A Fast and Efficient Method for Solving the Multiple Closed Curve Detection Problem

Una Radojičić[a], Rudolf Scitovski[b] and Kristian Sabo[c]

Department of Mathematics, University of Osijek, Trg Ljudevita Gaja 6, 31 000 Osijek, Croatia

Keywords: Multiple Closed Curve Detection Problem, k-means Algorithm, Globally Optimal Partition, DIRECT.

Abstract: The paper deals with the multiple closed curve detection problem, i.e. the multiple circle and the multiple ellipse detection problem are especially considered. Based on data coming from a number of closed curves not known in advance, it is necessary to recognize these curves. The approach we propose is based on center-based clustering, which means that the problem is reduced to searching for an optimal partition whose cluster-centers are the closed curves we searched for. By using a modified well-known k-means algorithm with carefully chosen initial approximation we shall look for these curves. For detecting the most appropriate number of clusters (closed curves), we have also proposed a new, specialized index that was significantly better than other well-known indexes.

1 INTRODUCTION

Let $\mathcal{A} = \{a^i = (x_i, y_i)^T \in \mathbb{R}^2 : i = 1, \dots, m\}$, $\mathcal{A} \subset \Delta := [\alpha_1, \beta_1] \times [\alpha_2, \beta_2]$ be a set of data points coming from a number of multiple closed curves in the plane not known in advance that should be reconstructed or detected. Suppose that data coming from a curve are homogeneously distributed around that curve such that random errors from normal distribution with expectation 0 are added to uniformly distributed points on the curve in the direction of a normal.

The paper is organized as follows. In Section 3 the multiple closed curve detection problem is defined, a modification of the well-known k-means algorithm for such problems is given and two new algorithms are proposed, in which a new index for detecting the most appropriate number of clusters is applied. In Section 4, we analyze some special cases and give corresponding illustrative and numerical examples. Finally, some conclusions are given in Section 5.

2 PRELIMINARIES

The method for solving the multiple closed curve detection problem proposed in this paper is based on center-based clustering; hence we shall outline the main facts. In addition to using some distance-like function $d : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+$, $\mathbb{R}_+ = [0, +\infty\rangle$ (see e.g. (Kogan, 2007; Sabo et al., 2013)), the set $\mathcal{A}$ can be partitioned into k clusters $\pi_1, \dots, \pi_k$ with centers

$$c_j := \operatorname*{argmin}_{x \in \mathrm{conv}(\mathcal{A})} \sum_{a^i \in \pi_j} d(x, a^i), \tag{1}$$

by solving an appropriate *global optimization problem* (GOP)

$$\operatorname*{argmin}_{c \in \mathrm{conv}(\mathcal{A})^k} F(c), \quad F(c) = \sum_{i=1}^{m} \min_{1 \le j \le k} d(c_j, a^i), \tag{2}$$

where $F : \mathbb{R}^{kn} \to \mathbb{R}_+$ is a Lipschitz continuous symmetric, but nondifferentiable and nonconvex function. Problem (2) is well known under the name center-based clustering problem.

The problem of recognizing some closed curve and the problem of detecting a partition with the most appropriate number of clusters with curve-centers can be observed separately, but these two problems are essentially closely related. The new index, proposed in our paper, is specialized in such situations and shows very good results.

[a] https://orcid.org/0000-0003-0329-0595
[b] https://orcid.org/0000-0002-7386-5991
[c] https://orcid.org/0000-0002-1787-3161

Radojičić, U., Scitovski, R. and Sabo, K.
A Fast and Efficient Method for Solving the Multiple Closed Curve Detection Problem.
DOI: 10.5220/0007238902690276
In *Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2019)*, pages 269-276
ISBN: 978-989-758-351-3

3 THE MULTIPLE CLOSED CURVE DETECTION PROBLEM

In our paper, we consider the multiple closed curve detection problem as a special center-based clustering problem, where cluster-centers are closed curves $C_j(p_j)$, $j = 1,\dots,k$, where $p_j \in \mathbb{R}^n$ is a vector of parameters. This means that the set $\mathcal{A}$ will be partitioned into $k \geq 1$ nonempty mutually disjoint clusters whose centers will be curves C_j. In what follows, they will be called C-cluster-centers.

For example, we can consider the multiple circle detection problem (Akinlar and Topal, 2013; Scitovski and Marošević, 2014), the multiple ellipse detection problem (Akinlar and Topal, 2013; Grbić et al., 2016; Marošević and Scitovski, 2015; Moshtaghi et al., 2011), the multiple generalized circle detection problem (Thomas, 2011), etc. For each of these types of curves, specialized methods have been developed, which can be seen in the aforementioned references.

Such problems occur in a number of applications like pattern recognition, computer vision and robotics (Akinlar and Topal, 2013; Prasad et al., 2013), anomaly detection in wireless sensor networks (Moshtaghi et al., 2011), medical diagnosis (Grbić et al., 2016), agriculture etc.

According to (2), the problem of searching for an optimal k-partition will be defined as the following `GOP`:

$$\operatorname*{argmin}_{\mathbf{p}\in\mathbb{R}^{kn}} F(\mathbf{p}), \quad F(\mathbf{p}) = \sum_{i=1}^{m} \min_{1\leq j\leq k} \mathfrak{D}(C_j(p_j), a^i), \quad (3)$$

where $\mathbf{p} = (p_1,\dots,p_k)$ and $\mathfrak{D}(C_j(p_j), a^i)$ is the distance from the point $a^i \in \mathcal{A}$ to the curve C_j. This distance can be determined numerically (see e.g. (Uteshev and Goncharova, 2018)), but for the most commonly used curves there are explicit formulas.

Since with an increase in the number of clusters k in the partition the value of the corresponding objective function F_k decreases monotonically (Kogan, 2007; Scitovski and Scitovski, 2013) and since the number of clusters (the number of curves) k is not known in advance, we shall search for optimal k-partitions for $k = 1,2,\dots$ until the following condition is met for some small $\varepsilon_B > 0$ (say .005) (Bagirov, 2008):

$$\tfrac{F_{k-1}-F_k}{F_1} < \varepsilon_B. \quad (4)$$

Among the obtained optimal partition we should choose the one with the most appropriate number of clusters. Some modification of standard indexes such as Davies-Bouldin, Calinski-Harabasz, etc. (see e.g. (Morales-Esteban et al., 2014)), can be used for this purpose. By using a special structure of the problem in Subsection 3.4 we define a new index that is significantly better. This index will enable us to stop the procedure of searching for new optimal partitions before meeting criterion (4).

For solving `GOP` (3) we can use a derivative-free, deterministic sampling method for global optimization of a Lipschitz continuous function $g\colon \mathcal{D} \to \mathbb{R}$ defined on a bound-constrained region $\mathcal{D} \subset \mathbb{R}^p$ named Dividing Rectangles (`DIRECT`) was proposed by (Jones et al., 1993). The function g is first transformed into $f\colon [0,1]^p \to \mathbb{R}$, and after that, by means of a standard strategy (see, e.g. (Grbić et al., 2013; Gablonsky, 2001; Jones et al., 1993; Jones, 2001; Paulavičius and Žilinskas, 2014)), the unit hypercube $[0,1]^p$ is divided into smaller hyperrectangles, among which the so-called potentially optimal ones are first searched for and then further divided. It should be noted that this procedure does not assume knowing the Lipschitz constant $L > 0$. However, although the objective function F from (3) is Lipschitz-continuous, solving `GOP` (3) directly by applying the algorithm `DIRECT` is not acceptable since F is a symmetric function with $n \times k$ variables in k parameters $p_1,\dots,p_k \in \mathbb{R}^n$, which, due to these reasons, has at least $k!$ different points in which it reaches the global minimum. The `DIRECT` algorithm would search for all of these points and because of that its efficiency would be very low.

Therefore for solving `GOP` (3) we propose the method which can be shortly described in two steps. First, find a good initial approximation by applying the `DIRECT` algorithm and after that find an optimal solution by applying the k-means algorithm modified for the case of C-cluster-centers.

3.1 Searching for an Initial Approximation for GOP (3)

An initial approximation for `GOP` (3) will be searched for such that we consider the problem of searching for an optimal partition whose cluster-centers are some simple geometrical objects resembling our closed curves and possibly with a small number of parameters. By applying a smaller number of iterations of the `DIRECT` algorithm we can obtain an acceptable initial approximation for `GOP` (3). In this way, we will try to at least position well the closed curves we search for.

3.2 Modification of the k-means Algorithm for $\mathcal{C}$-cluster-centers

In order to find an optimal k-partition, we shall use the well-known k-means algorithm modified for the case of $\mathcal{C}$-cluster-centers. The algorithm can be described in two steps repeated successively.

Algorithm 1: Modification of the k-means algorithm for $\mathcal{C}$-cluster-centers (`KMCC`).

Modification of the k-means algorithm for $\mathcal{C}$-cluster-centers (`KMCC`).

A: For each set of mutually different curves $\mathcal{C}_1(p_1), \ldots, \mathcal{C}_k(p_k)$, the data set $\mathcal{A}$ should be partitioned into k disjoint nonempty clusters by using the minimum distance principle

$$\pi_j = \{\mathfrak{D}(\mathcal{C}_j(t_j), a) \leq \mathfrak{D}(\mathcal{C}_s(t_s), a), \forall s \neq j\} \quad (5)$$

B: For the given partition $\Pi = \{\pi_1, \ldots, \pi_k\}$ of the set $\mathcal{A}$, we should determine the corresponding $\mathcal{C}$-cluster-centers $\hat{\mathcal{C}}_j(\hat{p}_j)$, $j = 1, \ldots, k$ by solving the following `GOP`s :

$$\hat{\mathcal{C}}_j(\hat{p}_j) \in \underset{p \in \mathbb{R}^n}{\operatorname{argmin}} \sum_{a \in \pi_j} \mathfrak{D}(\mathcal{C}_j(p), a). \quad (6)$$

`GOP` (6) could be solved by applying the `DIRECT` algorithm, but if we are able to find a good initial approximation for the curve C_j as a representant of the cluster π_j, then we can apply some local optimization method (i.e. Nelder-Mead or some Quasi-Newton method).

3.3 A New Algorithm for Searching for an Optimal k-partition

As already mentioned in Subsection 3.1, a new algorithm for solving `GOP` (3) refers to finding a good initial approximation, which will be used in the `KMCC` algorithm in order to search for an optimal solution.

Algorithm 2: Searching for an optimal k-partition.

Input: $\mathcal{A} \subset \Delta \subset \mathbb{R}^2$ {Set of data points}; $k \geq 1$;

1: Choose a family of some simple curves $\tilde{\mathcal{C}}_j$ and find an approximation of `GOP` (3) for them by applying a few iterations of the `DIRECT` algorithm;

2: Apply the `KMCC` algorithm with the initial approximation from Step 1;

Output: $\{k, \Pi_k^\star, \{\mathcal{C}_1^\star, \ldots, \mathcal{C}_k^\star\}, F_k^\star\}$.

3.4 A New Index for Detecting the Most Appropriate Number of Clusters

Let $\mathcal{A} \subset \mathbb{R}^n$ be a given set of data points coming from a number of closed curves not known in advance. Assume that data coming from a curve are homogeneously distributed in the neighborhood of that curve such that random errors from normal distribution with expectation 0 are added to uniformly distributed points on the curve in the direction of a normal. By uniformly distributed points on the curve we mean points randomly chosen in such a way that the probability of choosing a point on a particular arc is proportional to its length. By using the concept from the DBSCAN algorithm (see e.g. (Ester et al., 1996; Viswanath and Babu, 2009)), we will try to estimate the density of points in the neighborhood of the curve.

For the given $MinPts > 2$ (according to (Ester et al., 1996) we take $MinPts = 3$) and for every $a \in \mathcal{A}$ we define the radius $\varepsilon_a > 0$ of the circle in which there are $MinPts$ elements of the set $\mathcal{A}$. For the set $\mathcal{R} = \{\varepsilon_a \colon a \in \mathcal{A}\}$ of all these radii, we determine the mean and the variance, and we define the parameter ε as:

$$\varepsilon := \mu + 3\sigma,$$
$$\mu = \operatorname{Mean}(\mathcal{R}), \quad \sigma^2 = \tfrac{1}{|\mathcal{R}|-1} \sum_{a \in \mathcal{B}} (\varepsilon_a - \mu)^2. \quad (7)$$

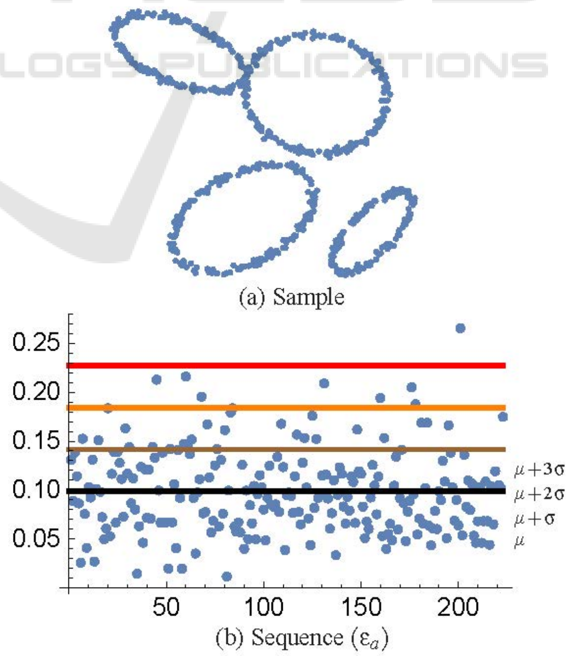


Figure 1: Determination of the parameter ε.

Example 1. *Let us consider the set of data $\mathcal{A}$ coming from 4 ellipses (see Fig. 1a). A sequence (ε_a) and the options for choosing the parameter ε are shown in*

Fig. 1b. We have opted for $\varepsilon = \mu + 3\sigma$ *because in this case, for almost all points* $a \in \mathcal{A}$*, the circle* $C(a,\varepsilon)$ *contains at least MinPts points from* $\mathcal{A}$*.*

Let $\Pi = \{\pi_1,\ldots,\pi_k\}$ be an optimal k-partition of the set $\mathcal{A}$, whose cluster-centers are closed curves $\mathcal{C}_1,\ldots,\mathcal{C}_k$ and let $\varepsilon > 0$ be the parameter value estimated by (7). For each cluster π_j, $j = 1,\ldots,k$ we define a set $V_j = \{\mathfrak{D}_1(\mathcal{C}_j,a)\colon a \in \pi_j\}$ of orthogonal distances from the points of the cluster π_j to the curve $\mathcal{C}_j$. Due to the character of the set $\mathcal{A}$, the elements of the set V_j come from the folded normal distribution with the location parameter 0 (see, e.g. (Tsagris et al., 2014)).

Because of the assumption of homogeneity of data around the curves searched for if the curve $\mathcal{C}_j$ is well detected, the absolute deviation of the majority of points from π_j to the curve $\mathcal{C}_j$ (i.e. *Quantile Absolute Deviation* (QAD)),

$$\mathtt{QAD}\,(j) = \mu_j + 2\sigma_j, \tag{8}$$

$$\mu_j = \mathrm{Mean}\,(V_j), \quad \sigma_j^2 = \tfrac{1}{|V_j|-1} \sum_{v_j \in V_j} (v_j - \mu_j)^2,$$

should be less than ε.

The new index should depict this property for every curve $\mathcal{C}_j$; thus the new *Density Based Clustering index* (DBC) is defined by:

$$\mathtt{DBC}\,(k) = \max_{j=1,\ldots,k} \mathtt{QAD}\,(j). \tag{9}$$

Note that the DBC-index defined in this way describes a k-partition such that it detects the cluster whose representant deviates most (in terms of the 95th percentile) from data associated to that cluster.

3.5 A New Algorithm for Searching for an Optimal Partition With the Most Appropriate Number of Clusters

The concept of the algorithm is as follows. First, according to (7), for the given set $\mathcal{A}$ we define the parameter ε. After that, in accordance with Algorithm 2, we determine optimal k-partitions and the corresponding values of the DBC-index in line with (8)-(9) for $k = 1,\ldots$ until the DBC(k) value becomes less than ε or until criterion (4) is met.

If among these k partitions we do not find a globally optimal one, the DBC-index will not be less than ε, and the algorithm will stop running based upon criterion (4) in the k-th step.

Algorithm 3: Searching for an optimal partition with the most appropriate number of clusters.

Input: $\mathcal{A} \subset \Delta \subset \mathbb{R}^2$ {Set of data points}; $\varepsilon_B > 0$ {According to (Bagirov, 2008)};
Determine parameter $\varepsilon > 0$ according to (7);
Set $k = 1$ and call Algorithm 2; Determine Π_k, $\mathcal{C}_k$, F_1, QAD, and DBC;
if DBC $< \varepsilon$, **then**
  GO TO Output;
end if
while
Set $k = k+1$ and call Algorithm 2; Determine Π_k, $\mathcal{C}_1,\ldots,\mathcal{C}_k$, F_2, QAD, and DBC;
DBC $> \varepsilon$ & $(F_1 - F_2)/F_1 > \varepsilon_B$ **do**
  Set $F_1 = F_2$;
end while
if DBC $< \varepsilon$, **then**
  k curves $\mathcal{C}_1,\ldots,\mathcal{C}_k$ detected;
else
  Investigate especially cases when QAD$(j) < \varepsilon$;
end if
Output: $\{k$, Closed curves detected$\}$.

Remark 1. *If* DBC $> \varepsilon$ *occurs at the end of Algorithm 3, then all components of the vector* $(QAD(1),\ldots,QAD(k))$ *should be checked up. For every* $s \in \{1,\ldots,k\}$*, for which* $QAD(s) < \varepsilon$*, there is hope that the s-th curve has been detected, which can be additionally checked up based upon the density of data by curves* $\rho_s = \frac{|\pi_s|}{|\mathcal{C}_s|}$*,* $s = 1,\ldots,k$*.*

4 SOME SPECIAL CURVES

Some closed curves will be analyzed that most frequently occur in the literature and that have significant applications in different areas.

4.1 Multiple Circle Detection Problem

Let $\mathcal{A} \subset \Delta \subset \mathbb{R}^2$ be a set of data points coming from a number of multiple circles in the plane not known in advance.

When searching for an optimal k-partition (see Subsection 3.3) we will use the algebraic distance $\mathfrak{D}(\mathcal{C}(S,r),a)$ from the point $a \in \mathcal{A}$ to the circle $\mathcal{C}$ with the center S and the radius r, and when defining the set V_j by the DBC-index (see Subsection 3.4) we will use the *Least Absolute Distance* (LAD) $\mathfrak{D}_1(\mathcal{C}(S,r),a)$:

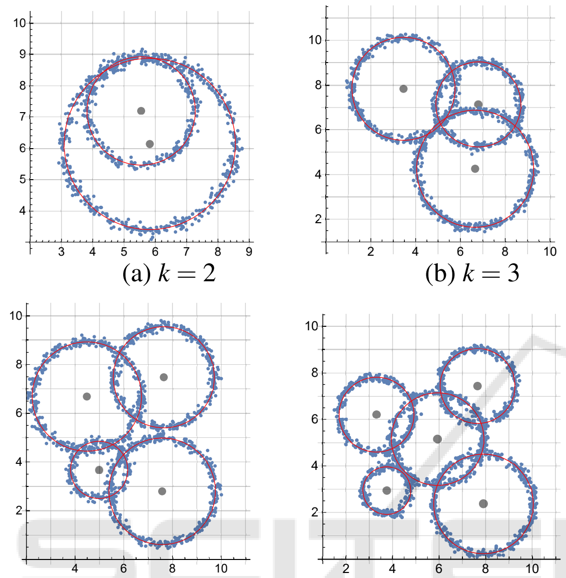
$$\mathfrak{D}_1(\mathcal{C}(S,r),a) = |\|S-a\| - r| \tag{10}$$

$$\mathfrak{D}(\mathcal{C}(S,r),a) = (\|S-a\|^2 - r^2)^2 \tag{11}$$

Now GOP (3) can be written in the following way:

$$\underset{S\in\Delta^k,\, r\in[0,R]^k}{\operatorname{argmin}} \sum_{i=1}^{m} \min_{1\leq j\leq k} \mathfrak{D}(C_j(S_j,r_j),a^i). \tag{12}$$

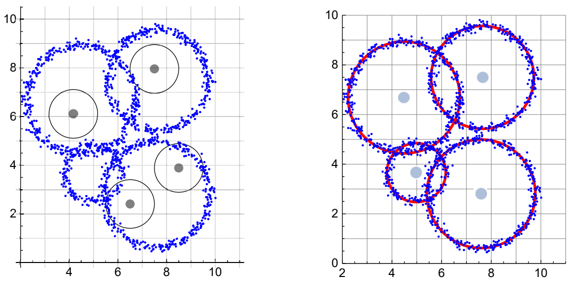
Example 2. *Let $\mathcal{A}$ be the set of data points coming from k circles $C_1,\ldots,C_k$ homogeneously distributed in the neighborhood of the circles. Four selected examples with $k=2,3,4,5$ circles are shown in Fig. 2.*



(a) $k=2$ (b) $k=3$ (c) $k=4$ (d) $k=5$

Figure 2: Four selected examples with $k=2,3,4,5$ circles.

Let us consider a data point set $\mathcal{A}$ shown in Fig. 2c. According to Algorithm 2, by using the DIRECT *algorithm, we first determine an initial approximation (unit circles shown in Fig. 3a), and after that, by using the* KMCC *algorithm, we obtain the solution (Fig. 3b)).*



(a) Initial approximation (b) Solution

Figure 3: Implementation of Algorithm 2 for circles.

Searching for an optimal partition with the most appropriate number of clusters will be illustrated by using the same example. Algorithm 3 terminates when the value of the DBC*-index drops below the estimated value of the parameter $\varepsilon=0.227$ defined by (7).*

We obtained $DBC(1)=1.93$, $DBC(2)=1.07$, $DBC(3)=0.47$, *and* $DBC(4)=0.17$. *This means that the 4-partition is an optimal partition and all circles have been recognized.*

Example 3. *The method for solving the multiple circle detection problem given in Algorithm 3 will be tested on 100 sets of data points generated as in Example 2. Whether some circle is recognized among calculated circles will be established by* QAD*-index or by the Hausdorff distance between these circles. As can be seen in Table 1, Algorithm 3 recognizes very well circles the set of data points $\mathcal{A}$ was generated from and necessary* CPU*-time is reasonably low. All evaluations were done on the basis of our own* Mathematica*-modules, and were performed on the computer with a* `2.90 GHz Intel(R) Core(TM)i7-75000 CPU` *with* `16GB` *of* `RAM`.

Table 1: Result analysis for circles.

No. circ.	Circles detected 2	3	4	5	6	CPU-time DIRECT	k-means	Total
2	19	0	0	0	0	0.147	1.175	1.322
3	0	20	2	0	0	2.339	4.858	7.197
4	0	0	31	0	0	2.563	7.967	10.53
5	1	0	0	26	1	28.60	18.25	46.85

4.2 Multiple Ellipse Detection Problem

Let $\mathcal{A}\subset\Delta\subset\mathbb{R}^2$ be a set of data points coming from a number of multiple ellipses in the plane not known in advance.

An ellipse E can be interpreted as a *Mahalanobis circle* (M-circle) $\mathfrak{M}(S,r;\Sigma)$ with the center at the point $S\in\mathbb{R}^2$, the radius r and the covariance matrix Σ,

$$\mathfrak{M}(S,r;\Sigma)=\{u\in\mathbb{R}^2\colon d_M(S,u;\Sigma)=r^2\}, \tag{13}$$

where $d_M\colon \mathbb{R}^2\times\mathbb{R}^2\to\mathbb{R}_+$ is the *Mahalanobis distance-like function* given by

$$\begin{aligned} d_M(S,u;\Sigma) &= \sqrt{\det\Sigma}(S-u)^T\Sigma^{-1}(S-u) \\ &= \|u-v\|_\Sigma^2. \end{aligned} \tag{14}$$

When searching for an optimal k-partition (see Subsection 3.3) we will use the algebraic distance $\mathfrak{D}(\mathfrak{M}(S,r,\Sigma),a)$ from the point $a\in\mathcal{A}$ to the M-circle $\mathfrak{M}$, and when defining the set V_j by the DBC-index (see Subsection 3.4) we will use the LAD-distance $\mathfrak{D}_1(\mathfrak{M}(S,r,\Sigma),a)$ (see e.g. (Grbić et al., 2016)):

$$\mathfrak{D}_1(\mathfrak{M}(S,r,\Sigma),a)=|\|S-a\|_\Sigma-r| \tag{15}$$

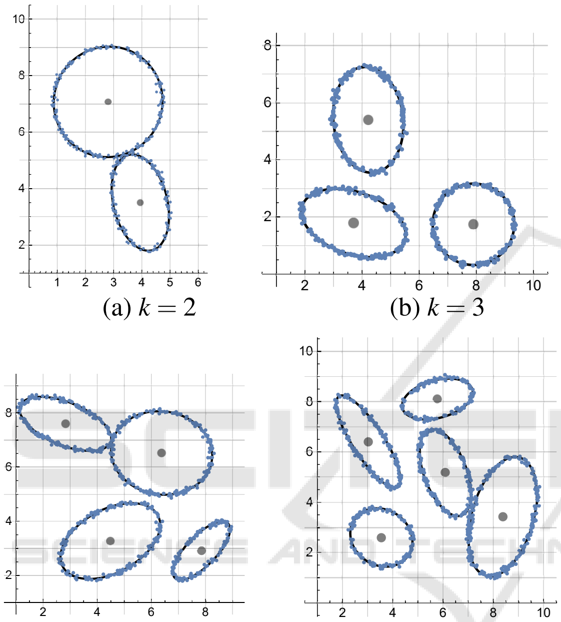
$$\mathfrak{D}(\mathfrak{M}(S,r,\Sigma),a)=(\|S-a\|_\Sigma^2-r^2)^2 \tag{16}$$

Now `GOP` (3) can be written as:

$$\operatorname*{argmin}_{\substack{S\in\Delta^k,\, r\in[0,R]^k \\ \Sigma\in M_2^k}} \sum_{i=1}^{m} \min_{1\le j\le k} \mathfrak{D}(\mathfrak{M}_j(S_j,r_j,\Sigma_j),a^i) \qquad (17)$$

where M_2 is the set of positive definite symmetric matrices of order two.

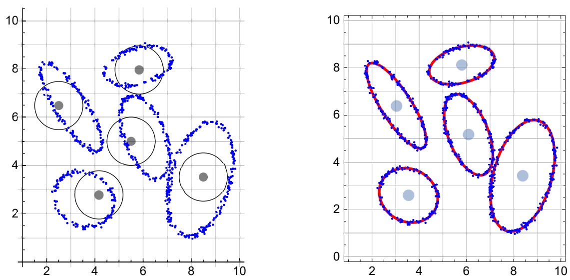
Example 4. *Let $\mathcal{A}$ be the set of data points coming from k ellipses $E_1,\dots,E_k$ homogeneously distributed in the neighborhood of the ellipses. Four selected examples with $k = 2,3,4,5$ circles are shown in Fig. 4.*



(a) $k = 2$ (b) $k = 3$ (c) $k = 4$ (d) $k = 5$

Figure 4: Four selected examples with $k = 2,3,4,5$ ellipses.

Let us consider a data point set $\mathcal{A}$ shown in Fig. 4d. According to Algorithm 2, by using the `DIRECT` *algorithm, we first determine an initial approximation (unit circles shown in Fig. 5a), and after that, by using the* `KMCC` *algorithm, we obtain the solution (Fig. 5b).*



(a) Initial approximation (b) Solution

Figure 5: Implementation of Algorithm 2 for ellipses.

Table 2: Result analysis for ellipses that mostly do not intersect.

No.	Ellipses detected					CPU -time		
ell	1	2	3	4	5	DIRECT	k-means	Total
2	0	20	0	0	0	0.093	1.258	1.351
3	0	0	25	0	0	0.262	3.319	3.580
4	2	0	0	23	2	1.141	8.776	9.917
5	0	1	0	0	27	4.916	19.11	24.03

Table 3: Result analysis for intersecting ellipses.

No.	Ellipses detected					CPU -time		
ell.	1	2	3	4	5	DIRECT	k-means	Total
2	0	20	0	0	0	0.093	2.270	2.363
3	0	1	24	0	0	0.319	4.766	5.086
4	1	3	0	22	1	3.337	15.55	18.89
5	0	2	5	1	20	15.36	31.24	46.60

Searching for an optimal partition with the most appropriate number of clusters will be illustrated by using the same example (see Fig. 4d). The estimated value of parameter (7) is $\varepsilon = .235$, and Algorithm 3 terminates when the value of the `DBC`*-index drops below that value. We obtained* $\mathtt{DBC}(1) = 2.72$, $\mathtt{DBC}(2) = 1.70$, $\mathtt{DBC}(3) = 1.46$, $\mathtt{DBC}(4) = 1.22$, *and* $\mathtt{DBC}(5) = 0.19$. *This means that the 5-partition is an optimal partition and all ellipses have been recognized.*

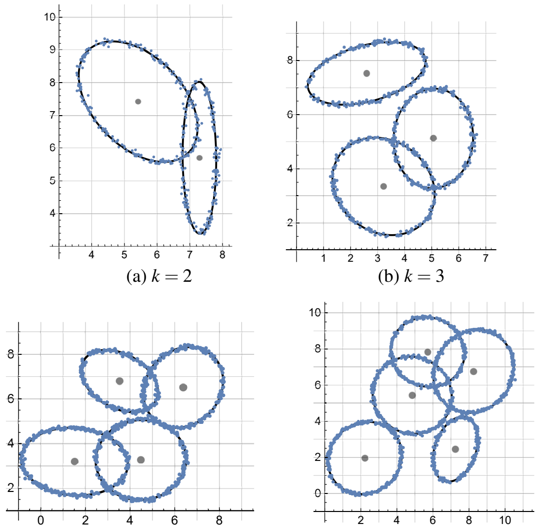
Example 5. *The method for solving the multiple ellipse detection problem given in Algorithm 3 will be tested on 100 sets of data points generated as in Example 4. Whether some ellipse, interpreted as an M-circle $\mathfrak{M}(S,r,\Sigma)$, is recognized among calculated M-circles as an M-circle $\mathfrak{N}(\hat{S},\hat{r},\hat{\Sigma})$ will be established by* `QAD`*-index or by calculating the distance (Grbić et al., 2016):*

$$\mathfrak{D}(\mathfrak{N},\mathfrak{M}) = \sqrt{d_M(\hat{S},S,\hat{\Sigma}+\Sigma)} + |\hat{r}-r|. \qquad (18)$$

As can be seen in Table 2, Algorithm 3 recognizes very well ellipses the set of data points $\mathcal{A}$ was generated from as in Example 4 and necessary `CPU` *-time is very low.*

Example 6. *Similarly to Example 5, Algorithm 3 will be tested on 100 sets of data points generated from $2,3,4$ or 5 intersected ellipses. Four different examples can be seen in Fig. 6*
As can be seen in Table 3, Algorithm 3 still recognizes well ellipses and `CPU` *-time increases a little bit in the case of many ellipses.*

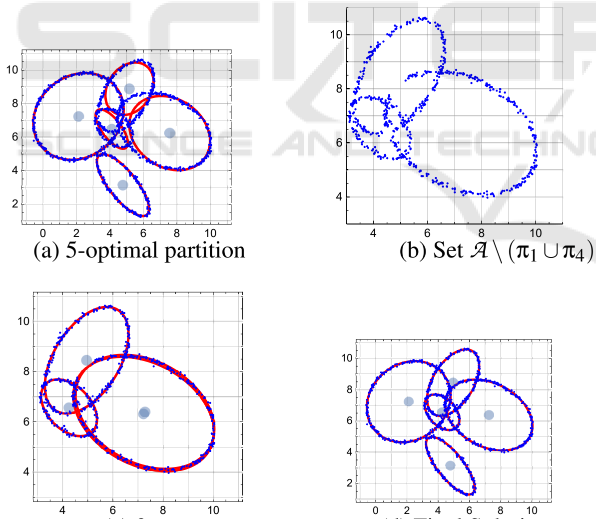
Example 7. *Another advantage of the* `DBC`*-index will be illustrated by an example in which a globally optimal partition has not been found (see Fig. 7a). The estimated value of the parameter ε is $\varepsilon = 0.217$, and the algorithm terminates when condition (4) is met,*



(a) $k = 2$ (b) $k = 3$ (c) $k = 4$ (d) $k = 5$

Figure 6: Four selected examples with $k = 2, 3, 4, 5$ intersected ellipses.

with $\varepsilon_B = 0.005$. The 5-partition that was found is not globally optimal, and characteristics of its clusters are given in Table 4: centers of ellipse-centers, values of the parameter QAD *and densities of points around ellipses according to Remark 1.*



(a) 5-optimal partition (b) Set $\mathcal{A} \setminus (\pi_1 \cup \pi_4)$ (c) 3-opt. part. (d) Final Solution

Figure 7: An example of data for which Algorithm 3 has not found any globally optimal partition.

Table 4: Characteristics of calculated ellipses.

j	S_j	QAD (j)	ρ_j
1	$(2.12, 7.23)$	0.134	20.6
2	$(7.57, 6.24)$	0.198	17.5
3	$(5.17, 8.89)$	0.618	24.9
4	$(4.78, 3.14)$	0.133	19.9
5	$(4.11, 6.48)$	0.471	30.2

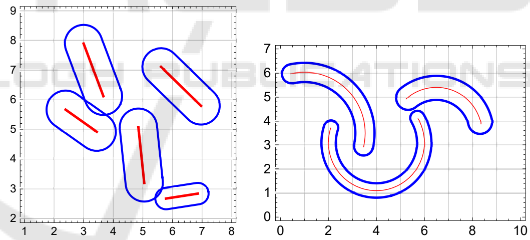
It can be stated that ellipses E_1, E_2, E_4 have been recognized among original ellipses since the corresponding values of the parameter QAD *are less than* 0.217, *where we should carefully take into account ellipse E_2 since the density of points ρ_2 significantly deviates from the average density* $\bar{\rho} = |\mathcal{A}| / \sum_{j=1}^{5} |E_j| = 21.5$.

On the basis of this brief analysis, we can leave out clusters π_1, π_4 from the set $\mathcal{A}$ and reconsider a simple multiple ellipse detection problem for the set $\mathcal{A} \setminus (\pi_1 \cup \pi_4)$ (see Fig. 7b). We obtain the remaining three ellipses (see Fig. 7c). By applying the KMCC *algorithm to the whole set $\mathcal{A}$, the initial approximation obtained in this way yields the Final Solution (see Fig. 7d).*

4.3 Some Other Possibilities

Some other similar problems can be treated in the same way. For example, we can observe a multiple generalized circle detection problem, where a generalized circle implies a set $\mathcal{C}(O(p), r) = \{u \in \mathbb{R}^2 \colon D(O(p), u) = r, r > 0, p \in \mathbb{R}^n\}$, where $O(p)$ is some simple geometrical object.

For example, $O(p)$ can be a line segment $\text{seg}(\mu, \nu)$ ending in points $\mu, \nu \in \Delta$ (see Fig. 8a), or an arc $\text{arc}(C, t_1, t_2)$ of a circle, where C is the center of a circle, and $t_1, t_2 \in [0, 2\pi]$, $t_1 < t_2$ (see Fig. 8b).



(a) (b)

Figure 8: Generalized circles.

After defining the distance from the point $a \in \mathcal{A}$ to the generalized circle $\mathcal{C}$,

$$\mathfrak{D}(\mathcal{C}(O(p), r), a) = (D(O(p), a)^2 - r^2)^2, \qquad (19)$$

where D is the distance from the point $a \in \mathcal{A}$ to geometrical objects $O(p)$, then GOP (3) becomes

$$\underset{p \in \mathbb{R}^{nk}, r \in [0,R]^k}{\operatorname{argmin}} \sum_{i=1}^{m} \min_{1 \le j \le k} \mathfrak{D}(\mathcal{C}_j(O_j(p_j), r_j), a^i). \qquad (20)$$

5 CONCLUSIONS

The proposed method for solving the multiple closed curve detection problem has been shown to be very efficient in special cases of multiple circle and multiple ellipse detection problems.

Let us also mention that in the literature there are some other possibilites such as path-based clustering approach (see (Fischer and Buhmann, 2003)) that can also be used for solving multiple closed curve detection problem.

The proposed DBC -index for detecting the most appropriate number of clusters (closed curves) enables this procedure to be carried out as fast as possible. In more complex cases of intersecting closed curves, i.e. when Algorithm 3 does not give a globally optimal partition, construction of the DBC -index enables the detection of some closed curves and the algorithm to be run on a constrained data set.

The proposed DBC-index should be further investigated and corrected. We will try to apply the proposed method to other curves. In the meantime, we successfully applied the method to multiple generalised circle detection problem with application in Escherichia Coli and Enterobacter-cloaca recognizing.

The proposed algorithm could also be applied to real-world images.

ACKNOWLEDGEMENTS

The author would like to thank Mrs. Katarina Moržan for significantly improving the use of English in the paper. This work was supported by the Croatian Science Foundation through research grants IP-2016-06-6545 and IP-2016-06-8350 .

REFERENCES

Akinlar, C. and Topal, C. (2013). Edcircles: A real-time circle detector with a false detection control. *Pattern Recognition*, 46:725–740.

Bagirov, A. M. (2008). Modified global *k*-means algorithm for minimum sum-of-squares clustering problems. *Pattern Recognition*, 41:3192–3199.

Ester, M., Kriegel, H., and Sander, J. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 226–231, Portland.

Fischer, B. and Buhmann, J. M. (2003). Path-based clustering for grouping of smooth curves and texture segmentation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 25:1–6.

Gablonsky, J. M. (2001). Direct version 2.0. Technical report, Center for Research in Scientific Computation. North Carolina State University.

Grbić, R., Grahovac, D., and Scitovski, R. (2016). A method for solving the multiple ellipses detection problem. *Pattern Recognition*, 60:824–834.

Grbić, R., Nyarko, E. K., and Scitovski, R. (2013). A modification of the DIRECT method for Lipschitz global optimization for a symmetric function. *Journal of Global Optimization*, 57:1193–1212.

Jones, D. R. (2001). The direct global optimization algorithm. In Floudas, C. A. and Pardalos, P. M., editors, *The Encyclopedia of Optimization*, pages 431–440. Kluwer Academic Publishers, Dordrect.

Jones, D. R., Perttunen, C. D., and Stuckman, B. E. (1993). Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications*, 79:157–181.

Kogan, J. (2007). *Introduction to Clustering Large and High-dimensional Data*. Cambridge University Press, New York.

Marošević, T. and Scitovski, R. (2015). Multiple ellipse fitting by center-based clustering. *Croatian Operational Research Review*, 6:43–53.

Morales-Esteban, A., Martínez-Álvarez, F., Scitovski, S., and Scitovski, R. (2014). A fast partitioning algorithm using adaptive Mahalanobis clustering with application to seismic zoning. *Computers & Geosciences*, 73:132–141.

Moshtaghi, M., Havens, T. C., Bezdek, J. C., Park, L., Leckie, C., Rajasegarar, S., Keller, J. M., and Palaniswami, M. (2011). Clustering ellipses for anomaly detection. *Pattern Recognition*, 44:55–69.

Paulavičius, R. and Žilinskas, J. (2014). *Simplicial Global Optimization*. Springer.

Prasad, D. K., Leung, M. K. H., and Quek, C. (2013). Ellifit: An unconstrained, non-iterative, least squares based geometric ellipse fitting method. *Pattern Recognition*, 46:1449–1465.

Sabo, K., Scitovski, R., and Vazler, I. (2013). One-dimensional center-based l_1-clustering method. *Optimization Letters*, 7:5–22.

Scitovski, R. and Marošević, T. (2014). Multiple circle detection based on center-based clustering. *Pattern Recognition Letters*, 52:9–16.

Scitovski, R. and Scitovski, S. (2013). A fast partitioning algorithm and its application to earthquake investigation. *Computers & Geosciences*, 59:124–131.

Thomas, J. C. R. (2011). A new clustering algorithm based on k-means using a line segment as prototype. In Martin, C. S. and Kim, S.-W., editors, *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 638–645. Springer Berlin Heidelberg.

Tsagris, M., Beneki, C., and Hassani, H. (2014). On the folded normal distribution. *Mathematics*, 2:12–28.

Uteshev, A. Y. and Goncharova, M. V. (2018). Point-to-ellipse and point-to-ellipsoid distance equation analysis. *Journal of Computational and Applied Mathematics*, 328:232–251.

Viswanath, P. and Babu, V. S. (2009). Rough-DBSCAN: a fast hybrid density based clustering method for large data sets. *Pattern Recognition Letters*, 30:1477–1488.