcal{C}_p$ containing p and its $r-1$ nearest points, denoted by N. Let l_p be the ray containing p. We have two cases.
Case 1: p and N are on ray l_p.
In this case there is an optimal r-gather clustering with a single ray cluster $N \cup \{p\}$, a contradiction.
Case 2: Otherwise. There is a point q in N which is not on l_p.
By Lemma 4 there is an optimal r-gathering with $\{p\} \cup N$, and since N consists of the $r-1$ nearest neighbors of p, all the points on l_p are contained in $\mathcal{C}_p$. Thus the points in $C \setminus \mathcal{C}_p$ are lying on other $d-1$ rays except l_p. By inductive hypothesis there is an optimal r-gather clustering of $C \setminus \mathcal{C}_p$ with at most $d-2$ multi-ray clusters. Thus the claim holds. □

Corollary 1. *If an optimal r-gahter clustering consists of only multi-ray clusters, then C has at most $(d-2)r + 2r - 1 = dr - 1$ points.*

We now give an outline of our algorithm which constructs an optimal r-gathering clustering on a star. We first choose every possible at most $dr-1$ candidate points for multi-ray clusters. We find the optimal r-gather clustering consisting of only multi-ray clusters for each candidate points, by repeatedly searching for the farthest point from o and its $r-1$ nearest point as a multi-ray cluster of the remaining set of points, by the algorithm **Multi-rayClusters**.

We now have the following lemma.

Lemma 6. *Let $A = \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \cdots, \mathcal{C}_{|A|}\}$ be the clusters computed by Algorithm **Multi-rayClusters**. If A has only multi-ray clusters, then A is an optimal r-gather clustering of S.*

Proof. The proof of this lemma is immediate from Lemma 4. □

We now give an algorithm r**GatherClusteringOnStar** to construct an optimal r-gather clustering of C on a star. We have the following theorem.

Algorithm 1: Multi-rayClusters(C)

```
Input  : A set C of points on a star
Output: An r-gather clustering with only multi-ray clusters
if |C| < r then
    return ∅;
endif
i ← 1;
while |C| ≠ 0 do
    if |C| < 2r then
        Create new cluster C_i = C;
    else
        p ← farthest point from o in C;
        C_i ← {p, p_1, p_2, ··· , p_{r−1}} where p_i is the i-th nearest point of p
          in C;
    endif
    C ← C \ C_i;
    i ← i + 1;
end
return {C_1, C_2, C_3, ··· , C_{i−1}}
```

Theorem 1. *The algorithm r**GatherClusteringOnStar** constructs an optimal r-gather clustering of C on star in* $O(rn + (r+1)^d dr)$ *time.*

Proof. We first prove the correctness of the algorithm. By Lemma 2 multi-ray clusters in an optimal r-gathering are located near o, and by Corollary 1 the number of customers in the multi-ray clusters is at most $dr - 1$. The algorithm r**GatherClusteringOnStar** considers every possible choice of the set of points for multi-ray clusters having at most $dr - 1$ points. The algorithm considers the solution for each possible choice for multi-ray clusters with the solution obtained by 1-dimensional algorithm for remaining points on each ray, and choose the solution having minimum cost. Thus the algorithm produces an optimal r-gather clustering.

We now prove that the algorithm runs in linear time. We consider points in each ray are in sorted order according to the distance from o. The d nested loops iterates $\prod_{j=1}^{d}(n_j + 1)$ times. Thus the number of points involved in all calls to Multi-rayClusters is at most $(r+1)^d dr$, since $\sum_{j=1}^{d} n_j = dr - 1$. Within each nested loop we repeatedly compute multi-ray clusters which takes linear time in total. We also compute single-ray clusters on each of the d rays. Rather than computing those single-ray clusters each time in the loop, we compute the r-gather clustering for points consisting of i farthest points from o, for each i, and for each ray in $O(rn)$ time total [10]. Thus to compute all the required cases for single-ray cluster we need total $O(rn)$ time. Thus the time complexity of the algorithm is $O(rn + (r+1)^d dr)$. □

If both r and d are constants then this is linear.

Algorithm 2: rGatherClusteringOnStar(C)

```
Input  : A set C of points on star L = {l_1, l_2, l_3, ..., l_d}
Output: An optimal r-gather clustering of C
if |C| < r then
    return ∅;
endif
Best ← ∅;
Let n_1, n_2, ..., n_d be the number of points of C in each ray l_1, l_2, ..., l_d;
for i_1 ← 0 to n_1 do
    for i_2 ← 0 to n_2 do
        for i_3 ← 0 to n_3 do
            ...;
            for i_d ← 0 to n_d do
                if i_1 + i_2 + ... + i_d < dr then
                    S be the set of points consisting of i_1, i_2, ..., i_d closest
                     points from o for ray l_1, l_2, ..., l_d;
                    R_m ← Multi-rayClusters(S);
                    R_i ← r-gather clustering of remaining points of ray l_i
                     by 1D algorithm;
                    R ← R_m ∪ R_1 ∪ R_2 ∪ ... ∪ R_d;
                    if R is the best r-gather clustering so far then
                        Best ← R;
                    endif
                endif
            end
            ...;
        end
        if i_1 + i_2 ≥ dr then
            break;
        endif
    end
    if i_1 ≥ dr then
        break;
    endif
end
return Best;
```

4 *r*-Gathering on a Star

In this section we give an algorithm for the r-gathering problem on a star.

Let C be a set of customers and F be a set of facilities on a star $\mathcal{L} = \{l_1, l_2, \cdots, l_d\}$ of d rays with center o. We regard the set C as the union of d sets $C_1, C_2, \cdots, C_d$ where C_i is the set of customers on ray l_i. Similarly, F is the union of $F_1, F_2, \cdots, F_d$ where F_i is the set of facilities on ray l_i. In any optimal r-gathering each open facility serves at least r customers. However the number of customers assigned to an open facility can be more than $2r - 1$. In such case we regard the set of customers assigned to a facility as the union of clusters $\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_k$ sharing a facility and each of which satisfies $r \leq |\mathcal{C}_i| < 2r$. Thus we can think of the r-gathering problem in a similar way to the r-gather clustering problem in Section 3, and Lemma 1 holds for the clusters of r-gathering. We denote by $A(\mathcal{C})$ the facility to which the customers in $\mathcal{C}$ is assigned in r-gathering A. We define the cost of a cluster $\mathcal{C}$, denoted by $cost(\mathcal{C})$, in r-gathering A as $\max_{c \in \mathcal{C}}\{d(c, A(c))\}$. It is easy to observe that Lemma 2 also holds for the clusters of r-gatherings. We now prove that Lemma 3 also holds for the r-gathering problem.

Lemma 7. *There is an optimal r-gathering including at most one multi-ray cluster having more than r customers.*

Proof. Omitted. □

A customer on a ray $l \in \mathcal{L}$ is the *boundary customer* of l if it is the farthest customer on l from o. We now give the following lemma.

Lemma 8. *If $|C| \geq 2r$ and there is an optimal r-gathering A with only multi-ray clusters, then there is an optimal r-gathering with only multi-ray clusters satisfying the following (a) and (b). Let f be the farthest open facility from o in A and l be the ray containing f.*
(a) The boundary customer p of l and its $r-1$ nearest customers form a multi-ray cluster, if l has a customer,
(b) All customers are assigned to f and the farthest boundary customer p from o and its $r-1$ nearest customers form a multi-ray cluster, if l has no customer.

Proof. (a) We denote by N the set of the $r-1$ nearest customers of p. We first prove that there is an optimal solution with the customers in $N \cup \{p\}$ are assigned to f. Assume for a contradiction that in any optimal solution $N \cup \{p\}$ are not assigned to f. Let A be an optimal solution with the maximum number of customers in $N \cup \{p\}$ are assigned to f. Let $\mathcal{C}_p$ be the multi-ray cluster assigned to f, and q be a customer in $N \cup \{p\}$ but $q \notin C_p$. Let q is assigned to f'. Since $\mathcal{C}_p$ has at least r customers, there is a customer $p' \in \mathcal{C}_p$ not in $N \cup \{p\}$ and lying on a ray except l. We now derive a new r-gathering A' by reassigning q to f and p' to f'. Since $d(o, f') \leq d(o, f)$, we have $d(f', p') \leq d(o, f') + d(o, p') \leq d(o, f) + d(o, p') = d(f, p')$. Now if q is (1) not on l or (2) q is on l with $d(o, q) \leq d(o, f)$ then $d(f, q) \leq d(f, p')$. Otherwise, q is on l with $d(o, q) > d(o, f)$ holds, then we have $d(f, q) \leq d(f, p)$. Thus the cost of A' does not exceed the cost of A, and A' has more customers in $N \cup \{p\}$ assigned to f. A contradiction. Thus the customers in $N \cup \{p\}$ are contained in $\mathcal{C}_p$.

We now prove that $N \cup \{p\}$ form a multi-ray cluster. Assume for a contradiction that in any optimal r-gathering $N \cup p$ is not a cluster. Let A' be an optimal r-gathering with the cluster $\mathcal{C}_p$ containing p having the minimum number of customers not in $N \cup \{p\}$. Since $\mathcal{C}_p$ is a multi-ray cluster, $\mathcal{C}_p$ has a customer s

not in $N \cup \{p\}$ and lying on a ray except l. We can reassign s to some open facility $f' \neq f$ without increasing the cost, since $d(o, f') \leq d(o, f)$, $d(s, f')$ does not exceed $d(s, f)$. A contradiction.

(b) We first prove that all customers are assigned to f. Assume for a contradiction that there is an open facility $f' \neq f$ to which some customers are assigned. Since f is the farthest open facility from o and there is no customer on l, we can reassign all customers to f' without increasing the cost of the r-gathering. A contradiction.

The proof of the 2nd part of Lemma 8(b) is similar to the proof of Lemma 8(a). □

We now prove that Lemma 5 also holds for r-gathering.

Lemma 9. *If an optimal r-gathering consists of only multi-ray clusters, then there is an optimal r-gathering consisting of at most $d-1$ multi-ray clusters, where d is the number of rays containing a customer.*

Proof. Omitted. □

We now give algorithm Multi-rayClusters2. If there is an optimal r-gathering with only multi-ray clusters, then the algorithm finds such an r-gathering, by repeatedly removing a cluster ensured by Lemma 8.

Lemma 10. *If there is an optimal r-gathering consisting of only multi-ray clusters, then Algorithm* **Multi-rayClusters2** *finds an optimal r-gathering. The running time of the algorithm is $O(2^d(r+d)d + (d+\log m)d)$.*

Proof. If there is an optimal r-gathering with only multi-ray clusters, then, by repeatedly removing a cluster ensured by Lemma 8, we can find a sequence $\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_k$ of multi-ray clusters such that $\mathcal{C}_i$ consists of exactly r customers in $C \setminus (\mathcal{C}_1 \cup \mathcal{C}_2 \cup \cdots \mathcal{C}_{i-1})$ except the last cluster $\mathcal{C}_k$ with $r \leq |\mathcal{C}_k| \leq 2r-1$. The algorithm checks every possible sequence of the rays containing the farthest open facility and chooses the best one as an optimal r-gathering. Note that if a cluster is a single-ray cluster, then the algorithm skips recursive call, since it try to find an r-gathering consisting of only multi-ray clusters.

We now estimate the running time of the algorithm.

By Lemma 9 the depth of the recursive calls is at most $d-1$. Thus, by the tree structure of the calls, the number of calls is at most $d!$. The algorithm repeatedly constructs a multi-ray cluster with exactly r customers by Lemma 8, which takes $O(r+d)$ time for each and $O(r+d)d$ time in total. The cluster is assigned to its best facility of the cluster. The best facility of a multi-ray cluster is the nearest facility to the mid-point of the farthest two customers on two different rays in the cluster. The best facility can be found in $O(d+\log m)$ time for each cluster. Thus the algorithm runs in $O(d!((r+d)d + (d+\log m))d)$ time.

We can improve the running time by modifying the algorithm to save the solution of each subproblem in a table. The number of distinct subproblems is the number of the combinations of the lines checked. Thus the number of distinct subproblem is $\sum_{j=1}^{d-1} \binom{d}{i} = O(2^d)$. Then the runtime is $O(2^d(r+d)d + (d+\log m)d)$. □

Algorithm 3: Multi-rayClusters2(C, F)

```
Input  : A set C of customers and a set of F of facilities on a star
Output: An r-gathering with only multi-ray clusters
if |C| < r or the number of rays containing customers is at most one
 then
  | return ∅;
endif
if |C| < 2r or the number of rays containing customers is two then
  | Assign C to its best facility; /* Lemma 8(b) */
  | return {C};
endif
Ans ← ∅;
Best ← ∞;
for each ray l_i containing a customer do
  | C_i ← p_i and its r − 1 nearest customers in C; /* Lemma 8(a) */
  | if C_i is a multi-ray cluster then
  |   | Assign C_i to its best facility;
  |   | A ← {C_i}∪ Multi-rayClusters2(C \ C_i, F);
  |   | if cost(A) < Best then
  |   |   | Best ← cost(A);
  |   |   | Ans ← A;
  |   | endif
  | endif
end
return Ans
```

Theorem 2. *An optimal r-gathering of C to F can be computed in $O(n+r^2m+d^2r^2(d+\log m)+(r+1)^d 2^d(r+d)d)$ time.*

Proof. Similar to Theorem 1 we can prove the number of possible choices of the customers for multi-ray clusters is at most $(r+1)^d dr$. For each choice we compute an r-gathering with Multi-rayClusters2 and compute r-gatherings of the remaining one-dimensional problems, then combine them to form an r-gathering of C to F. Then output the best one. This construction of multi-ray clusters needs $O(2^d(r+d)d+(d+\log m)d)$ for each. To eliminate redundant computation we precompute the best facilities of each pair in the dr customers which are candidate for the farthest two customers in possible multi-ray clusters. Such precomputation takes $O(d^2r^2(d+\log m))$ time. We can solve all possible one dimensional r-gathering problem in $O(n+r^2m)$ time in total [10] and we store the solutions in a table. Note that when we solve one dimensional r-gathering problem of ray l, we may assign a cluster to the nearest facility to o located on other ray, however one can compute such f quickly. Thus the time complexity of finding an optimal r-gathering is $O(n+r^2m+d^2r^2(d+\log m)+(r+1)^d 2^d(r+d)d)$. □

If both r and d are constant, then this is linear.

5 Conclusion

In this paper we presented an $O(rn + (r+1)^d dr)$ time algorithm to solve rhe r-gather clustering problem when all points are lying on a star with d rays. We also gave an $O(n + r^2 m + d^2 r^2 (d + \log m) + (r+1)^d 2^d (r+d)d)$ time algorithm to solve the r-gathering problem when all customers and facilities are lying on a star with d rays.
Can we solve the problems more efficiently or can we solve the problems on more general input like on a tree?

Acknowledgement. We thank CodeCrafters International and Investortools, Inc. for supporting the first author under the grant "CodeCrafters-Investortools Research Grant".

References

1. Agarwal, P.K., Sharir, M.: Efficient algorithms for geometric optimization. ACM Comput. Surv. 30(4), 412–458 (1998)
2. Aggarwal, G., Panigrahy, R., Feder, T., Thomas, D., Kenthapadi, K., Khuller, S., Zhu, A.: Achieving anonymity via clustering. ACM Trans. Algorithms 6(3), 49:1–49:19 (2010)
3. Akagi, T., Nakano, S.: On r-gatherings on the line. In: Wang, J., Yap, C. (eds.) Proceedings of Frontiers in Algorithmics. vol. 9130 of Lecture Notes in Computer Science, pp. 25–32. Springer International Publishing, Cham (2015)
4. Armon, A.: On min-max r-gatherings. Theor. Comput. Sci. 412(7), 573 – 582 (2011)
5. Frederickson, G.N., Johnson, D.B.: Generalized selection and ranking: Sorted matrices. SIAM J. Comput. 13(1), 14–30 (1984)
6. Gabow, H.N., Tarjan, R.E.: Algorithms for two bottleneck optimization problems. J. Algorithms 9(3), 411 – 417 (1988)
7. Guha, S., Meyerson, A., Munagala, K.: Hierarchical placement and network design problems. In: Proceedings 41st Annual Symposium on Foundations of Computer Science. pp. 603–612 (2000)
8. Han, Y., Nakano, S.: On r-gatherings on the line. In: Proceedings of FCS 2016. pp. 99–104 (2016)
9. Karget, D.R., Minkoff, M.: Building steiner trees with incomplete global knowledge. In: Proceedings 41st Annual Symposium on Foundations of Computer Science. pp. 613–623 (2000)
10. Nakano, S.: A simple algorithm for r-gatherings on the line. In: Rahman, M.S., Sung, W.K., Uehara, R. (eds.) Proceedings of WALCOM: Algorithms and Computation. vol. 10755 of Lecture Notes in Computer Science, pp. 1–7. Springer International Publishing, Cham (2018)
11. Zeng, J., Telang, G., Johnson, M.P., Sarkar, R., Gao, J., Arkin, E.M., Mitchell, J.S.B.: Mobile r-gather: Distributed and geographic clustering for location anonymity. In: Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing. pp. 7:1–7:10. Mobihoc '17, ACM (2017)