Fast and Exact Algorithms for Some NP-Hard 2-Clustering Problems in the One-Dimensional Case

Alexander Kel'manov[1,2](✉) and Vladimir Khandeev[1,2](✉)

[1] Sobolev Institute of Mathematics, 4 Koptyug Avenue, 630090 Novosibirsk, Russia
[2] Novosibirsk State University, 2 Pirogova Street, 630090 Novosibirsk, Russia
{kelm,khandeev}@math.nsc.ru

Abstract. We consider several well-known optimization 2-clustering (2-partitioning) problems of a finite set of points in Euclidean space. All clustering problems considered are induced by applied problems in Data analysis, Data cleaning, and Data mining. In the general case, all these optimization problems are strongly NP-hard. In the paper, we present a brief overview of the results on the problems computational complexity and on their solvability in the one-dimensional case. We present and propose well-known and new, simple and fast exact algorithms with $\mathcal{O}(N \log N)$ and $\mathcal{O}(N)$ running times for the one-dimensional case of these problems.

Keywords: Euclidean space · 2-clustering · 2-partitioning · NP-hardness · Polynomial-time solvability in the 1D case · Fast exact algorithms

1 Introduction

The subject of this research is some hard-to-solve discrete optimization problems that model some simplest applied problems of cluster analysis and data interpretation. Our goal is to analyze and systematize the issues of constructing efficient algorithms that ensure fast and exact problems-solving in the one-dimensional case.

It is known that in terms of applied problem-solving the computer processing of large-scaling data [1] the existing exact and approximate polynomial-time algorithms having theoretical accuracy guarantee but quadratic and higher running time are often unclaimed in applications [2,3]. In other words, these strongly justified polynomial-time algorithms are not used or are rarely used in practice due to the "large" (quadratic and higher) running-time. On the other hand, many hard-to-solve computer geometric problems arising in the data analysis and interpretation [4,5] are solvable in polynomial time when the space dimension is fixed. At the same time, fast polynomial algorithms for solving problems are efficient tools for finding out the structure (i.e. Data mining) [4] of large

W. M. P. van der Aalst et al. (Eds.) AIST 2019, LNCS 11832, pp. 377–387, 2019.
https://doi.org/10.1007/978-3-030-37334-4_34

data by projecting it into spaces of lower dimension (for example, into three-dimensional space, or a plane, or a number line). These projective mathematical tools are popular among data analytics [6] since these tools allow one to interpret the data by the visual representation. In this connection, the construction of fast algorithms having almost linear or linear running-time to solve special cases (in which the space dimension is fixed) of the problems are important mathematical research directions. This paper belongs to these directions.

The paper has the following structure. Section 2 presents the mathematical formulations of the problems under consideration. Interpretations of problems are presented in the next section for demonstrating their origins and their connection with the problems of data analysis. Section 4 provides a brief overview of the results of the computational complexity of problems. In the next section, we present existing results on the problems polynomial solvability in the case of fixed space dimension. Finally, in Sect. 6, the existing and new algorithms are presented, which find the exact solution of the problems in linear or almost linear time in the one-dimensional case.

2 Problems Formulations

Everywhere below $\mathbb{R}$ denotes the set of real numbers, $\|\cdot\|$ denotes the Euclidean norm, and $\langle\cdot,\cdot\rangle$ denotes the scalar product.

All the problems considered below are the problems of 2-partitioning of input points set. In the problems of searching for one subset, the second cluster is understood as the subset that complements this cluster to the input set. A point in the d-dimensional space is interpreted as the measuring result of a set of d characteristics (features) of an object or as the vector (force), i.e. as the segment directed from the origin to this point in the space.

The problems under consideration have the following formulations.

Problem 1 (*Longest Normalized Vector Sum*). *Given:* an N-element set $\mathcal{Y}$ of points in d-dimensional Euclidean space. *Find:* a nonempty subset $\mathcal{C} \subseteq \mathcal{Y}$ such that

$$F(\mathcal{C}) = \frac{1}{|\mathcal{C}|}\Big\|\sum_{y\in\mathcal{C}} y\Big\|^2 \rightarrow \max .$$

Problem 2 (1-*Mean and Given* 1-*Center Clustering*). *Given:* an N-element set $\mathcal{Y}$ of points in d-dimensional Euclidean space. *Find:* a 2-partition of $\mathcal{Y}$ into clusters $\mathcal{C}$ and $\mathcal{Y}\setminus\mathcal{C}$ such that

$$S(\mathcal{C}) = \sum_{y\in\mathcal{C}} \|y - \overline{y}(\mathcal{C})\|^2 + \sum_{y\in\mathcal{Y}\setminus\mathcal{C}} \|y\|^2 \rightarrow \min, \tag{1}$$

where $\overline{y}(\mathcal{C}) = \frac{1}{|\mathcal{C}|}\sum_{y\in\mathcal{C}} y$ is the centroid of $\mathcal{C}$.

Problem 3 (*Longest M-Vector Sum*). *Given:* an N-element set $\mathcal{Y}$ of points in d-dimensional Euclidean space and some positive integer M. *Find:* a nonempty subset $\mathcal{C} \subseteq \mathcal{Y}$ of size M such that

$$H(\mathcal{C}) = \Big\|\sum_{y\in\mathcal{C}} y\Big\| \to \max\,. \tag{2}$$

Problem 4 (*Constrained* 1-*Mean and Given* 1-*Center Clustering*). *Given:* an N-element set $\mathcal{Y}$ of points in d-dimensional Euclidean space and some positive integer M. *Find:* a 2-partition of $\mathcal{Y}$ into clusters $\mathcal{C}$ and $\mathcal{Y} \setminus \mathcal{C}$ minimizing the value of (1) under constraint $|\mathcal{C}| = M$.

Problem 5 (*Longest Vector Sum*). *Given:* an N-element set $\mathcal{Y}$ of points in d-dimensional Euclidean space. *Find:* a subset $\mathcal{C} \subseteq \mathcal{Y}$ maximizing the value of (2).

Problem 6 (*M-Variance*). *Given:* an N-element set $\mathcal{Y}$ of points in d-dimensional Euclidean space and some positive integer M. *Find:* a subset $\mathcal{C} \subseteq \mathcal{Y}$ of size M such that

$$Q(\mathcal{C}) = \sum_{y\in\mathcal{C}} \|y - \overline{y}(\mathcal{C})\|^2 \to \min\,.$$

Problem 7 (*Maximum Size Subset of Points with Constrained Variance*). *Given:* an N-element set $\mathcal{Y}$ of points in d-dimensional Euclidean space and some real number $\alpha \in (0, 1)$. *Find:* a subset $\mathcal{C} \subset \mathcal{Y}$ of the largest size such that

$$Q(\mathcal{C}) \leq \alpha \sum_{y\in\mathcal{Y}} \|y - \overline{y}(\mathcal{Y})\|^2,$$

where $\overline{y}(\mathcal{Y}) = \frac{1}{|\mathcal{Y}|}\sum_{y\in\mathcal{Y}}$ is the centroid of $\mathcal{Y}$.

Problem 8 (*Smallest M-Enclosing Ball*). *Given:* an N-element set $\mathcal{Y}$ of points in d-dimensional Euclidean space and some positive integer number M. *Find:* a minimum radius ball covering M points.

3 Interpretations and Origins of the Problems

All of the formulated optimization problems have simple interpretations in the geometric, statistical, physical, biomedical, geophysical, industrial, economic, anti-terrorism and social terms. One can find some interpretations in the papers cited below. An interested reader can easily give his own interpretation. In this paper, we limit ourselves to a few simple interpretations.

Problems 1–4 arose in connection with the solution of an applied signal processing problem, namely, the problem of joint detecting a quasiperiodically repeating pulse of unknown shape and evaluating this shape under Gaussian noise with zero mean [7–9]. Apparently, the first note on these problems was made in [7]. In these problems, the cluster center specified at the origin corresponds to the mean equal to zero.

It should be noted that simpler optimization problems, which are induced by the applied problems of detecting and discriminating of pulses with given forms in the noise conditions, are characteristic, in particular, for radar, electronic reconnaissance, hydroacoustics, geophysics, technical and medical diagnostics, and Space monitoring (see, for example, [10–12]).

The Problem 1 objective function can be rewritten as

$$F(\mathcal{C}) = \frac{1}{|\mathcal{C}|}\sum_{y\in\mathcal{C}}\sum_{x\in\mathcal{C}}\langle y, x\rangle = \frac{1}{|\mathcal{C}|}\sum_{y\in\mathcal{C}}\Big\langle y, \sum_{x\in\mathcal{C}} x\Big\rangle = \sum_{y\in\mathcal{C}}\langle y, \overline{y}(\mathcal{C})\rangle.$$

Therefore, maximization Problem 1 can be interpreted as a search for a subset $\mathcal{C}$ of objects (forces) that are most similar to each other in the terms of an average value of the sum of all scalar products. Another interpretation is the search for a subset of forces that are most co-aimed with the vector from the origin to the point $\overline{y}(\mathcal{C})$, i.e. from the origin to an unknown centroid. Maximization Problems 1 and 5 have similar interpretations.

Apparently, in [13], Problem 6 was first formulated. This problem models a simplest data analysis problem, namely finding a subset of M similar objects in the set of N objects. In this problem, the similarity of objects is interpreted as the minimum total quadratic scatter of points in a set with respect to some unknown "average" object (centroid), which may not belong to the input set. Equivalent treatment of similarity is minimum of the sum of squares of all possible pairwise distances between objects since for the objective function of the Problem 6, the following equality holds

$$Q(\mathcal{C}) = \frac{1}{2|\mathcal{C}|}\sum_{x\in\mathcal{C}}\sum_{y\in\mathcal{C}}\|x-y\|^2.$$

Problem 7 models [14] the search for the largest subset of similar objects under the upper bound (restriction) on the similarity criterion of Problem 6, i.e. on the total quadratic scatter of points in the desired cluster. In accordance with this restriction, Problem 7 can be interpreted as clearing data from so-called outliers that violate intracluster homogeneity (see, for example, [15–17]). As a result of solving the problem, all data that have significant quadratic scatter will belong to the complementary cluster. In this problem, the degree of the desired cluster homogeneity is governed by the given number α.

Finally, Problem 8, formulated in [18], as a generalization of the well-known problem of the Chebyshev center, has a simple geometric formulation that does not require any explanation. On the other hand, in applications, this problem arises whenever it is necessary to cover (for example, surround or locate in the territory) a given number of objects in the conditions of limited resources (for example, financial or energy).

4 The Computational Complexity of Problems: Existing Results

In [19] the authors proved the strong NP-hardness of Problem 1 by polynomial reducibility of the known NP-hard problem 3-Satisfiability [20] to Problem 1. This result implies the strong NP-hardness of Problem 3 since the objective functions of these problems are related by equality

$$F(\mathcal{C}) = \frac{1}{|\mathcal{C}|} H^2(\mathcal{C}).$$

Indeed, it follows from this equality that polynomial solvability of Problem 3 would imply polynomial solvability of Problem 1 (it would be sufficient to iterate over the finite number of admissible M). Note that chronologically, NP-hardness of Problem 3 was first proved (however, NP-hardness of Problem 3 does not imply NP-hardness of Problem 1). Recall that for proving the intractability of Problem 3, the authors of [8,9,21] constructed polynomial-time reduction of the known NP-hard Clique problem [20] to Problem 3.

By virtue of equality

$$S(\mathcal{C}) = \sum_{y \in \mathcal{Y}} \|y\|^2 - F(\mathcal{C}), \tag{3}$$

the strong NP-hardness of Problem 2 follows from its polynomial equivalence to Problem 1. From the strong NP-hardness of Problem 2 follows the strong NP-hardness of Problem 4 for the same reason as the strong NP-hardness of Problem 3 follows from the strong NP-hardness of Problem 1. Indeed, polynomial solvability of Problem 4 would imply polynomial solvability of Problem 2.

In [22] the authors proved the strong NP-hardness of Problem 5 by polynomial reducibility of the known NP-hard problem 3-Satisfiability [20] to Problem 5.

Further, the paper [23] presents the proof of the strong NP-hardness of Problem 6. In this paper there is a simple proof of polynomial reducibility to Problem 6 of the well-known [24] NP-hard Clique problem on a homogeneous graph with non-fixed degree.

The authors of [14] proved the strong NP-hardness of Problem 7 by showing that decision forms of Problems 6 and 7 are equivalent.

Finally, the paper [18] presents the proof of the strong NP-hardness of Problem 8. To do this, the authors of the cited paper have shown polynomial reducibility to Problem 8 of Clique problem.

5 Exact Algorithms for the Problems in the Multidimensional Case: Existing Results

Exact algorithms of exponential time complexity were constructed for multidimensional case of Problems 1–6 in a number of papers. These algorithms are polynomial for the case of fixed space dimension (or for the case of space dimension bounded from above by some constant).

For Problem 1, an algorithm given in [25] finds the exact solution of the problem in $\mathcal{O}(d^2N^{2d})$ time. The authors of [29] proposed an accelerated algorithm with $\mathcal{O}(dN^{d+1})$ running time.

Since Problem 2 is polynomially equivalent to Problem 1, the exact solution of Problem 2 can be found in the same time as the exact solution of Problem 1.

Polynomial solvability of Problem 3 in the case of fixed space dimension follows from [28]. The authors of [25] and [29] presented exact algorithms for Problem 3 with running time $\mathcal{O}(d^2N^{2d})$ and $\mathcal{O}(dN^{d+1})$, respectively.

Problem 4 is polynomially equivalent to Problem 3. Therefore, the solution of Problem 4 can be found in the same time as the solution of Problem 3.

Further, polynomial solvability of Problem 5 in the case of fixed space dimension follows from [27]. The authors of [26] presented an algorithm with $\mathcal{O}(d^2N^d)$ running time. An improved algorithm with $\mathcal{O}(dN^{d+1})$ running time is proposed in [29]. In addition, the author of [30] presented a faster algorithm with $\mathcal{O}(N^{d-1}(d+\log N))$ running time for the case $d \geq 2$.

Algorithms proposed in [13,29] find exact solution of Problem 6 in $\mathcal{O}(dN^{d+1})$ time. The feature of the algorithm proposed in [29] is that it allows one to find solutions for all admissible values of M at once.

An exact algorithm for Problem 7, obviously, can be obtained from the exact algorithm proposed in [29] for Problem 6. Running time of this algorithm is $\mathcal{O}(dN^{d+1})$.

Finally, the issue of polynomial time solvability of Problem 8 for an arbitrary but fixed dimension d of space is open till now.

It follows from the above results that for $d = 1$ these algorithms find solutions in time which quadratically depends on the power N of the input set.

Below we present simple and fast exact algorithms that find solutions of one dimensional case of the problems in $\mathcal{O}(N)$ (for Problems 3, 4, 5) or $\mathcal{O}(N \log N)$ (for Problems 2, 7, 8) time. Here, for completeness, we present known algorithms for Problems 1 and 6, the time complexity of which is $\mathcal{O}(N \log N)$.

6 Fast and Exact Algorithms for the Problems in the One-Dimensional Case

Hereafter one-dimensional ($d = 1$) cases of Problems 1–8 we will denote as Problem $X - 1D$, where X is the number of the problem.

Let us formulate algorithms for solving the problems.

Algorithm $\mathcal{A}_1$ for Problem 1-1D.

Input: the set $\mathcal{Y}$.

`Step` 1. Split $\mathcal{Y}$ into the two subsets $\mathcal{Y}^+ = \{y \in \mathcal{Y} \mid y > 0\}$ and $\mathcal{Y}^- = \{y \in \mathcal{Y} \mid y < 0\}$. Sort their elements so that $\mathcal{Y}^+ = \{y_1^+ \geq y_2^+ \geq \ldots \geq y_{|\mathcal{Y}^+|}^+ > 0\}$ and $\mathcal{Y}^- = \{y_1^- \leq y_2^- \leq \ldots \leq y_{|\mathcal{Y}^-|}^- < 0\}$.

`Step` 2. Calculate $F_i^+ = F(\{y_1^+, y_2^+, \dots, y_i^+\})$, $i = 1, \dots, |\mathcal{Y}^+|$; put $F^+ = \max\{F_1^+, \dots, F_{|\mathcal{Y}^+|}^+\}$ and $U^+ = \{\{y_1^+, y_2^+, \dots, y_i^+\} \mid F_i^+ = F^+\}$.

`Step` 3. Calculate $F_i^- = F(\{y_1^-, y_2^-, \dots, y_i^-\})$, $i = 1, \dots, |\mathcal{Y}^-|$; put $F^- = \max\{F_1^-, \dots, F_{|\mathcal{Y}^-|}^-\}$ and $U^- = \{\{y_1^-, y_2^-, \dots, y_i^-\} \mid F_i^- = F^-\}$.

`Step` 4. Put $F_A = \max\{F^+, F^-\}$, and also $\mathcal{C}_A = U^+$ if $F^+ \geq F^-$, and $\mathcal{C}_A = U^-$ if $F^+ < F^-$.

Output: the subset $\mathcal{C}_A$, the value F_A.

Proposition 1. *Algorithm $\mathcal{A}_1$ finds an optimal solution of Problem 1-1D in $\mathcal{O}(N \log N)$ time.*

This algorithm was proposed in [19] for construction of the approximation scheme which is polynomial in the case of fixed space dimension. The same paper established the accuracy and running time (determined by the sorting time) of the algorithm.

Algorithm $\mathcal{A}_2$ for Problem 2-1D.

Input: the set $\mathcal{Y}$.

`Step` 1. Find the solution $\mathcal{C}_A$ of Problem 1 using Algorithm $\mathcal{A}_1$.

`Step` 2. Calculate $S_A = \sum_{y \in \mathcal{Y}} y^2 - F_A(\mathcal{C}_A)$.

Output: the subset $\mathcal{C}_A$, the value S_A.

Proposition 2. *Algorithm $\mathcal{A}_2$ finds an optimal solution of Problem 2-1D in $\mathcal{O}(N \log N)$ time.*

The validity of the statement follows from the fact that, in accordance with (3), in the one-dimensional case the following holds

$$S(\mathcal{C}) = \sum_{y \in \mathcal{Y}} y^2 - F(\mathcal{C}).$$

Algorithm $\mathcal{A}_3$ for Problem 3-1D.

Input: the set $\mathcal{Y}$, positive integer M.

`Step 1.` Form a subset $\mathcal{C}_1$ of the M largest elements of $\mathcal{Y}$. Calculate $H(\mathcal{C}_1)$.

`Step 2.` Form a subset $\mathcal{C}_2$ of the M smallest elements of $\mathcal{Y}$. Calculate $H(\mathcal{C}_2)$.

`Step 3.` Put $\mathcal{C}_A = \mathcal{C}_1$ and $H_A = H(\mathcal{C}_1)$ if $H(\mathcal{C}_1) \leq H(\mathcal{C}_2)$. Otherwise put $\mathcal{C}_A = \mathcal{C}_2$ and $H_A = H(\mathcal{C}_2)$.

Output: the subset $\mathcal{C}_A$, the value H_A.

Proposition 3. *Algorithm $\mathcal{A}_3$ finds an optimal solution of Problem 3-1D in $\mathcal{O}(N)$ time.*

The accuracy of the algorithm follows from the fact that in the one-dimensional case for the function (2) the following holds

$$H(\mathcal{C}) = \Big|\sum_{y \in \mathcal{C}} y\Big|. \tag{4}$$

The time complexity of selecting M largest (or smallest) elements determines the running time of the algorithm. This selecting can be made in $\mathcal{O}(N)$ operations without sorting (see, for example, [31]).

Algorithm $\mathcal{A}_4$ for Problem 4-1D.
Input: the set $\mathcal{Y}$, positive integer M.
`Step 1.` Find the solution $\mathcal{C}_A$ of Problem 3 using Algorithm $\mathcal{A}_3$.
`Step 2.` Calculate $S_A = \sum_{y \in \mathcal{Y}} y^2 - F_A(\mathcal{C}_A)$.
Output: the subset $\mathcal{C}_A$, the value S_A.

Proposition 4. *Algorithm $\mathcal{A}_4$ finds an optimal solution of Problem 4-1D in $\mathcal{O}(N)$ time.*

The validity of the statement follows from the polynomial equivalence of Problems 3 and 4.

Algorithm $\mathcal{A}_5$ for Problem 5-1D.
Input: the set $\mathcal{Y}$.
`Step 1.` Form the subset $\mathcal{C}_1 = \{y \in \mathcal{Y} \mid y \geq 0\}$. Calculate $H(\mathcal{C}_1)$.
`Step 2.` Form the subset $\mathcal{C}_2 = \{y \in \mathcal{Y} \mid y \leq 0\}$. Calculate $H(\mathcal{C}_2)$.
`Step 3.` Put $\mathcal{C}_A = \mathcal{C}_1$ and $H_A = H(\mathcal{C}_1)$ if $H(\mathcal{C}_1) \leq H(\mathcal{C}_2)$. Otherwise put $\mathcal{C}_A = \mathcal{C}_2$ and $H_A = H(\mathcal{C}_2)$.
Output: the subset $\mathcal{C}_A$, the value H_A.

Proposition 5. *Algorithm $\mathcal{A}_5$ finds an optimal solution of Problem 5-1D in $\mathcal{O}(N)$ time.*

The accuracy of the algorithm follows from (4). The running time of the algorithm follows from the fact that constructing subsets $\mathcal{C}_1$ and $\mathcal{C}_2$ can be done in time $\mathcal{O}(N)$.

Algorithm $\mathcal{A}_6$ for Problem 6-1D.
Input: the subset $\mathcal{Y}$, positive integer M.
`Step 0.` Put $m = 1$; $Q_A = +\infty$; $\mathcal{C}_A = \emptyset$.
`Step 1.` Using sorting form the tuple $\mathcal{Y}_{1,N} = (y_1, \ldots, y_N)$, where $y_1 < \ldots < y_N$.
`Step 2.` Calculate $f_{m,\,m+M-1}$ using formula

$$f_{i,j} = \sum_{k=i}^{j} (y_k - \overline{y}(\mathcal{Y}_{i,j}))^2 \equiv \sum_{k=i}^{j} y_k^2 - \frac{1}{j-i+1} \left(\sum_{k=i}^{j} y_k \right)^2, \tag{5}$$

where

$$\mathcal{Y}_{i,j} = (y_i, \ldots, y_j),$$

and

$$\overline{y}(\mathcal{Y}_{i,j}) = \frac{1}{j-i+1} \sum_{k=i}^{j} y_k$$

is the centroid of $\mathcal{Y}_{i,j}$, at $i = m$ and $j = m + M - 1$.

```
Step 3. If $f_{m,\,m+M-1} \le Q_A$ then put $Q_A = f_{m,\,m+M-1}$, $\mathcal{C}_A = \mathcal{Y}_{m,\,m+M-1}$.
Step 4. If $m < N - M + 1$ then put $m = m + 1$ and go to Step 1; otherwise go to output.
Output: the subset $\mathcal{C}_A$, the value $Q_A$.
```

Proposition 6. *Algorithm $\mathcal{A}_6$ finds an optimal solution of Problem 6-1D in $\mathcal{O}(N \log N)$ time.*

This statement is based on the fact that each value of $\sum_{k=i}^{j} y_k$ and $\sum_{k=i}^{j} y_k^2$ can be found in $\mathcal{O}(1)$ time using sliding window sums. This algorithm was recently justified in [32].

Algorithm $\mathcal{A}_7$ for Problem 7-1D.

```
Input: the set $\mathcal{Y}$, real number $\alpha$.
Step 0. Put $m = 1$, $M = 0$, $M_A = 0$. Calculate $B = \alpha \sum_{y \in \mathcal{Y}} \|y - \overline{y}(\mathcal{Y})\|^2$.
Step 1. Using sorting form the tuple $\mathcal{Y}_{1,N} = (y_1, \ldots, y_N)$, where $y_1 < \ldots < y_N$.
Step 2. Calculate $f_{m,\,m+M}$ using formula (5) at $i = m$ and $j = m + M$. If $f_{m,\,m+M} \le B$ then go to Step 3. Otherwise go to Step 5.
Step 3. Put $M = M + 1$. If $M > M_A$ then put $M_A = M$, $\mathcal{C}_A = \mathcal{Y}_{m,\,m+M-1}$.
Step 4. If $m + M \le N$ then go to Step 2. Otherwise go to output.
Step 5. If $m < N$ then put $m = m + 1$, $M = M - 1$ and go to Step 2. Otherwise go to output.
Output: the subset $\mathcal{C}_A$, the value $M_A$.
```

Proposition 7. *Algorithm $\mathcal{A}_7$ finds an optimal solution of Problem 7-1D in $\mathcal{O}(N \log N)$ time.*

The algorithm accuracy follows from the monotonicity property [14] of the function $Q(\mathcal{C})$. The sorting determines the algorithm running time since the calculations in Step 2 can be performed in $\mathcal{O}(1)$ time using prefix summation and this step is performed no more than $\mathcal{O}(N)$ times.

Algorithm $\mathcal{A}_8$ for Problem 8-1D.

```
Input: the set $\mathcal{Y}$, positive integer $M$.
Step 0. Put $m = 1$; $r_A = 0$; $\mathcal{C}_A = \emptyset$.
Step 1. Using sorting form the tuple $\mathcal{Y}_{1,N} = (y_1, \ldots, y_N)$, where $y_1 < \ldots < y_N$.
Step 2. Calculate $r_{m,\,m+M-1} = (y_{m+M-1} - y_m)/2$.
Step 3. If $r_{m,\,m+M-1} > r_A$ then put $r_A = r_{m,\,m+M-1}$, $\mathcal{C}_A = \mathcal{Y}_{m,\,m+M-1}$.
Step 4. If $m < N - M + 1$ then put $m = m + 1$ and go to Step 1; Otherwise go to output.
Output: the subset $\mathcal{C}_A$, the value $r_A$.
```

Proposition 8. *Algorithm $\mathcal{A}_8$ finds an optimal solution of Problem 8-1D in $\mathcal{O}(N \log N)$ time.*

The algorithm accuracy follows from the fact that in the one-dimensional case a minimum radius ball enclosing points $y_i, y_{i+1}, \ldots, y_j$, where $y_i < y_{i+1} < \ldots < y_j$, has a radius of $(y_j - y_i)/2$. The algorithm running time is determined by sorting.

7 Conclusion

The paper provides a brief overview of the complexity of some recently identified optimization problems of 2-clustering a finite set of points in Euclidean space. We present fast and exact algorithms for the one-dimensional case of these problems. In our opinion, these algorithms will serve as a good tool for solving the problems of projective analysis and interpretation of big data.

Acknowledgments. This work was supported by the Russian Foundation for Basic Research, project nos. 18-31-00398, 19-01-00308, and 19-07-00397, by the Russian Academy of Science (the Program of basic research), project 0314-2019-0015, and by the Russian Ministry of Science and Education under the 5-100 Excellence Programme.

References

1. Chen, M., Mao, S., Zhang, Y., Leung, V.C.: Big Data Related Technologies, Challenges and Future Prospects. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-06245-7
2. Inmon, W.H.: Oracle: Building High Perfomance Online Systems. QED Information Sciences, Wellesley (1989)
3. Inmon, W.H.: Building the Data Warehouse, 4th edn. Wiley, Hoboken (2005)
4. Aggarwal, C.C.: Data Mining: The Textbook. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-14142-8
5. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)
6. Inmon, W.H., Nesavich, A.: Tapping into Unstructured Data: Integrating Unstructured Data and Textual Analytics into Business Intelligence. Prentice-Hall, Upper Saddle River (2008)
7. Kel'manov, A.V., Khamidullin, S.A., Kel'manova, M.A.: Joint finding and evaluation of a repeating fragment in noised number sequence with given number of quasiperiodic repetitions. In: Book of Abstract of the Russian Conference "Discrete Analysis and Operations Research" (DAOR 2004), p. 185. Sobolev Institute of Mathematics SB RAN, Novosibirsk (2004). (in Russian)
8. Gimadi, E.Kh., Kel'manov, A.V., Kel'manova, M.A., Khamidullin, S.A.: A posteriori detection of a quasi periodic fragment in numerical sequences with given number of recurrences. Sib. J. Ind. Math. **9**(1(25)), 55–74 (2006) (in Russian)
9. Gimadi, E.Kh., Kel'manov, A.V., Kel'manova, M.A., Khamidullin, S.A.: A posteriori detecting a quasiperiodic fragment in a numerical sequence. Pattern Recogn. Image Anal. **18**(1), 30–42 (2008)
10. Kel'manov, A.V., Khamidullin, S.A.: Posterior detection of a given number of identical subsequences in a quasi-periodic sequence. Comput. Math. Math. Phys. **41**(5), 762–774 (2001)
11. Kel'manov, A.V., Jeon, B.: A posteriori joint detection and discrimination of pulses in a quasiperiodic pulse train. IEEE Trans. Sig. Proc. **52**(3), 645–656 (2004)
12. Carter, J.A., Agol, E., et al.: Kepler-36: a pair of planets with neighboring orbits and dissimilar densities. Science **337**(6094), 556–559 (2012)
13. Aggarwal, H., Imai, N., Katoh, N., Suri, S.: Finding k points with minimum diameter and related problems. J. Algorithms. **12**(1), 38–56 (1991)

14. Ageev, A.A., Kel'manov, A.V., Pyatkin, A.V., Khamidullin, S.A., Shenmaier, V.V.: Approximation polynomial algorithm for the data editing and data cleaning problem. Pattern Recogn. Image Anal. **27**(3), 365–370 (2017)
15. Osborne, J.W.: Best Practices in Data Cleaning: A Complete Guide to Everything You Need to Do Before and After Collecting Your Data, 1st edn. SAGE Publication, Inc., Los Angeles (2013)
16. de Waal, T., Pannekoek, J., Scholtus, S.: Handbook of Statistical Data Editing and Imputation. Wiley, Hoboken (2011)
17. Greco, L.: Robust Methods for Data Reduction Alessio Farcomeni. Chapman and Hall/CRC, Boca Raton (2015)
18. Shenmaier, V.V.: Complexity and approximation of the smallest k-enclosing ball problem. J. Appl. Ind. Math. **7**(3), 444–448 (2013)
19. Kel'manov, A.V., Pyatkin, A.V.: On a version of the problem of choosing a vector subset. J. Appl. Ind. Math. **3**(4), 447–455 (2009)
20. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, San Francisco (1979)
21. Baburin, A.E., Gimadi, E.Kh., Glebov, N.I., Pyatkin, A.V.: The problem of finding a subset of vectors with the maximum total weight. J. Appl. Ind. Math. **2**(1), 32–38 (2008)
22. Pyatkin, A.V.: On complexity of a choice problem of the vector subset with the maximum sum length. J. Appl. Ind. Math. **4**(4), 549–552 (2010)
23. Kel'manov, A.V., Pyatkin, A.V.: NP-completeness of some problems of choosing a vector subset. J. Appl. Ind. Math. **5**(3), 352–357 (2011)
24. Papadimitriou, C.H.: Computational Complexity. Addison-Wesley, New-York (1994)
25. Gimadi, E.K., Pyatkin, A.V., Rykov, I.A.: On polynomial solvability of some problems of a vector subset choice in a Euclidean space of fixed dimension. J. Appl. Ind. Math. **4**(1), 48–53 (2010)
26. Baburin, A.E., Pyatkin, A.V.: Polynomial algorithms for solving the vector sum problem. J. Appl. Ind. Math. **1**(3), 268–272 (2007)
27. Hwang, F.K., Onn, S., Rothblum, U.G.: Polynomial time algorithm for shaped partition problems. SIAM J. Optim. **10**(1), 70–81 (1999)
28. Onn, S., Schulman, L.J.: The vector partition problem for convex objective functions. Math. Oper. Res. **26**(3), 583–590 (2001)
29. Shenmaier, V.V.: Solving some vector subset problems by Voronoi diagrams. J. Appl. Ind. Math. **10**(4), 560–566 (2016)
30. Shenmaier, V.V.: An exact algorithm for finding a vector subset with the longest sum. J. Appl. Ind. Math. **11**(4), 584–593 (2017)
31. Wirth, I.: Algorithms + Data Structures = Programs. Prentice Hall, Upper Saddle River (1976)
32. Kel'manov, A.V., Ruzankin, P.S.: Improved exact algorithm for M-variance problem in the one-dimensional case. Pattern Recogn. Image Anal. (2019, accepted)