Yugoslav Journal of Operations Research
34 (2024), Number 2, 337–353
DOI: https://doi.org/10.2298/YJOR2023021K

FAST APPROXIMATION ALGORITHMS FOR SOME MAXIMIN CLUSTERING PROBLEMS

V. KHANDEEV
Sobolev Institute of Mathematics, 4 Koptyug Ave., Novosibirsk, Russia 630090
khandeev@math.nsc.ru

S. NESHCHADIM
Novosibirsk State University, 2 Pirogova St., Novosibirsk, Russia 630090
s.neshchadim@g.nsu.ru

Received: February 2023 / Accepted: November 2023

Abstract: In this paper, we consider three cases of an intractable problem of searching for two subsets in a finite set of points of Euclidean space. In all three cases, it is required to maximize the minimum cluster's cardinality under constraint on each cluster's scatter. The scatter is the sum of the distances from the cluster elements to the center, which is defined differently in each of the three cases. In the first case, cluster centers are fixed points. In the second case, the centers are unknown points from the input set. In the third case, the centers are defined as the centroids (the arithmetic mean) of clusters. We propose a general scheme that allows us to build a polynomial 1/2-approximation algorithm for a generalized problem and can be used for constructing 1/2-approximation algorithms for the first two cases and for the one-dimensional third case. Also we show how, using precomputed general information, their time complexities can be reduced to the complexity of sorting. Finally, we present the results of computational experiments showing the accuracy of the proposed algorithms on randomly generated input data.

Keywords: Euclidean space, clustering, max-min problem, NP-hardness, bounded scatter, approximation algorithm.

MSC: 90C27, 90C59.

Preliminary version of the paper was presented at the International conference Mathematical Optimization Theory and Operations Research (MOTOR 2022) [1].

1. INTRODUCTION

The subject of this paper is three cases of a problem of finding two disjoint subsets in a finite set of points of Euclidean space. This problem models the applied problems of searching for a family of objects in the case when each family consists of objects similar in the sense of some criterion. Such problem is often found in applications such as pattern recognition and machine learning [2], data analysis [3], data cleaning [4].

In all three considered cases, it is required to maximize the minimum cluster's cardinality so that in each cluster the total intra-cluster scatter relative to the center does not exceed the specified threshold. In each case, the center is determined in its own way.

NP-hardness was previously proved for all three cases, even in the subcase of a one-dimensional space (see the next section). This paper aims to construct fast approximation algorithms with guaranteed accuracy for the problem considered.

2. PROBLEM STATEMENTS AND SIMILAR PROBLEMS

In this paper we will use the following definitions.

Definition 1. *$\mathcal{P}(\mathbb{R}^d)$ is the set of all finite subsets of $\mathbb{R}^d$.*

Definition 2. *$F\left(\mathcal{C}, z\right)\colon \mathcal{P}(\mathbb{R}^d) \times \mathbb{R}^d \to \mathbb{R}_+$ is a function such that*

$$F\left(\mathcal{C}, z\right) := \sum_{y \in \mathcal{C}} \|y - z\|_2,$$

where $\|v\|_2 = \sqrt{v_1^2 + \ldots + v_d^2}$ for each $v = (v_1, \ldots, v_d) \in \mathbb{R}^d$.

Definition 3. *Centroid of a set $\mathcal{C} \in \mathcal{P}(\mathbb{R}^d)$ is a point $\bar{y}(\mathcal{C})$ from $\mathbb{R}^d$ such that*

$$\bar{y}(\mathcal{C}) = \frac{1}{|\mathcal{C}|} \sum_{y \in \mathcal{C}} y.$$

We consider three cases of the following clustering problem.

Bounded Sums-of-Distances Clustering ($\mathcal{Y}$, c_1, c_2, A).
Given an N-element set $\mathcal{Y} = \{y_1, \ldots, y_N\}$ of points of Euclidean space $\mathbb{R}^d$, centers c_1, c_2 (as described later), a non-negative number $A \in \mathbb{R}^+$.

Find non-empty disjoint subsets $\mathcal{C}_1, \mathcal{C}_2 \subset \mathcal{Y}$ such that the minimum subset size is maximal. In other words,

$$\min\left(|\mathcal{C}_1|, |\mathcal{C}_2|\right) \to \max, \tag{1}$$

where

$$F\left(\mathcal{C}_i, c_i\right) \le A,\ i = 1, 2. \tag{2}$$

Further we will refer to this problem as **BSD-CLUSTER**. Three considered cases differ by the type of centers. In the first case, the centers c_1 and c_2 are equal to the given points $z_1, z_2 \in \mathbb{R}^d$, respectively — problem **BSD-CLUSTER**($\mathcal{Y}$, z_1, z_2, A). In the second case, the centers are unknown and are to be found, but must be from the initial set. In this case we will denote the problem as **BSD-CLUSTER**($\mathcal{Y}$, $*$, $*$, A). Finally, in the third case, the centers are equal to the centroids $\bar{c}_i = \bar{y}(\mathcal{C}_i)$, $i = 1, 2$, and the problem will be denoted as **BSD-CLUSTER**($\mathcal{Y}$, $\bar{c}_1$, $\bar{c}_2$, A).

The considered problem has the following interpretation. There is a group of objects, among which there are two disjoint subgroups of similar objects. Also, there are some "outliers", which are objects that do not belong to either of the two subgroups. It is known that each subgroup has its own reference object and that in a subgroup the scatter relative to this reference object does not exceed a certain value. The scatter is defined as the sum of measures of "dissimilarity" of objects within the subgroup and the corresponding reference object. Given this constraint, it is required to find the largest subgroups of similar objects.

Similar clustering problems, in which the cardinalities of the sought subsets are maximized, can be found, for example, in [5], [6], [7], [8]. Like **BSD-CLUSTER**, the problems considered in these papers model the search for homogeneous subsets, which is a typical problem for editing [9] and clearing [4] data. However, we note that despite the similarity of the statements, the algorithms proposed in [5], [6], [7], [8] are not applicable to **BSD-CLUSTER**.

Previously, it was proved [10] that all three described cases of **problem BSD-CLUSTER** are NP-hard even in the one-dimensional case. In addition, it is known [11] that this property also holds for quadratic analogs of the first two cases, that is, for problems in which the squares of distances are summed in (2).

In [1], $\frac{1}{2}$-approximation algorithms for the problem **BSD-CLUSTER** with fixed and unknown centers and for the one-dimensional case of problem with centroids are proposed. The running time of the algorithms for the case of unknown centers is $\mathcal{O}(N^3 \log N)$ and for the other two cases is $\mathcal{O}(N^2 \log N)$.

In this paper, we present modifications of these algorithms that allow us to speed up each of them by a factor of N.

The paper has the following structure. Section 3 provides a formulation of the problem that generalizes considered problem and justifies an approach that allows one to find a $\frac{1}{2}$-approximate solution to the general problem. In Section 4, we show the complexity of a simple approximation algorithm constructed in this way for problem with fixed centers and present an approach that allows us to speed up this algorithm by eliminating unnecessary operations. Further, in Sections 5 and 6, the existence of similar algorithms for the problem with unknown centers and the one-dimensional case of the problem with centroids are shown. Finally, in Section 7, we present the results of computational experiments showing the accuracy of the proposed algorithms on randomly generated input data.

3. GENERALIZED PROBLEM

Let us present a general approach to solving the considered problem, that is not based on the specifics of the scatter function.

By d-dimensional scatter function we will understand an arbitrary function $\mathcal{F} : \mathcal{P}(\mathbb{R}^d) \to \mathbb{R}_+$ such that for any $\mathcal{C}' \subset \mathcal{C}'' \subset \mathbb{R}^d$ inequality $F(\mathcal{C}') \leq F(\mathcal{C}'')$ holds.

Now, let us consider the following general problem of finding two clusters of the given size with a limited scatter. The scatter of those clusters will be determined by two arbitrary scatter functions.

Problem CLUSTER($\mathcal{Y}$, $\mathcal{F}_1$, $\mathcal{F}_2$, A, M). *Given* a set $\mathcal{Y} = \{y_1, \ldots, y_N\} \subset \mathbb{R}^d$, d-dimensional scatter functions $\mathcal{F}_1, \mathcal{F}_2$, a non-negative number $A \in \mathbb{R}^+$, and a positive integer $M \in \mathbb{N}$. *Find* non-empty disjoint subsets $\mathcal{C}_1, \mathcal{C}_2 \subset \mathcal{Y}$ with cardinalities equal to M such that

$$\mathcal{F}_i\left(\mathcal{C}_i\right) \leq A,\ i = 1, 2,$$

or prove that they do not exist.

Since the problem formulated in Section 2 is the problem of finding two clusters of the maximum size rather than the given one, let us consider respective modification of **CLUSTER** problem. In this problem, we will denote the last argument as $M_{\max}$ (instead of a given cluster size M).

Problem CLUSTER($\mathcal{Y}$, $\mathcal{F}_1$, $\mathcal{F}_2$, A, $M_{\max}$). *Given* a set $\mathcal{Y} = \{y_1, \ldots, y_N\} \subset \mathbb{R}^d$, d-dimensional scatter functions $\mathcal{F}_1, \mathcal{F}_2$, and a non-negative number $A \in \mathbb{R}^+$. *Find* non-empty disjoint subsets $\mathcal{C}_1, \mathcal{C}_2 \subset \mathcal{Y}$ such that the minimum subset size is maximal:

$$\min\left(|\mathcal{C}_1|, |\mathcal{C}_2|\right) \to \max,$$

where

$$\mathcal{F}_i\left(\mathcal{C}_i\right) \leq A,\ i = 1, 2.$$

Note that problem **CLUSTER**($\mathcal{Y}$, $\mathcal{F}_1$, $\mathcal{F}_2$, A, $M_{\max}$) (which we will refer to as maximin **CLUSTER** problem) is a generalization of three cases described in Section 2. Indeed, problem with fixed centers is equivalent to problem **CLUSTER**($\mathcal{Y}$, $F\left(\mathcal{C}, z_1\right)$, $F\left(\mathcal{C}, z_2\right)$, A, $M_{\max}$). Problem with unknown centers is reduced (see Section 5) to problem **CLUSTER**($\mathcal{Y}$, F_2^M, F_2^M, A, $M_{\max}$), where

$$F_2^M(\mathcal{C}) = \min_{u \in \mathcal{Y}} F(\mathcal{C}, u).$$

Finally, problem with centroids is the same as problem **CLUSTER**($\mathcal{Y}$, $F(\mathcal{C}, \bar{y}(\mathcal{C}))$, $F(\mathcal{C}, \bar{y}(\mathcal{C}))$, A, $M_{\max}$).

We also consider an analog of problem **CLUSTER** in which it is required to find only one subset of $\tilde{\mathcal{Y}} \subseteq \mathcal{Y}$ of given cardinality with the minimal scatter. In this problem, we will denote the second last argument as $A_{\min}$ (instead of a given scatter bound A).

Problem CLUSTER($\tilde{\mathcal{Y}}$, $\mathcal{F}$, $A_{\min}$, M). *Given* a set $\tilde{\mathcal{Y}} = \{y_1, \ldots, y_k\} \subset \mathbb{R}^d$, a d-dimensional scatter function $\mathcal{F}$ and a positive integer $M \in \mathbb{N}$. *Find* M-element subset $\mathcal{C}$ of $\tilde{\mathcal{Y}}$ with minimal scatter $\mathcal{F}(\mathcal{C})$.

Let us consider two arbitrary scatter functions $\mathcal{F}_1$, $\mathcal{F}_2$ and assume that there is an algorithm that allows us to find optimal solutions to one-cluster problems **CLUSTER**($\tilde{\mathcal{Y}}$, $\mathcal{F}_i$, $A_{\min}$, M), $i = 1, 2$. Then we can propose the following algorithm for solving problem **CLUSTER**($\mathcal{Y}$, $\mathcal{F}_1$, $\mathcal{F}_2$, A, M).

Essentially, in this algorithm, firstly we try to find the best subset in terms of the first scatter function, and then among the remaining elements — the best subset in terms of the second scatter function. If the found subsets do not satisfy the scatter constraints, then we repeat the same procedure, but in a different order.

```
Algorithm  $\mathcal{A}(\mathcal{Y}, \mathcal{F}_1, \mathcal{F}_2, A, M)$
Input: $\mathcal{Y} \subset \mathbb{R}^d$, scatter functions $\mathcal{F}_1$, $\mathcal{F}_2$, $A \in \mathbb{R}_+$, $M \in \mathbb{N}$.
1: If $2M > N$, terminate the algorithm (no solution is constructed).
2: Find the optimal solution $\mathcal{C}_1$ to problem CLUSTER($\mathcal{Y}$, $\mathcal{F}_1$, $A_{\min}$, $M$), then
   find the optimal solution $\mathcal{C}_2$ to problem CLUSTER($\mathcal{Y} \setminus \mathcal{C}_1$, $\mathcal{F}_2$, $A_{\min}$, $M$).
   If $\mathcal{F}_1(\mathcal{C}_1) > A$ or $\mathcal{F}_2(\mathcal{C}_2) > A$, go to Step 3. Otherwise, go to Step 4 (a feasible
   solution with minimal cardinality $M$ has been constructed).
3: Find the optimal solution $\mathcal{C}_2$ to problem CLUSTER($\mathcal{Y}$, $\mathcal{F}_2$, $A_{\min}$, $M$), then
   find the optimal solution $\mathcal{C}_1$ to problem CLUSTER($\mathcal{Y} \setminus \mathcal{C}_2$, $\mathcal{F}_1$, $A_{\min}$, $M$).
   If $\mathcal{F}_1(\mathcal{C}_1) > A$ or $\mathcal{F}_2(\mathcal{C}_2) > A$, terminate the algorithm (no solution is con-
   structed). Otherwise, go to Step 4 (a feasible solution has been constructed).
4: Return the sets $\mathcal{C}_1$ and $\mathcal{C}_2$ as the result of the algorithm.
```

Remark 4. *It is worth mentioning that if the scatter functions $\mathcal{F}_1$ and $\mathcal{F}_2$ are equal at the input of the algorithm $\mathcal{A}(\ldots, A, M)$, then Step 3 can be skipped, because, in this case, Step 2 and Step 3 either both find no solution or find the same solution.*

Let us consider an arbitrary instance of maximin problem **CLUSTER**($\mathcal{Y}$, $\mathcal{F}_1$, $\mathcal{F}_2$, A, $M_{\max}$), where $\mathcal{Y} \subset \mathbb{R}^d$; $\mathcal{F}_1$, $\mathcal{F}_2$ are d-dimensional scatter functions; $A \in \mathbb{R}_+$. Let $\mathcal{C}_1^*$, $\mathcal{C}_2^*$ be an arbitrary feasible solution to this problem. Denote the minimal cardinality of $\mathcal{C}_1^*$ and $\mathcal{C}_2^*$ by M^*. Then the following statements hold true (in [1], the first two of them are intermediate results of the proofs of Proposition 1 and Theorem 1, respectively).

Proposition 5. *If M^* is even, i.e., $M^* = 2K^*$, the algorithm $\mathcal{A}(\ldots, A, M)$ applied to problem* ***CLUSTER****($\mathcal{Y}$, $\mathcal{F}_1$, $\mathcal{F}_2$, A, K^*) will construct a feasible solution.*

Proof. Using the existence of a solution with minimum cardinality M^* and monotonicity of the scatter function, it can be shown that at the second step of the algorithm $\mathcal{A}(\mathcal{Y}, \mathcal{F}_1, \mathcal{F}_2, A, M)$, a feasible solution will be constructed, since when constructing a solution to a one-cluster problem, there will always be at least K^* available elements from each solution's cluster with minimal cardinality $M^* = 2K^*$. □

Proposition 6. *If M^* is odd, i.e., $M^* = 2K^* + 1$, the algorithm $\mathcal{A}(\dots, A, M)$ applied to problem **CLUSTER**($\mathcal{Y}$, $\mathcal{F}_1$, $\mathcal{F}_2$, A, $K^* + 1$) will construct a feasible solution.*

Proof. Suppose that considered algorithm $\mathcal{A}$ didn't construct a feasible solution with sets of cardinality of $K^* + 1$. Let $\mathcal{C}_1^{**}$ be the set constructed at Step 2. Its scatter is not greater than A ($\mathcal{F}_1(\mathcal{C}_1^{**}) \le A$), since any $(K^* + 1)$-element subset of $\mathcal{C}_1^*$ is a feasible solution to Problem **CLUSTER**($\mathcal{Y}$, $\mathcal{F}_1$, $A_{\min}$, $K^* + 1$) and has a scatter of no more than A. Then, let $\mathcal{C}_2^{**}$ be the second set constructed at Step 2. From our assumption that no feasible solution was found, it follows that the pair $\mathcal{C}_1^{**}$ and $\mathcal{C}_2^{**}$ is not a feasible solution. Using that, we will show that $\mathcal{C}_1^{**}$ is a subset of $\mathcal{C}_2^*$.

Suppose the opposite is true: there is an element from $\mathcal{C}_1^{**}$ that is not contained in $\mathcal{C}_2^*$.

Then the set $\mathcal{C}_2^* \setminus \mathcal{C}_1^{**}$ contains at least $K^* + 1$ elements and its scatter does not exceed A. However, since $\mathcal{C}_2^* \setminus \mathcal{C}_1^{**} \subset \mathcal{Y} \setminus \mathcal{C}_1^{**}$, then any $(K^* + 1)$-element subset of $\mathcal{C}_2^* \setminus \mathcal{C}_1^{**}$ is a feasible solution to Problem **CLUSTER**($\mathcal{Y} \setminus \mathcal{C}_1^{**}$, $\mathcal{F}_2$, $A_{\min}$, $K^* + 1$) and has a scatter of no more than A. Therefore, $\mathcal{C}_2^{**}$ has a scatter of no more than A, which contradicts the fact that pair $\mathcal{C}_1^{**}$ and $\mathcal{C}_2^{**}$ is not a feasible solution to maximin **CLUSTER** problem. Hence, $\mathcal{C}_1^{**} \subset \mathcal{C}_2^*$.

Since no solution was found at Step 2, the algorithm will proceed to Step 3.

By analogy with Step 2, the constructed set $\mathcal{C}_2^{***}$ has a scatter not exceeding A, therefore the second set will be constructed further. Let $\mathcal{C}_1^{***}$ and $\mathcal{C}_2^{***}$ be the sets of cardinality $K^* + 1$ which are constructed at Step 3. According to our assumption, a pair $\mathcal{C}_1^{***}$ and $\mathcal{C}_2^{***}$ of sets is not a feasible solution. Then, by analogy with the reasoning for Step 2, we get that $\mathcal{C}_2^{***} \subset \mathcal{C}_1^*$.

The sets $\mathcal{C}_1^{**}$ and $\mathcal{C}_2^{***}$ don't intersect because they are subsets of the sets $\mathcal{C}_2^*$ and $\mathcal{C}_1^*$, respectively, that do not intersect by assumption. Hence, $\mathcal{C}_2^{***} \subset \mathcal{Y} \setminus \mathcal{C}_1^{**}$ and Problem **CLUSTER**($\mathcal{Y} \setminus \mathcal{C}_1^{**}$, $\mathcal{F}_2$, $A_{\min}$, $K^* + 1$) has a feasible solution $\mathcal{C}_2^{***}$ with a scatter of no more than A and a scatter of optimal solution $\mathcal{C}_2^{**}$ does not exceed A. As a result, we get that $\mathcal{C}_2^{**}$ satisfies the restriction on the scatter and also have the cardinality of $K^* + 1$. It follows from this that at Step 2, a feasible solution for maximin **CLUSTER** problem is constructed with the sets of $K^* + 1$ cardinality, which contradicts our assumption. □

Proposition 7. *If there is an algorithm that solves problem **CLUSTER**($\tilde{\mathcal{Y}}$, $\mathcal{F}$, $A_{\min}$, M) in time $\mathcal{O}(T(k))$, where $k = |\tilde{\mathcal{Y}}|$, then algorithm $\mathcal{A}(\dots, A, M)$ can be implemented in $\mathcal{O}(T(N))$ time, where $N = |\mathcal{Y}|$.*

Proof. It is enough to note that in the considered algorithm, the search for solutions to one-cluster problems is performed no more than four times. □

Now we can construct an approximation algorithm for maximin problem **CLUSTER**($\mathcal{Y}$, $\mathcal{F}_1$, $\mathcal{F}_2$, A, $M_{\max}$) using $\mathcal{A}(\dots, A, M)$. One possible approach is to start from $M = 1$ and increase M by one until the algorithm for problem **CLUSTER**($\mathcal{Y}$, $\mathcal{F}_1$, $\mathcal{F}_2$, A, M) finds no solution. However, the running time can be reduced by using a binary search by the value of M. It suffices to start with the boundaries $M = 1$ (for which a solution must exist) and $M = \lceil \frac{N}{2} \rceil + 1$ (there are no solutions with such cardinality). Formally, the algorithm can be written as follows.

```
Algorithm  $\mathcal{A}(\mathcal{Y}, \mathcal{F}_1, \mathcal{F}_2, A, M_{\max})$
Input: $\mathcal{Y} \subset \mathbb{R}^d$, scatter functions $\mathcal{F}_1$, $\mathcal{F}_2$, $A \in \mathbb{R}_+$.
1: Let $M_f = 1$, $M_t = \lceil \frac{N}{2} \rceil + 1$. Construct a solution $\mathcal{C}_1$, $\mathcal{C}_2$ to problem CLUSTER($\mathcal{Y}$, $\mathcal{F}_1$, $\mathcal{F}_2$, $A$, $M$) for $M = 1$. If there is no solution, then terminate the algorithm (maximin CLUSTER problem has no solutions).
2: If $M_f + 1 = M_t$, then go to Step 4, otherwise go to Step 3.
3: Let $M = \lceil \frac{M_f + M_t}{2} \rceil$. Construct a solution to problem CLUSTER($\mathcal{Y}$, $\mathcal{F}_1$, $\mathcal{F}_2$, $A$, $M$). If the solution has been constructed, then put $M_f = M$ and store the constructed solution in $\mathcal{C}_1$, $\mathcal{C}_2$ (an approximate solution of cardinality $M$ has been constructed); otherwise put $M_t = M$. Go to Step 2.
4: Return the sets $\mathcal{C}_1$ and $\mathcal{C}_2$ as the result of the algorithm.
```

The accuracy of algorithm $\mathcal{A}(\ldots, A, M_{\max})$ is established by the following proposition.

Proposition 8. *Algorithm $\mathcal{A}(\ldots, A, M_{\max})$ constructs a $\frac{1}{2}$-approximate solution to problem **CLUSTER**($\mathcal{Y}$, $\mathcal{F}_1$, $\mathcal{F}_2$, A, $M_{\max}$).*

Proof. Let $\mathcal{C}_1^*$, $\mathcal{C}_2^* \subset \mathcal{Y}$ be the optimal solution to maximin **CLUSTER** problem and $M^* = \min(|\mathcal{C}_1^*|, |\mathcal{C}_2^*|)$. It means there are feasible solutions with cardinalities from 1 to M^* since the scatter function is monotonous. Then it follows from Propositions 5 and 6 that algorithm $\mathcal{A}(\ldots, A, M)$ is able to construct feasible solutions for $M = 1, \ldots, \lceil \frac{M^*}{2} \rceil$.

Clearly, during the execution of Steps 2–4 of the algorithm, the variable M_f always contains the cardinality for which algorithm $\mathcal{A}(\ldots, A, M)$ constructs a solution and the variable M_t always contains the cardinality for which the algorithm finds no solution.

After $\mathcal{O}(\log N)$ iterations of Step 3, the algorithm will reach the case where $M_f + 1 = M_t$. After that, the algorithm will stop and return a solution with sets of cardinality M_f. We need to show that $2 * M_f \geq M^*$. Let us assume that this is not the case, i.e., $M_f < \frac{M^*}{2}$. But in this case $M_t = M_f + 1 \leq \lceil \frac{M^*}{2} \rceil$. However, we have previously shown that for all cardinalities not exceeding $\lceil \frac{M^*}{2} \rceil$, the algorithm constructs a feasible solution, which contradicts the fact that M_t contains the cardinality for which the algorithm does not find a solution. Thus, we have obtained a contradiction, and hence $2 * M_f \geq M^*$, i.e., the result of the algorithm will indeed be a $\frac{1}{2}$-approximate solution. □

The time complexity of algorithm $\mathcal{A}(\ldots, A, M_{\max})$ is established by the following proposition.

Proposition 9. *If algorithm $\mathcal{A}(\ldots, A, M)$ finds the solution to problem **CLUSTER** in $\mathcal{O}(T(N))$ time, then the total time complexity of the original algorithm $\mathcal{A}(\ldots, A, M_{\max})$ is $\mathcal{O}(T(N)\log N)$.*

Proof. The number of repetitions of Steps 2 and 3 of algorithm $\mathcal{A}(\ldots, A, M_{\max})$ is determined by the complexity of the binary search on the interval $[1, \lceil \frac{N}{2} \rceil + 1]$ and can be estimated by $\mathcal{O}(\log N)$. By the assumption of the proposition, at each repetition the number of operations is equal to $\mathcal{O}(T(N))$, so the total complexity of the $\mathcal{A}(\ldots, A, M_{\max})$ algorithm is $\mathcal{O}(T(N)\log N)$. □

So, this section shows that having an exact algorithm for solving one-cluster problem with $T(N)$ complexity for a certain scatter function, we can construct a $\frac{1}{2}$-approximate solution to maximin **CLUSTER** problem with the same scatter functions with complexity $\mathcal{O}(T(N)\log N)$. Described in this paper approach differs from the approach from [1] in that binary search is being used instead of linear search. That allows us to run the proposed algorithm not in $T(N) * N$, but in $T(N) * \log N$ time. In the following sections, we will propose additional modifications that would allow us to eliminate unnecessary operations in the process of solving one-cluster problems and obtain a $\frac{1}{2}$-approximate solution to the original problem with a time complexity that equals to the time of sorting the initial set according to the scatter functions.

4. ALGORITHM FOR PROBLEM WITH FIXED CENTERS

4.1. Exact algorithm for one-cluster problem

Let us consider problem **CLUSTER** and its modifications produced by **BSD-CLUSTER** with fixed centers. One can use the following simple approach to solving these problems.

Proposition 10. *Problem **CLUSTER**($\tilde{\mathcal{Y}}$, $\mathcal{F}$, $A_{\min}$, M), where $\mathcal{F}(\mathcal{C}) = \sum_{y\in\mathcal{C}} \|y - z\|$, is solvable in time $\mathcal{O}(k(d + \log k))$, where $k = |\tilde{\mathcal{Y}}|$.*

Proof. It suffices to calculate distances between points of $\tilde{\mathcal{Y}}$ and the point z, sort the elements of the set $\tilde{\mathcal{Y}}$ in non-decreasing order of this distance, and choose the first M elements of the sorted set. □

Then, according to Propositions 8 and 9, one can construct an algorithm that finds a $\frac{1}{2}$-approximate solution to maximin **CLUSTER** problem in $\mathcal{O}(N\log N(d+\log N))$.

Note that with this approach, unnecessary operations are performed to sort the original set. Indeed, in the $\mathcal{A}(\ldots, A, M_{\max})$ algorithm, the $\mathcal{A}(\ldots, A, M)$ algorithm is repeatedly called, which uses the algorithm from Proposition 10 for solving one-cluster problem. But the input set (or its subset) is being sorted each time in non-decreasing distance to the point z_1 or z_2. Since the number of calls to the algorithm $\mathcal{A}(\ldots, A, M)$ is estimated from above by $\mathcal{O}(\log N)$, the number of sorts of the input set is estimated by the same value.

In order to preserve the simplicity of solving one-cluster problem, but to reduce the running time of the final algorithm, we propose the following accelerated approach which is still based on sorting the input set.

Consider an arbitrary N-element set $\mathcal{Y} = \{y_1, \ldots, y_N\}$, $y_i \in \mathbb{R}^d$, $i = 1, \ldots, N$. Denote y_i by $\mathcal{Y}[i]$. First, in order not to recalculate the distances between the elements of the input set and the point z, these distances can be calculated and stored as a sequence $\mathcal{D}$ of length N, where

$$\mathcal{D}[j] = \|\mathcal{Y}[j] - z\|,\ 1 \le j \le N. \tag{3}$$

In addition, it is necessary to sort the elements of the input set by non-decreasing order of distances to the point z and store the indices of the elements of the sorted set as a sequence $\mathcal{Z}$. Thus, we will assume that $\mathcal{Z}$ is a sequence of numbers from 1 to N such that

$$\mathcal{D}[\mathcal{Z}[m]] \le \mathcal{D}[\mathcal{Z}[n]],\ 1 \le m < n \le N. \tag{4}$$

Finally, instead of the input subset $\tilde{\mathcal{Y}} \subseteq \mathcal{Y}$ in the one-cluster problem, we will use its characteristic vector, i.e., a sequence $\mathcal{M}$ of zeros and ones of length N such that

$$\tilde{\mathcal{Y}} = \{\mathcal{Y}[j] \mid \mathcal{M}[j] = 0,\ 1 \le j \le N\}. \tag{5}$$

In other words, the sequence $\mathcal{M}$ will define a set of “prohibitions” — a set of indices of elements of the sequence $\mathcal{Y}$ that cannot be used in valid solutions of one-cluster problem.

Using such structures, it is easy to find an optimal solution to the one-cluster problem where the input set is a subset $\tilde{\mathcal{Y}} \subseteq \mathcal{Y}$ in $\mathcal{O}(N)$ time — just choose M of the first elements of $\mathcal{Z}$ that are allowed by the $\mathcal{M}$ sequence. We will denote the corresponding algorithm by $\mathcal{A}_{z_1,z_2}\ (\mathcal{Z}, \mathcal{D}, \mathcal{M}, A_{\min}, M)$. Also, for convenience, we assume that the output of the algorithm is a sequence $\mathcal{B}$ of zeros and ones such that $\mathcal{C}^* = \{\mathcal{Y}[j] \mid \mathcal{B}[j] = 1,\ 1 \le j \le N\}$ is the optimal solution to the problem **CLUSTER**($\tilde{\mathcal{Y}}$, $\mathcal{F}$, $A_{\min}$, M), where $\mathcal{F}(\mathcal{C}) = \sum\limits_{y \in \mathcal{C}} \|y - z\|$.

4.2. Approximation algorithm for problem with fixed centers

Using the modified algorithm for one-cluster problem, let us formulate a modified algorithm for **CLUSTER** problem.

```
Algorithm $\mathcal{A}_{z_1,z_2}(\mathcal{Z}_1, \mathcal{Z}_2, \mathcal{D}_1, \mathcal{D}_2, A, M)$
Input: sequences $\mathcal{Z}_1$, $\mathcal{Z}_2$ of numbers from 1 to $N$ and sequences $\mathcal{D}_1$, $\mathcal{D}_2$ of real numbers such that
    $\mathcal{D}_i[\mathcal{Z}_i[m]] \leq \mathcal{D}_i[\mathcal{Z}_i[n]],\ 1 \leq m < n \leq N,\ i = 1, 2;$
    $A \in \mathbb{R}$; integer number $M \leq N$.
1: If $2M > N$, then terminate the algorithm (no solution is constructed).
2: In $\mathcal{M}_1$ write the result of the algorithm $\mathcal{A}_{z_1,z_2}$ $(\mathcal{Z}_1, \mathcal{D}_1, \mathcal{M}_0, A_{\min}, M)$, where $\mathcal{M}_0$ is the zero-filled $N$-element sequence. In $\mathcal{M}_2$ write the result of the $\mathcal{A}_{z_1,z_2}$ $(\mathcal{Z}_2, \mathcal{D}_2, \mathcal{M}_1, A_{\min}, M)$, algorithm.
   If $\sum_{j=1}^{N} \mathcal{M}_1[j] * \mathcal{D}_1[j] > A$ or $\sum_{j=1}^{N} \mathcal{M}_2[j] * \mathcal{D}_2[j] > A$, then go to Step 3. Otherwise, go to Step 4.
3: In $\mathcal{M}_2$ write the result of the algorithm $\mathcal{A}_{z_1,z_2}$ $(\mathcal{Z}_2, \mathcal{D}_2, \mathcal{M}_0, A_{\min}, M)$, where $\mathcal{M}_0$ is the zero-filled $N$-element sequence. In $\mathcal{M}_1$ write the result of the $\mathcal{A}_{z_1,z_2}$ $(\mathcal{Z}_1, \mathcal{D}_1, \mathcal{M}_2, A_{\min}, M)$, algorithm.
   If $\sum_{j=1}^{N} \mathcal{M}_1[j] * \mathcal{D}_1[j] > A$ or $\sum_{j=1}^{N} \mathcal{M}_2[j] * \mathcal{D}_2[j] > A$, then terminate the algorithm (no solution is constructed). Otherwise, go to Step 4.
4: Form the sets $\mathcal{C}_1$ and $\mathcal{C}_2$ according to the formulas
       $\mathcal{C}_i = \{\mathcal{Y}[j] \mid \mathcal{M}[j] = 1,\ 1 \leq j \leq N\},\ i = 1, 2,$
   and return them as a result.
```

Proposition 11. *For **BSD-CLUSTER**($\mathcal{Y}$, z_1, z_2, A) with fixed centers, there is a $\frac{1}{2}$-approximation algorithm with time complexity $\mathcal{O}(N(d + \log N))$.*

Proof. From the equality $G_i = \sum_{j=1}^{N} \mathcal{M}_i[j] * \mathcal{D}_i[j] = \mathcal{F}_i(\mathcal{C}_i)$ it follows that the $\mathcal{A}_{z_1,z_2}(\ldots, A, M)$ algorithm is a formal description of the $\mathcal{A}(\ldots, A, M)$ algorithm in the case when the scatter function corresponds to **BSD-CLUSTER**($\mathcal{Y}$, z_1, z_2, A). Therefore, from Proposition 8 follows the statement that if the $\mathcal{A}_{z_1,z_2}(\ldots, A, M)$ algorithm is used in the $\mathcal{A}(\ldots, A, M_{\max})$ algorithm to solve problem **CLUSTER**$(\ldots, A, M)$, then the result solution will be a $\frac{1}{2}$-approximate solution to **BSD-CLUSTER** with fixed centers.

Let us estimate the complexity of this approach. To form the sequences $\mathcal{Z}_1, \mathcal{Z}_2, \mathcal{D}_1, \mathcal{D}_2$, it is necessary to sort the points of the initial set (according to the distances to points z_1 and z_2), which requires $\mathcal{O}(N(d + \log N))$ operations. The complexity of the $\mathcal{A}_{z_1,z_2}(\ldots, A, M)$ algorithm is $\mathcal{O}(N)$, since each of its steps requires no more than $\mathcal{O}(N)$ operations. And since the algorithm for problem **CLUSTER**$(\ldots, A, M)$ is called $\mathcal{O}(\log N)$ times (see Proposition 9), the total complexity is equal to $\mathcal{O}(N(d + \log N)) + \mathcal{O}(\log N)\mathcal{O}(N) = \mathcal{O}(N(d + \log N))$. □

Remark 12. *Note that the execution of both Steps 2 and 3 of the* $\mathcal{A}(\ldots, A, M)$ *algorithm is necessary to obtain a* $\frac{1}{2}$*-approximate solution to problem with fixed centers. Indeed, consider the following one-dimensional instance:*

$$\mathcal{Y} = \{0, 8, 12, 43, 96, 99\},\ A = 156,\ z_1 = 73,\ z_2 = 112.$$

For this instance, there is a valid solution with a minimum cluster size of three:

$$\mathcal{C}_1 = \{8, 12, 43\},\ \mathcal{C}_2 = \{0, 96, 99\},\ \max\{\mathcal{F}(\mathcal{C}_1, z_1), \mathcal{F}(\mathcal{C}_2, z_2)\} = 156.$$

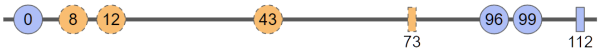

Figure 1: The instance of problem with fixed centers and its valid solution

However, if only Step 2 of the algorithm is executed, then in the end we will not be able to obtain a feasible solution with clusters of size 2, since the set $\mathcal{C}_1$ *will be equal to* $\{96, 99\}$ *and* $\mathcal{C}_2$ *will be equal to* $\{12, 43\}$*, with the total scatter that exceeds* $A = 156$*. However, if* $\mathcal{C}_2$ *is constructed first, then the set* $\{96, 99\}$ *will be obtained,* $\mathcal{C}_1$ *will be equal to* $\{12, 43\}$*, and this solution is feasible (the maximum of the scatters of* $\mathcal{C}_i$ *is 91). Thus, there are instances of problem with fixed centers for which algorithm* $\mathcal{A}(\ldots, A, M)$ *produces a* $\frac{1}{2}$*-approximate solution only at Step 3.*

5. ALGORITHM FOR PROBLEM WITH CENTERS FROM THE INPUT SET

5.1. Exact algorithm for one-cluster problem

In [1], it is shown that **BSD-CLUSTER** with centers from the input set reduces to problem **CLUSTER**($\mathcal{Y}$, F_2^M, F_2^M, A, $M_{\max}$), where

$$F_2^M(\mathcal{C}) = \min_{u \in \mathcal{Y}} F(\mathcal{C}, u). \quad (6)$$

Also it can be shown that if we know $\frac{1}{2}$-approximate solution to problem **CLUSTER**($\mathcal{Y}$, F_2^M, F_2^M, A, $M_{\max}$), then we can calculate the centers

$$u_i = \arg\min_{u \in \mathcal{Y}} F(\mathcal{C}_i, u),\ i = 1, 2, \quad (7)$$

in quadratic time and obtain a $\frac{1}{2}$-approximate solution to original problem with centers from the input set.

As in the case of problem with fixed centers, if we construct an algorithm that finds an exact solution to one-cluster problem generated by problem with centers from the input set, then it can be used to construct an algorithm that finds a $\frac{1}{2}$-approximate solution to respective maximin **CLUSTER** problem.

As in Section 4.1, we propose the approach which allows one to simplify solving one-cluster problem generated by problem with centers from the input set and avoid unnecessary sortings at the same time. First of all, we need to calculate all the distances

$$\mathcal{D}_i[j] = \|\mathcal{Y}[j] - \mathcal{Y}[i]\|,\ 1 \le i, j \le N, \tag{8}$$

sort the input set N times using these distances and store the results as sequences $\mathcal{Z}_1, \ldots, \mathcal{Z}_N$, where

$$\mathcal{D}_i[\mathcal{Z}_i[m]] \le \mathcal{D}_i[\mathcal{Z}_i[n]],\ 1 \le m < n \le N,\ 1 \le i \le N. \tag{9}$$

As in the case with the previous problem, using the proposed structures, it is possible in $\mathcal{O}(N^2)$ time to find a solution to the one-cluster problem with arbitrary subset $\tilde{\mathcal{Y}} \subseteq \mathcal{Y}$ as input set. It is enough to iterate over all possible values of the center, for each center select M of the allowed elements closest to it, and among all such solutions choose the solution with the minimal scatter. We will denote the corresponding algorithm as $\mathcal{A}_{*,*}(\mathcal{Z}_1, \ldots, \mathcal{Z}_N, \mathcal{D}_1, \ldots, \mathcal{D}_N, \mathcal{M}, A_{\min}, M)$. By analogy with the problem with fixed centers, the output of the algorithm is a sequence $\mathcal{B}$ of zeros and ones such that $\mathcal{C}^* = \{\mathcal{Y}[j] \mid \mathcal{B}[j] = 1,\ 1 \le j \le N\}$, and the index j of the chosen center, such that $\mathcal{C}^*$ is the optimal solution to the problem **CLUSTER**$(\tilde{\mathcal{Y}}, F_2^M, A_{\min}, M)$ and $F_2^M(\mathcal{C}^*) = \sum\limits_{y \in \mathcal{C}^*} \|y - \mathcal{Y}[j]\|$.

5.2. Approximation algorithm for problem with centers from the input set

The following algorithm finds a solution to **CLUSTER** problem corresponding to problem with centers from the input set.

```
Algorithm $\mathcal{A}_{*,*}(\mathcal{Z}_1, \ldots, \mathcal{Z}_N, \mathcal{D}_1, \ldots, \mathcal{D}_N, A, M)$
Input: sequences $\mathcal{Z}_1, \ldots, \mathcal{Z}_N$ of numbers from 1 to $N$ and sequences $\mathcal{D}_1, \ldots, \mathcal{D}_N$
    of real numbers such that (9) holds; $A \in \mathbb{R}^+$; positive integer $M \le N$.
1: If $2M > N$, then terminate the algorithm (no solution is constructed).
2: In $\mathcal{M}_1$ and $\hat{j}_1$ write the result of $\mathcal{A}_{*,*}(\mathcal{Z}_1, \ldots, \mathcal{Z}_N, \mathcal{D}_1, \ldots, \mathcal{D}_N, \mathcal{M}_0, A_{\min}, M)$,
    where $\mathcal{M}_0$ is a zero-filled sequence of length $N$.
3: In $\mathcal{M}_2$ and $\hat{j}_2$ write the result of $\mathcal{A}_{*,*}(\mathcal{Z}_1, \ldots, \mathcal{Z}_N, \mathcal{D}_1, \ldots, \mathcal{D}_N, \mathcal{M}_1, A_{\min}, M)$.
    If $\sum_{k=1}^{N} \mathcal{M}_1[k] * \mathcal{D}_{\hat{j}_1}[k] > A$ or $\sum_{k=1}^{N} \mathcal{M}_2[k] * \mathcal{D}_{\hat{j}_2}[k] > A$, terminate the algorithm
    (no solution is constructed). Otherwise, go to Step 3.
4: Form the sets $\mathcal{C}_1$ and $\mathcal{C}_2$ according to the formulas

        $\mathcal{C}_i = \{\mathcal{Y}[j] \mid \mathcal{M}_i[j] = 1,\ 1 \le j \le N\},\ i = 1, 2.$

    Return clusters $\mathcal{C}_1, \mathcal{C}_2$ and their corresponding centers $\mathcal{Y}[\hat{j}_1], \mathcal{Y}[\hat{j}_2]$ as the result
    of the algorithm.
```

Proposition 13. *For **BSD-CLUSTER**($\mathcal{Y}$, $*$, $*$, A), there is a $\frac{1}{2}$-approximation algorithm with time complexity $\mathcal{O}(N^2(d + \log N))$.*

Proof. Consider an arbitrary instance of this problem: $\mathcal{Y} \subset \mathbb{R}^d$ and $A \in \mathbb{R}_+$. Based on it, we construct an instance of problem **CLUSTER**($\mathcal{Y}$, F_2^M, F_2^M, A, $M_{\max}$). Let $\mathcal{C}_1$, $\mathcal{C}_2$ be the result of algorithm $\mathcal{A}(\dots, A, M_{\max})$ applied to this instance, where algorithm $\mathcal{A}_{*,*}(\mathcal{Z}_1, \dots, \mathcal{Z}_N, \mathcal{D}_1, \dots, \mathcal{D}_N, A, M)$ is used to solve **CLUSTER** problem. As in Proposition 11, since the $\mathcal{A}_{*,*}(\dots, A, M)$ algorithm is a formal description of the $\mathcal{A}(\dots, A, M)$ algorithm in the case when the scatter function corresponds to **BSD-CLUSTER**($\mathcal{Y}$, $*$, $*$, A), then the obtained solution is $\frac{1}{2}$-approximate for the considered instance. Therefore $\mathcal{C}_1$, $\mathcal{C}_2$, u_1, u_2 are the $\frac{1}{2}$-approximate solution to original considered problem, where u_i, $i = 1, 2$, are determined by formula (7) and can be calculated in quadratic time $\mathcal{O}(|\mathcal{Y}|^2)$. Thus, $\mathcal{C}_i$, $i = 1, 2$, can be constructed in $\mathcal{O}(N^2(d + \log N))$ time, and u_i, $i = 1, 2$, — in time $\mathcal{O}(N^2)$. So, the total complexity is $\mathcal{O}(N^2(d + \log N))$. □

6. ALGORITHM FOR PROBLEM WITH GEOMETRIC CENTERS

6.1. Exact algorithm for one-cluster problem

Let us consider the one-dimensional case of **BSD-CLUSTER** with geometric centers. As for two previous types of centers, we justify the approach for the one-dimensional case of maximin **CLUSTER** problem generated by problem with geometric centers, which allows us to find a $\frac{1}{2}$-approximate solution.

The following property of the scatter function $F_3(\mathcal{C}) = F(\mathcal{C}, \bar{y}(\mathcal{C}))$ is proved in [1].

Proposition 14. *Let $\mathcal{C} = \{y_1, \dots, y_k\} \subset \mathbb{R}$, $y_{\min} = \min\limits_{y \in \mathcal{C}} y$ and $y_{\max} = \max\limits_{y \in \mathcal{C}} y$. Then for every z such that $y_{\min} < z < y_{\max}$, at least one of the following inequalities holds:*

1. $F_3(\mathcal{C} \cup \{z\} \setminus \{y_{\max}\}) < F_3(\mathcal{C})$
2. $F_3(\mathcal{C} \cup \{z\} \setminus \{y_{\min}\}) < F_3(\mathcal{C})$

It follows from Proposition 14 that if in the one-dimensional case of one-cluster problem with geometric centers all elements in $\mathcal{Y}$ are different, then the optimal solution to this problem consists of M consecutive points of the set $\mathcal{Y}$. Indeed, suppose that for the optimal solution $\mathcal{C}^* = \{y_{i_1}, y_{i_2}, \dots, y_{i_k}\}$, where $y_{i_1} < y_{i_2} < \ldots < y_{i_k}$, there exists $z \in \mathcal{Y} \setminus \mathcal{C}^*$ such that $y_{i_1} < z < y_{i_k}$. Then, according to Proposition 14, the scatter of one of the sets $\mathcal{C}^* \cup \{z\} \setminus \{y_{i_k}\}$, $\mathcal{C}^* \cup \{z\} \setminus \{y_{i_1}\}$ is less than the scatter of the set $\mathcal{C}^*$, which contradicts the optimality of $\mathcal{C}^*$.

Similarly, it can be proved that if the set $\mathcal{Y} = \{y_1, \dots, y_N\} \subset \mathcal{R}$, where $y_1 \leq y_2 \leq \ldots \leq y_N$, contains identical elements, then among the optimal solutions of considered one-cluster problem there is a solution that consists of the elements of the set $\mathcal{Y}$ with consecutive indices.

An algorithm that solves a one-cluster problem with centroids is described in [1]. In the case when the input set is sorted, the algorithm has the complexity $\mathcal{O}(N)$.

Therefore, we can preliminarily sort the input set $\mathcal{Y}$ in non-decreasing order and save the resulting indices as a sequence $\mathcal{L}$:

$$\mathcal{Y}[\mathcal{L}[m]] \leq \mathcal{Y}[\mathcal{L}[n]],\ 1 \leq m < n \leq |\mathcal{L}|. \tag{10}$$

Then, among all optimal solutions of considered one-cluster problem, there is a solution that contains consecutive elements from $\mathcal{L}$.

By analogy with the algorithm from [1], if the sequence $\mathcal{L}$ is known, it is possible to find an optimal solution to a one-cluster problem in time $\mathcal{O}(N)$. We will denote the corresponding algorithm as $\mathcal{A}_{\bar{c}_1,\bar{c}_2}(\mathcal{L}, \mathcal{Y}, A_{\min}, M)$. Unlike in the previous algorithms for one-cluster problems, it is convenient to assume that the output of this algorithm is a sequence i_1, …, i_M of consecutive elements from $\mathcal{L}$, such that the set $\mathcal{C}^* = \{\mathcal{Y}[i_1], \ldots, \mathcal{Y}[i_M]\}$ is the optimal solution to the problem **CLUSTER**($\tilde{\mathcal{Y}}$, $\mathcal{F}$, $A_{\min}$, M), where $\tilde{\mathcal{Y}} = \{\mathcal{Y}[\mathcal{L}[m]] \mid 1 \leq m \leq |\mathcal{L}|\}$, $\mathcal{F}(\mathcal{C}) = F_3(\mathcal{C})$.

6.2. Approximation algorithm for problem with geometric centers

The following algorithm finds a solution to the one-dimensional case of **CLUSTER** problem corresponding to **BSD-CLUSTER** with geometric centers.

```
Algorithm  $\mathcal{A}_{\bar{c}_1,\bar{c}_2}(\mathcal{L}, \mathcal{Y}, A, M)$
Input: a sequence $\mathcal{L}$ of numbers from 1 to $N$ and a sequence $\mathcal{Y}$ of real numbers
    such that (10) holds; $A \in \mathbb{R}^+$; positive integer $M \leq N$.
1: If $2M > N$, then terminate the algorithm (no solution is constructed).
2: Let $\mathcal{B}_1$ be the result of the $\mathcal{A}_{\bar{c}_1,\bar{c}_2}(\mathcal{L}, \mathcal{Y}, A_{\min}, M)$ algorithm and the sequence
   $\mathcal{B}_2$ be the result of $\mathcal{A}_{\bar{c}_1,\bar{c}_2}(\mathcal{L} \setminus \mathcal{B}_1, \mathcal{Y}, A_{\min}, M)$. Calculate centroids using the
   formula $\bar{y}_i = \frac{1}{M} \sum_{j=1}^{M} \mathcal{Y}[\mathcal{B}_i[j]]$, $i = 1, 2$.
   If $\sum_{j=1}^{M} |\mathcal{Y}[\mathcal{B}_1[j]] - \bar{y}_1| > A$ or $\sum_{j=1}^{M} |\mathcal{Y}[\mathcal{B}_2[j]] - \bar{y}_2| > A$, terminate the algorithm
   (no solution is constructed). Otherwise, go to Step 3.
3: Form the sets $\mathcal{C}_i$ according to the formulas

       $\mathcal{C}_i = \{\mathcal{Y}[\mathcal{B}_i[j]] \mid j = 1, \ldots, M\},\ i = 1, 2,$

   and return them as a result.
```

Remark 15. *The difference $\mathcal{L} \setminus \mathcal{B}_1$ in Step 2 can be calculated in linear time, since the sequence $\mathcal{B}_1$ consists of consecutive elements of $\mathcal{L}$.*

Proposition 16. *For the one-dimensional case of* ***BSD-CLUSTER****($\mathcal{Y}$, $\bar{c}_1$, $\bar{c}_2$, A), there is a $\frac{1}{2}$-approximate algorithm with time complexity $\mathcal{O}(N \log N)$.*

Proof. It can be proved in the same way as Propositions 11, 13. □

7. NUMERICAL EXPERIMENTS

This section will present the results of numerical experiments. The following procedure was used to generate problem instances. We considered fixed positive integers N, d, where N is the size of the input set $\mathcal{Y}$, d is the dimension of the Euclidean space, as well as some d-dimensional scatter functions $\mathcal{F}_1$, $\mathcal{F}_2$. After that, we randomly generated (based on some d-dimensional probability distribution) points $y_i = (y_i^1, \dots, y_i^d) \in \mathbb{R}^d$, $i = 1, \dots, N$. The maximum possible minimum cluster size with this configuration is $\lfloor \frac{N}{2} \rfloor$. To obtain a constraint A on the scatter of sets, we iterated over all admissible sets with $\lfloor \frac{N}{2} \rfloor$-element clusters $\mathcal{C}_1$, $\mathcal{C}_2$, and as A we choose the minimum of all $\max(\mathcal{F}_1(\mathcal{C}_1), \mathcal{F}_2(\mathcal{C}_2))$. Thus, as a result, we get an instance of maximin **CLUSTER** problem — the set $\mathcal{Y}$, the scatter functions $\mathcal{F}_1$, $\mathcal{F}_2$, and the scatter constraint A, to which we will apply the proposed approximate algorithm.

The distribution with the following density will be used:

$$f_{2,d}(x_1, \dots, x_d) = \frac{1}{2}(f_{(-1,\dots,0),\frac{1}{2}} + f_{(1,0\dots,0),\frac{1}{2}}), \tag{11}$$

where $f_{x,\sigma}$ is the density of the d-dimensional normal distribution with mean x and covariance matrix σI, where I is the identity d-dimensional matrix. For the one-dimensional case ($d = 1$), the density of the proposed distribution has the following form.

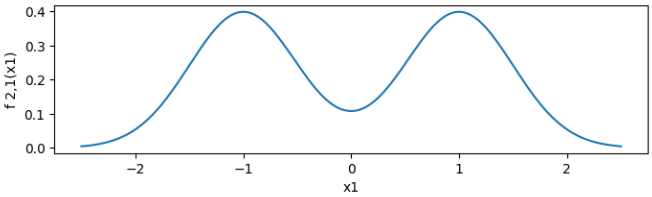

Figure 2: Density plot $f_{2,1}(x_1)$.

The calculation results are shown in Table 1, the values in each row are obtained after solving 60 generated instances (20 instances for each of three considered cases of **problem BSD-CLUSTER**).

Table 1: Results of numerical experiments.

$\lvert\mathcal{Y}\rvert$	d	a_{min} $(+a_{gain})$	a_{avg}	a_{max}
32	1	$\frac{12}{16}$ (+50%)	$\frac{14.496}{16}$	1
32	2	$\frac{12}{16}$ (+50%)	$\frac{14.352}{16}$	1
32	3	$\frac{12}{16}$ (+50%)	$\frac{14.448}{16}$	1
32	4	$\frac{13}{16}$ (+62.5%)	$\frac{14.528}{16}$	1
32	5	$\frac{12}{16}$ (+50%)	$\frac{14.448}{16}$	1
32	6	$\frac{13}{16}$ (+62.5%)	$\frac{14.624}{16}$	1
32	7	$\frac{13}{16}$ (+62.5%)	$\frac{14.608}{16}$	1
32	8	$\frac{12}{16}$ (+50%)	$\frac{14.640}{16}$	1
32	9	$\frac{13}{16}$ (+62.5%)	$\frac{14.784}{16}$	1
32	10	$\frac{13}{16}$ (+62.5%)	$\frac{14.640}{16}$	1

In this table, a_{min}, a_{avg}, and a_{max} are the minimum, average, and maximum approximation ratios, respectively. Since the a_{min} column contains rational numbers, for convenience of comparison, the numbers in the columns a_{min} and a_{avg} are represented as fractions with a denominator of 16. In the a_{min} column the brackets contain the value

$$a_{gain} = \frac{a_{min} - 1/2}{1/2} \cdot 100\%,$$

which is the gain of the minimum approximation ratio over the guaranteed (according to Propositions 11, 13, and 16) estimate 1/2.

Thus, for the proposed distributions, we find that in all test cases even the minimum obtained approximation ratio exceeds the guaranteed approximation ratio of proposed algorithms by at least 50 percent. The obtained average ratios also significantly exceed the guaranteed estimate $\frac{1}{2}$: the difference between 1 and the average approximation ratio is more than twice less than between 1 and the minimum approximation ratio.

8. CONCLUSION

In this paper, we have considered three cases of NP-hard maximin 2-clustering problem. We have constructed 1/2-approximation polynomial algorithms for the first two cases and for the special subcase (when the dimension of the space is equal to one) of the third case. The presented algorithms are N times faster compared to the 1/2-approximation algorithms proposed earlier, where N is equal to the number of elements of the input set. As a result, each accelerated algorithm has the same asymptotic time complexity as the sorting the initial set (according to the scatter functions). Therefore, constructing faster algorithms with the same accuracy, if possible, requires a fundamentally different approach. In addition, it is of interest to build more accurate polynomial algorithms for the problem considered, as well as approximation algorithms for the problems of finding more than two clusters.

Funding. The study was supported by the Russian Academy of Science (the Program of basic research), project FWNF-2022-0015.

REFERENCES

[1] V. Khandeev and S. Neshchadim, "Approximate algorithms for some maximin clustering problems," *Communications in Computer and Information Science*, vol. 1661, pp. 89–103, 2022. doi: 10.1007/978-3-031-16224-4_6

[2] C. Bishop, *Pattern Recognition and Machine Learning.* New York: Springer, 2006.

[3] C. Aggarwal, *Data Mining: The Textbook.* Switzerland: Springer, 2015.

[4] J. Osborne, *Best Practices in Data Cleaning: A Complete Guide to Everything You Need to Do Before and After Collecting Your Data.* London: SAGE, 2013.

[5] A. Ageev, A. Kel'manov, A. Pyatkin, S. Khamidullin, and V. Shenmaier, "Approximation polynomial algorithm for the data editing and data cleaning problem," *Pattern Recognition and Image Analysis*, vol. 27, no. 3, pp. 365–370, 2017. doi: 10.1134/S1054661817030038

[6] A. Eremeev, A. Kel'manov, A. Pyatkin, and I. Ziegler, "On finding maximum cardinality subset of vectors with a constraint on normalized squared length of vectors sum," *Lecture Notes in Computer Science*, vol. 10716, pp. 142–151, 2018. doi: 10.1007/978-3-319-73013-4_13

[7] A. Kel'manov, A. Panasenko, and V. Khandeev, "Exact algorithms of search for a cluster of the largest size in two integer 2-clustering problems," *Numerical Analysis and Applications*, vol. 12, no. 2, pp. 105–115, 2019. doi: 10.1134/S1995423919020010

[8] A. Kel'manov, A. Pyatkin, S. Khamidullin, V. Khandeev, Y. Shamardin, and V. Shenmaier, "An approximation polynomial algorithm for a problem of searching for the longest subsequence in a finite sequence of points in euclidean space," *Communications in Computer and Information Science*, vol. 871, pp. 120–130, 2018. doi: 10.1007/978-3-319-93800-4_10

[9] A. Farcomeni and L. Greco, *Robust methods for data reduction.* CRC press, 2016.

[10] V. Khandeev and S. Neshchadim, "Max-min problems of searching for two disjoint subsets," *Lecture Notes in Computer Science*, vol. 13078, pp. 231–245, 2021. doi: 10.1007/978-3-030-91059-4_17

[11] A. Kel'manov, A. Panasenko, and V. Khandeev, "Exact algorithms of search for a cluster of the largest size in two integer 2-clustering problems," *Numerical Analysis and Applications*, vol. 12, pp. 105–115, 2019. doi: 10.1134/S1995423919020010