ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
UDC/UDK 004.93'11:004.891

IMPROVED BISECTOR CLUSTERING OF UNCERTAIN DATA USING SDSA METHOD ON PARALLEL PROCESSORS

Ivica Lukić, Ninoslav Slavek, Mirko Köhler

Original scientific paper

Clustering uncertain objects is a well researched field. This paper is concerned with clustering uncertain objects with 2D location uncertainty due to object movements. Location of moving object is reported periodically, thus location is uncertain and described with probability density function (PDF). Data about moving objects and their locations are placed in distributed databases. Number of uncertain objects can be very large and obtaining quality result within reasonable time is a challenging task. Basic clustering method is UK-means, in which all expected distances (ED) from objects to clusters are calculated. Thus UK-means is inefficient. To avoid ED calculations various pruning methods are proposed. A survey of existing clustering methods is given in this paper and a combination of two methods is proposed. The first method, called Segmentation of Data Set Area is combined with Improved Bisector pruning to improve execution time of clustering uncertain data. In SDSA method, data set area is divided in many small segments, and only objects in that small segment are observed. Using segments there is a possibility for parallel computing, because segments are mutually independent, thus each segment can be computed on different core of parallel processor. Experiments were conducted to evaluate the effectiveness of the combined methods.

Keywords: *clustering, data mining, expected distance, parallel processing, uncertain data*

Razvrstavanje podataka s nesigurnošću pomoću poboljšane simetralne metode i SDSA metode

Izvorni znanstveni rad

Razvrstavanje podataka s nesigurnošću je vrlo istraživano područje. Ovaj rad posvećen je razvrstavanju objekata koji imaju nesigurnost 2D položaja uzrokovanog gibanjem objekata. Položaj pokretnog objekta izvještava se periodički, i stoga položaj objekta sadrži nesigurnost i opisan je funkcijom gustoće razdiobe (PDF). Podaci o takvim objektima i njihovim položajima čuvaju se u distribuiranim bazama podataka. Broj objekata s nesigurnošću može biti jako velik i dobivanje kvalitetnog rezultata u razumnom vremenu je zahtijevan zadatak. Najjednostavnija metoda za razvrstavanje je UK-means, u kojoj se računaju sve očekivane udaljenosti (ED) od objekata do središta grozdova. Stoga je UK-means nedjelotvorna metoda. Kako bi se izbjeglo računanje očekivanih udaljenosti predstavljene su brojne metode za odbacivanje. U radu je dan pregled postojećih metoda i predložena kombinacija dviju metoda. Prva metoda je nazvana podjela područja skupa podataka (SDSA) i kombinirana je s poboljšanom simetralnom metodom kako bi se skratilo vrijeme razvrstavanja podataka s nesigurnošću. Pomoću SDSA metode područje skupa podataka je podijeljeno na mala pravokutna područja i promatraju se samo objekti koji se nalaze u tom području. Koristeći mala pravokutna područja nudi se mogućnost za paralelno procesiranje, jer su područja međusobno neovisna i mogu se računati na različitim jezgrama procesora. Provedeni su pokusi kako bi se pokazala uspješnost nove kombinirane metode.

Ključne riječi: *očekivana udaljenost, podaci s nesigurnošću, paralelno procesiranje, razvrstavanje, rudarenje podataka*

1 Introduction

Data uncertainty is caused by many factors, such as measurement error, sampling discrepancy, outdated data source etc. Uncertainty is determined by location measurement error, speed of the moving objects, last reported direction and elapsed time. The proper data mining is the goal of all applications which deal with uncertain data. One application of data mining is clustering [18]. Basic clustering method for certain data is K-means. In K-means objects are divided into clusters so the total expected distance from objects to assigned cluster is minimized. For clustering uncertain data K-means is modified and called UK-means. One application of UK-means [2] is clustering data set of moving objects. Clustered objects are near to cluster centre and similar objects are in the same group. The result is better bandwidth utilization, local communication, better cluster locations and energy conservation. Object's location is reported periodically, and exact location must be estimated using the last known location and uncertainty value. Uncertainty depends on location measurement error, speed of the moving objects, last reported direction, elapsed time etc. Uncertain object is not represented by exact location but uncertainty region, which is represented by a probability-density function (PDF) [12]. Ordinary clustering methods are designed to handle point-valued data, and cannot handle uncertain data. Thus, uncertain data can be transformed in point presented data using the centre of object's PDF and apply ordinary clustering algorithm. However, in [2] is proved that clustering using PDF has better results. Objects location can be uncertain in two different ways, like existential uncertainty and value uncertainty. Object is existentially uncertain, when it is uncertain whether that object exists [17]. In a relational database object is associated with a probability value that indicates the confidence of its presence [3]. In the second case, object is known to exist, but its value is uncertain. Object is modelled as a minimum bounding region (MBR), which bounds all possible location values [2, 19]. In [4] are proposed indexing solutions for range queries over uncertain data. The same authors proposed solutions for aggregate queries such as the nearest neighbour queries in [3]. The result of cluster analysis is used to identify the most probable values of model parameters, like means of Gaussian mixtures, to identify high-density connected regions [6, 10, 11] like areas with high population density, or to minimise an objective function [15] like the total squared distance to cluster centres. In this paper the last case is studied, where the goal is to minimise the total squared distance to cluster centres. Different distance measures, like city-block distance, Euclidian distance or Minkowski distance are used to measure distance from objects to clusters [7]. In UK-means and other clustering methods PDF is represented by a set of sample values, and computational costs are significantly increased in comparison to simple distance calculation. To improve accuracy a large number of samples is needed to represent each PDF. Distance is calculated for all samples, thus computational costs are higher than in simple distance calculation [12]. In this paper 196 samples are used for object's PDF, and there is 196 distance calculations between object and cluster. Basic clustering algorithm UK-means [2] is ineffective, because expected distance (ED) is calculated from all objects to all clusters. In MinMax pruning [16] some clusters are pruned without calculating ED, thus it is significantly more effective than UK-means. Clustering uncertain data using Voronoi diagrams [5] is presented in [1, 8], as answer to Reverse Nearest Neighbour (RNN) queries [9]. Authors proved that Voronoi pruning is more effective than MinMax pruning. Improved Bisector Pruning method is presented in [13], and compared to existing methods. Authors proved that Improved bisector pruning method has significantly shorter execution times of clustering process than previous methods. In [14] Segmentation of Data Set Area (SDSA) method is presented. In SDSA method, data set area is divided in many small segments. Each segment is observed separately and only objects and clusters in observed segment, and clusters in neighbouring segments are observed. Thus, the number of objects and clusters in observations is decreased. The SDSA method can be implemented with all mentioned pruning methods. In this paper SDSA method is combined with Improved bisector pruning method, because it is a very fast cluster pruning method. The result is a new SDSA-IB method. The SDSA method is used for segmentation of data set area and Improved bisector is used for cluster pruning. By synthesis of these two methods, the new method has the best pruning qualities taken from Improved bisector pruning, and using SDSA method data set area is divided in segments and each segment is processed on parallel processors.

2 Comparison of existing methods

Objects location can exhibit uncertainty from two different aspects, as existential uncertainty and as value uncertainty. Object is existentially uncertain when doubts exist in the object's existence [17]. In a database object is associated with a probability value that indicates the confidence of its presence [4]. Efficient query evaluation on probabilistic databases is well explained in [5].

Value uncertainty appears when the object is known to exist, but its value is uncertain and location is not precise. Object is modelled as a minimum bounding region (MBR), which bounds all possible location values. In [1], MBR is described as probability density function. The clustering of objects with value uncertainty, such as location uncertainty, is studied in this paper. Object's locations are represented by probability density function, which is represented by sets of sample values. To improve accuracy a large number of samples is needed. Distance must be calculated for all samples, thus computational cost is higher than in simple distance calculation [11]. Primarily, the used terms are defined.

Definition 1: Uncertain objects are collection of data $O=\{o_1,\ldots, o_n\}$ in m dimensional space R^m, where distance between two objects is:

$$\mathrm{d}(o_i, c_j) \geq 0. \tag{1}$$

Definition 2: Probability density function of each object at each point $x \in R^m$ is $f_i(x) > 0 \;\; x \in R^m$, where for all points inside MBR:

$$\int\limits_{x \in R^m} f_i(x)\mathrm{d}x = 1. \tag{2}$$

Definition 3: Expected distance from object o_i to any point y is calculated using the next formula:

$$\mathrm{ED}(o_i, y) = \int\limits_{x \in A_i} \mathrm{d}(x, y) \cdot f_i(x)\mathrm{d}x, \tag{3}$$

A_i is finite region and $f_i(x) = 0$ outside region A_i.

Definition 4: Goal of clustering is to find a set of clusters points $C=\{c_1,\ldots,c_m\}$ and all relations among objects and clusters $h{:}\{1,\ldots,n\}\rightarrow\{1,\ldots,m\}$, which total expected distance from objects to cluster

$$\mathrm{TED} = \textstyle\sum_{i=1}^{n} \mathrm{ED}(o_i, c_{h(i)}), \tag{4}$$

is minimised. In clustering algorithm UK-means [1, 11], expected distance (ED) is calculated from all objects to all clusters. UK-means algorithm is shown in Fig. 1.

```
Choose $k$ arbitrary points as $c_j$ ($j = 1, \ldots, k$)
repeat
  for all $o_i \in O$ do /*assign objects to clusters*/
      for all $c_j \in C$ do
          Compute ED($o_i$,$c_j$).
      $h(i) \leftarrow j^*$ where $j^*$ minimises ED($o_i$,$c_j$)
      (among $c_j \in C$)
  for all $j = 1, \ldots, k$ do
      $c_j \leftarrow$ centroid of { $o_i \in O \mid h(i) = j$}
until $C$ and $h$ become stable
```

Figure 1 UK-means algorithm

As earlier stated, ED calculations are expensive, thus UK-means algorithm is ineffective. Each object o_i is assigned to cluster c_j, whose representative cluster point has the smallest expected distance ED(o_i, c_j) to object o_i. After all objects are assigned to clusters, cluster representative points are recomputed as the mean of all objects assigned to the cluster c_j. These steps are repeated until the solution converges.

Expected distance is calculated for each object cluster pair. Expected distance calculation requires numerically integrating function weighted by the corresponding probability density function. PDF is represented by a large number of sample points, and computational costs are very high. To improve performance of UK-means different algorithms are developed. Their intention is to reduce the time spent on ED calculations. In the next chapters a few of them are presented. In [16] MinMax pruning method is presented. It is more effective than UK-means, because many ED calculations are avoided. In MinMax pruning method minimum bounding rectangle

MBR is used to avoid unnecessary expected distance calculation. MBR is the smallest rectangle that is equal to finite region A_i as shown in Fig. 2.

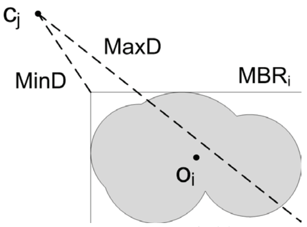

Figure 2 MBR of object

Using MBR and inexpensive Euclidian distance calculations some clusters are pruned as candidates for an object. Thus expected distances from those clusters to object are not computed. For each object minimum distance to cluster is defined:

$$\mathrm{MinD}(o_i,\ c_j) = \min_{x\epsilon \mathrm{MBR}_i}\mathrm{d}(x,\ c_j). \tag{5}$$

Maximum distance to cluster:

$$\mathrm{MaxD}(o_i,\ c_j) = \max_{x\epsilon \mathrm{MBR}_i}\mathrm{d}(x,\ c_j). \tag{6}$$

Smallest distance among all maximum distances:

$$\mathrm{Min\ MaxD}(o_i,\ c_j) = \min_{c_j\epsilon C}\{\mathrm{MaxD}(o_i,\ c_j)\}. \tag{7}$$

It is obvious that minimum distance from object to cluster is less than expected distance, and maximum distance is higher than expected distance from object to cluster, as shown in the next formula:

$$\mathrm{MinD}(o_i,\ c_j) \le \mathrm{ED}(o_i,\ c_j) \le \mathrm{MaxD}(o_i,\ c_j). \tag{8}$$

Then if it is satisfied:

$$MinD(o_i,\ c_p) \ge MaxD(o_i,\ c_j) \tag{9}$$

Without computing ED, cluster c_p is pruned from object o_i, and execution time is shortened. MinMax pruning algorithm is shown in Fig. 3.

```
Choose $k$ arbitrary points as $c_j$ ($j = 1, \dots, k$)
repeat
  for all $o_i \in O$ do /*assign objects to clusters*/
    for all $c_j \in C$ do
      Compute ED($o_i$,$c_j$).
    $h(i) \leftarrow j^*$ where $j^*$ minimises ED($o_i$,$c_j$)
    (among $c_j \in C$)
  for all $j = 1, \dots, k$ do
    $c_j \leftarrow$ centroid of $\{ o_i \in O \mid h(i) = j\}$
until $C$ and $h$ become stable
```

Figure 3 MinMax method algorithm

2.1 Improved bisector pruning method

Improved Bisector Pruning [13] inherits principles of Voronoi and Bisector pruning method and it is improvement of that method. Bisector pruning is a side product of Voronoi diagrams construction, and bisectors are calculated at little extra cost after Voronoi diagrams construction. Thus Bisector pruning is combined with Voronoi cell pruning [8]. In Improved Bisector pruning Voronoi diagrams are not constructed and bisectors are calculated using formula (12), what is more effective than Voronoi pruning. Bisector is line segment that is perpendicular to the line segment joining c_p and c_q, and that passes through the mid-point of the line segment. For each pair of clusters c_p and c_q in $C = \{c_1,\dots,c_m\}$ bisector $B_{p/q}$ is calculated using the following formula:

$$a = -\left(\frac{x_{cp} - x_{cq}}{y_{cp} - y_{cq}}\right), \tag{10}$$

$$b = \frac{x_{cp}^2 - x_{cq}^2 + y_{cp}^2 - y_{cq}^2}{2(y_{cp} - y_{cq})}, \tag{11}$$

$$B_{p/q} = a \times x + b. \tag{12}$$

Bisectors are constructed using representative cluster points (x_{cp}, y_{cp}) and (x_{cq}, y_{cq}). For each cluster pair is checked, if MBR_i of object o_i completely lies on the same side of bisector $B_{p/q}$ as cluster c_p, and if so, then cluster c_q is pruned from object o_i. And for opposite situation, if MBR_i of object o_i completely lies on the same side of bisector $B_{p/q}$ as cluster c_q, then cluster c_p is pruned from object o_i. Pruned cluster instantly is removed from cluster candidates. For 50 clusters there are 50×50 bisectors calculations. But for once pruned cluster, remaining bisectors are not constructed, and this number is significantly reduced. To prune clusters the next properties must be satisfied:

$$\begin{gathered}(y_{bcp} > y_{cp} \text{ and } y_{boi} > y_{oi}) \\ \text{or } (y_{bcp} < y_{cp} \text{ and } y_{boi} < y_{oi}).\end{gathered} \tag{13}$$

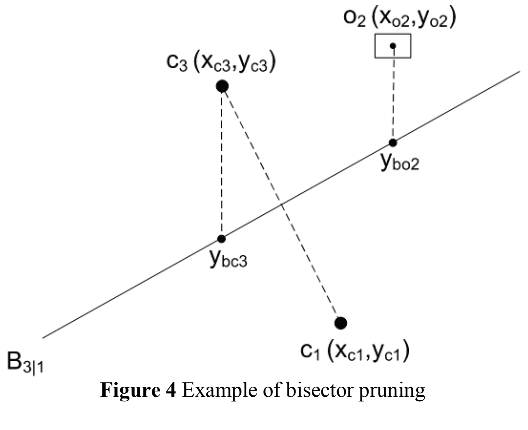
o_2 (x_{o2},y_{o2})
c_3 (x_{c3},y_{c3})
y_{bo2}
y_{bc3}
$B_{3|1}$
c_1 (x_{c1},y_{c1})

Figure 4 Example of bisector pruning

In Fig. 4 is explained the principle used in the above formula. For points on the perimeter of MBR_2 it is checked if they lie on the same side of bisector as cluster c_3. To check this statement coordinate x_{c3} of cluster c_3 and coordinate x_{o2} of point on the perimeter of MBR_2 are included in bisector formula and results y_{bc3} and y_{bo3} are obtained. From Fig. 4 is obvious, that object o_2 and cluster c_3 are on the same side of the bisector $B_{3/1}$ according to formula (13). Obtained result y_{bc3} is smaller than original coordinate y_{c3} of cluster c_3, and also y_{bo3} is smaller than original coordinate y_{o2} of point on perimeter of MBR_2. And for the opposite as shown in Fig. 5, the obtained result y_{bc3} is higher than y_{c3}, and y_{bo2} is higher than y_{o2}. First condition in formula (13) is satisfied, thus object o_2 is on the same side of bisector as cluster c_3. All above steps are repeated for peak points on MBR_2. If all points satisfy the formula, cluster c_1 is pruned from object o_2.

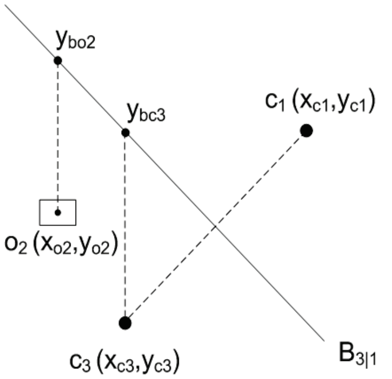

Figure 5 Example of bisector pruning

After iterating all cluster pairs, most of the clusters are pruned, and only for few remaining clusters ED calculation is needed. In Voronoi pruning, if all clusters except one are not pruned, ED must be calculated for all clusters. Thus, Voronoi diagram is combined with Bisector pruning, to prune some clusters and avoid unnecessary ED calculations. Besides that, calculations using formulas (10 ÷ 13) are faster than Voronoi diagrams construction and checking for each object if its MBR_i completely lies inside Voronoi cell. Improved Bisector pruning is described by the following algorithm:

```
Choose $k$ arbitrary points as $c_j$ ($j = 1, . . . , k$)
repeat
  for all $o_i \in O$ do /*assign objects to clusters*/
      for all $c_j \in C$ do
          Compute ED($o_i$,$c_j$).
      $h(i) \leftarrow j^*$ where $j^*$ minimises ED($o_i$,$c_j$)
      (among $c_j \in C$)
for all $j = 1, . . . , k$ do
      $c_j \leftarrow$ centroid of { $o_i \in O \mid h(i) = j$}
until $C$ and $h$ become stable
```

Figure 6 Improved bisector method algorithm

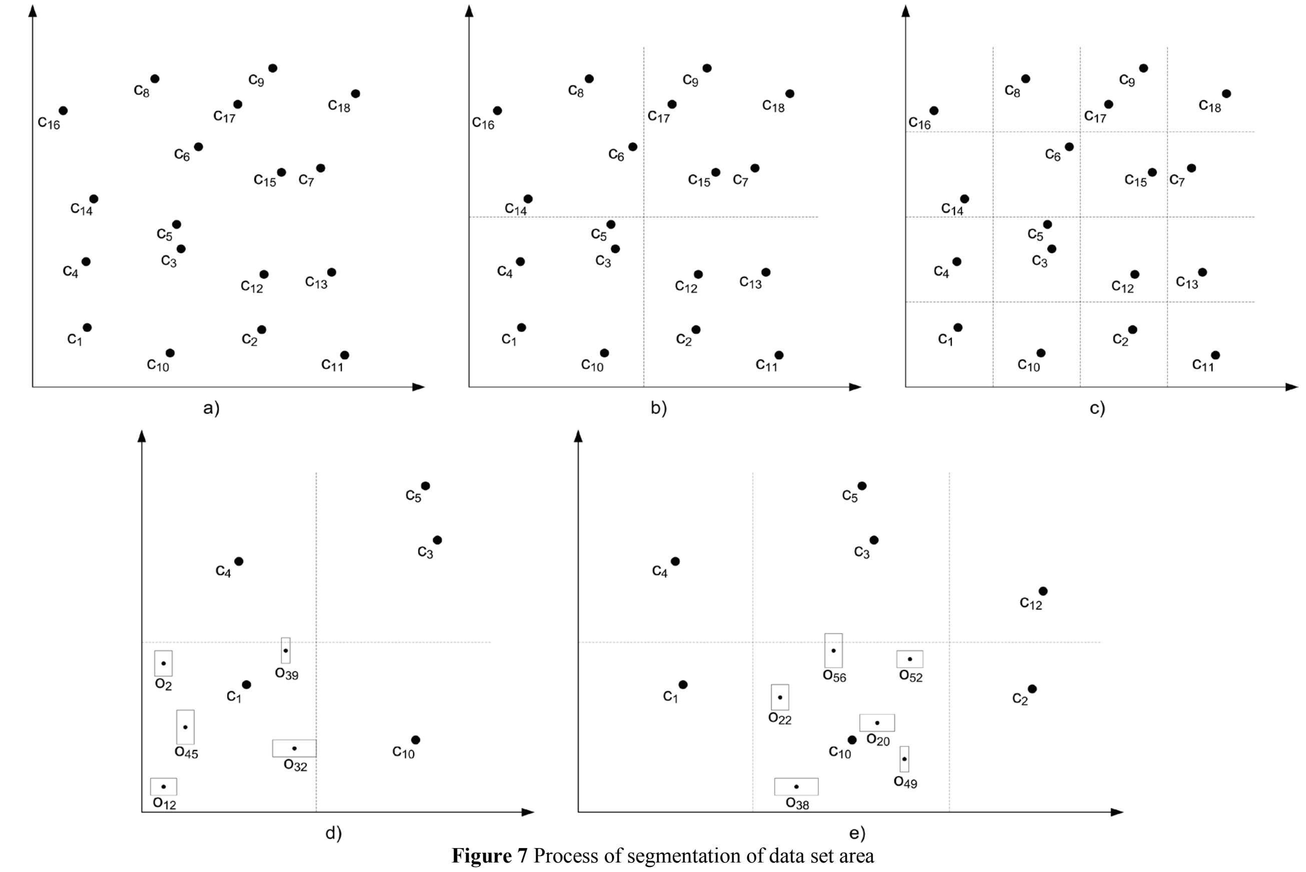

Figure 7 Process of segmentation of data set area

2.2 SDSA method

In SDSA method data set area is divided in small segments. Segments are parts of total data set area as shown in Fig. 7. All segments have the same size, and they are rectangular. Only objects in one segment are observed and pairs with clusters from that and neighbouring segments. Thus number of object clusters observations is decreased. This method can be combined with all existing pruning methods such as UK-means, MinMax pruning and Voronoi cell pruning. Experiments showed that SDSA method combined with other pruning methods speeds up clustering process, depending on the number of clusters and objects. Improvement of clustering process is reversely proportional to the size of segments. If segments are smaller than clustering the process is more effective. By decreasing the size of segments, the number of observed object clusters pairs is decreased, as shown in Fig. 7. In Fig. 7a is shown entire data set area C, and in Fig. 7b data set area is divided into four small clusters segments S_{SDSA}. Each segment is divided in four smaller segments. In Fig. 7c are shown 16 small segments S_{SDSA} and that is final number of segments in this paper. In Fig. 7d and Fig. 7e are shown enlarged areas with object inside them. For 1600 objects and 64 clusters, there are 102 400 object cluster pair calculations. If SDSA method is used, data set area is divided into 16 smaller segments S_{SDSA}, and clusters area C is divided into 4 small areas C_{SDSA}. Average number of objects in one segment is 100, and average number of clusters is four. Each segment is observed separately. Number of observed objects is 100, number of observed clusters 16 and number of segments is 16. Thus, total number of object cluster pair calculations is 25 600, what is four times less calculations. In this case there is no need to check all clusters, but only clusters which are near and surround the area. All remaining clusters are pruned for all objects inside the area. To conclude, SDSA pruning decreases total number of calculations by four times in this case. Decreasing total number of calculations is proportional to decreasing of clusters area. However, segmentation has size limits, which are dependent on number of clusters and their positioning. Segments must be surrounded by clusters, and if number of clusters is high, then segments can be very small and speed up the clustering process.

3 Experiments

For data set of n objects is generated with their corresponding uncertainties described by MBRs. All objects are located in [0,100] × [0,100] 2D space. MBRs are generated to have random side length for each object, but bounded with maximum length of 10. For each object MBR is divided into a $\sqrt{s} \times \sqrt{s}$ grid, where s is number of samples per object's probability density function. Probability for each cell is randomly generated and the sum of all probabilities must be equal to 1. All objects are randomly positioned in space. For this data set, the initial cluster centres are chosen uniformly from 2D space. Basic data set is represented in Table 1. Each experiment is repeated 20 times to get a more accurate average result. The experimental results are compared to ensure that each method has the same clustering results. All methods are implemented in MATLAB and carried out on PC with a processor Intel Core i7-870 2,93 GHz, and 4 GB of main memory.

Table 1 Basic data set

Parameter	Description	Value
n	number of uncertain objects	10 000
k	number of clusters	49
d	maximum side length of MBR	10
s	number of sample point	196

3.1 Basic data set experimental results

First experiment is conducted with basic parameters shown in Tab. 1. Results are shown in Tab. 2. In Tab. 2 execution times are given in seconds, and serial process is compared to parallel processes on two cores and four cores. SDSA-IB method is processed as serial process and has execution time of 55,146 seconds. SDSA-IBP2 and SDSA-IBP 4 methods are parallel processed and execution times are 35,531 and 30,767 seconds, that is reduction of 36,57 % and 44,21 %.

Table 2 Basic data set experimental results

Method	Execution time (s)
SDSA-IB	55,146
SDSA-IBP 2	35,531
SDSA-IBP 4	30,767

To conclude, for basic parameters it is recommended to use SDSA-IBP 4 method, because execution time is the shortest. In Fig. 8 is shown how segments are processed in SDSA-IBP 2 method. Grey segments marked with number one are processed by the first core, and white segments marked with number two are processed by the second core.

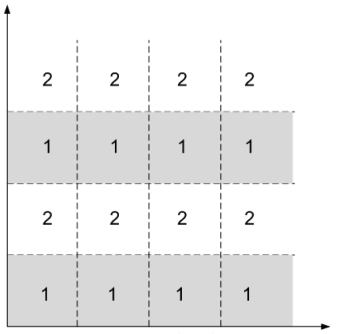

Figure 8 Segments processing in two core processor

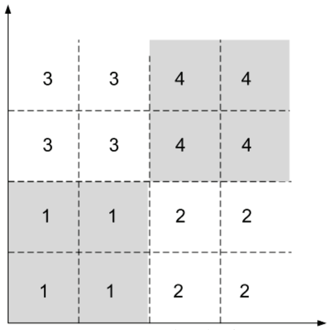

Figure 9 Segments processing in four core processor

In Fig. 9 is shown how segments are processed in SDSA-IBP 4 method. Grey segments are processed by the first and fourth core, and white segments are processed by the second and third core. In each core are processed four inside and four outside segments. In that way better distribution of execution time is achieved, because the processing of inside segments is more demanding. They are surrounded with more cluster segments than outside segments. For example, inside segment is surrounded with eight cluster segments, but outside segment is surrounded with five cluster segments. Thus, it is important that each process has the same number of inside and outside segments.

3.2 Experiments with various numbers of objects

In these experiments various numbers of objects are used, but other parameters retained basic values. Experiments started with 5000 objects and ended with

40 000. Experimental results are shown in Fig. 10. From Fig. 10 it is visible that for small number of objects (less than 5000) parallelization has minor execution time improvement to serial method, because communication between processes has significant contribution in total execution time.

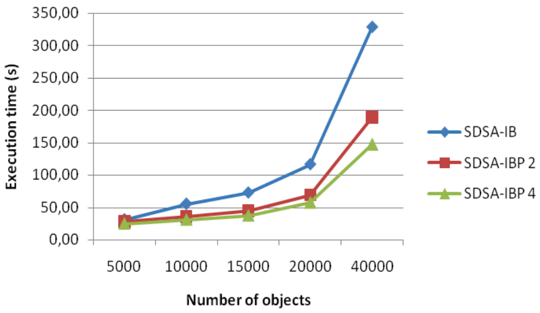

Figure 10 Experimental results for various numbers of objects

However, as the number of objects is increased, communication between processes is negligible in total execution time, and benefits of parallelization are visible. Execution times ratio of SDSA-IB and SDSA-IBP 2 is shown in Fig. 11.

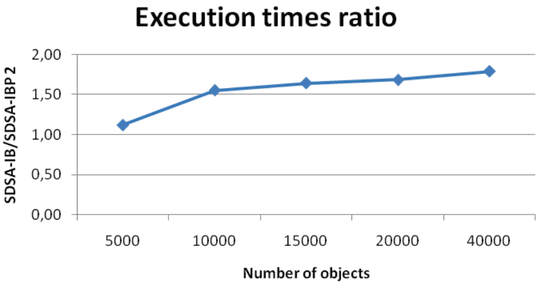

Figure 11 SDSA and SDSA-P 2 cores execution times ratio

Ratio for SDSA-IB and SDSA-IBP 4 method is similar, and there is no need to present it in figure. From Fig. 11 is obvious that execution time ratio rises with number of objects. For 10 000 objects execution times of parallel methods are significantly better. Finally, for 40 000 objects execution time of SDSA-IB method is 328,613 seconds, SDSA-IBP 2 method 189,394 seconds and SDSA-IBP 4 method 147,519 seconds. In this case execution times improvements are 43,47 % and 55,11 %. Thus, it is recommended to use parallel method for larger number of objects, where clustering process is more demanding.

3.3 Experiments with various numbers of clusters

In this experiment, number of clusters is varied from 16 to 144 with basic values for other parameters. Result is shown in Fig. 12. As number of clusters is increased, cluster centres are closer and there is less probability for successful cluster pruning for some object. Consequently, more ED will have to be calculated to assign object to the cluster. ED calculations has significant contribution to total time, thus with larger number of clusters parallel processes are better than serial process, because ED calculations are distributed to more processes and each process is calculated only one part of ED calculations. Execution times ratio of SDSA-IB and SDSA-IBP 2 is shown in Fig. 13.

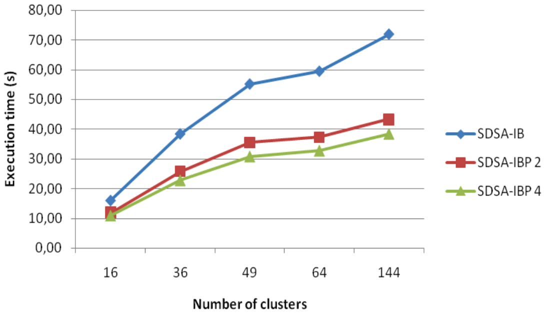

Figure 12 Experimental results for various numbers of clusters

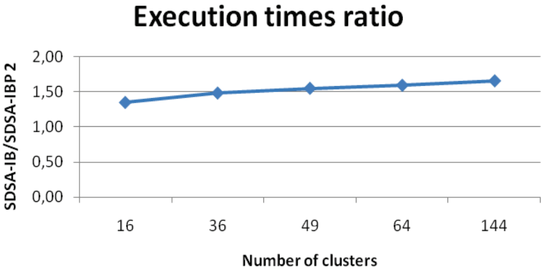

Figure 13 SDSA and SDSA-P 2 cores execution times ratio

Ratio for SDSA-IB and SDSA-IBP 4 method is similar, and there is no need to present it in figure. From Fig. 13 can be concluded that parallelization advantages come in place as the number of clusters is high, because execution times ratio is higher as the number of clusters is increased.

3.4 Experiments with various size of MBR

In this experiment, size of MBR is varied from 1 to 20 with basic values for other parameters. Results are shown in Fig. 14. It is visible that execution time is increased with the size of MBR. As the size of MBR is increased, there is more probability that MBR of object will overlap with the bisectors causing unsuccessful pruning. Consequence is more ED calculations.

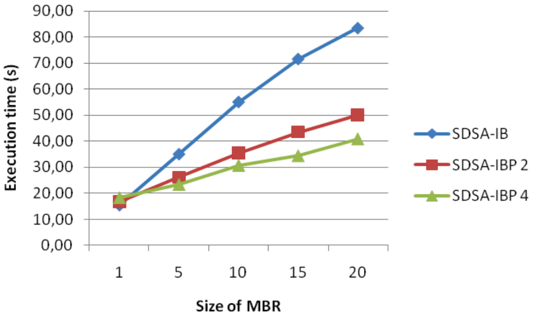

Figure 14 Experimental results for various size of MBR

As the size of MBR is increased, parallel processes are more effective, because ED calculations are again distributed to more processes and by each process is executed one part of ED calculations. However, serial process must calculate all ED calculations. Thus, in this case parallel methods are more effective. In Figure 15 is shown execution time ratio for various sizes of MBR. Ratio is from 1,01 for *MBR*=1 to 1,67 for *MBR*=20.

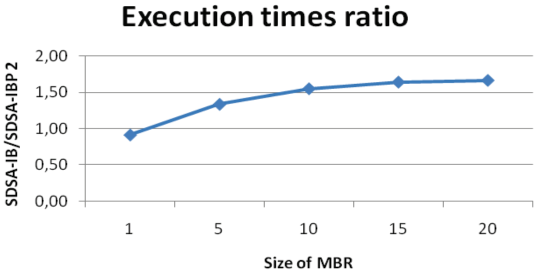

Figure 15 SDSA and SDSA-P 2 cores execution times ratio

4 Conclusion

In this paper is presented a new method for parallel clustering of uncertain objects. The new method is called SDSA-IB. It is a combination of improved bisector pruning method and SDSA method, which are presented in our previous papers. Using this new method execution time of clustering process is reduced and usefulness of finding uncertain objects in distributed databases is improved. SDSA methods were the best choice for parallelization. Each segment can be independently observed, what provided possibility for parallel execution. SDSA method can be easily parallelized with minor changes to serial method. Experimentally is proved effectiveness of parallel methods and execution time improvement as number of objects, clusters and size of MBR are increased, because communication between processes is less important in total execution time. Experiments are conducted as two parallel processes on two cores, and four parallel processes on four cores. However, in practice each segment can be one parallel process. In our experiments sixteen segments are used, and theoretically 16 parallel processes could be used, but the number of processes is limited by hardware configuration. The results of experiments showed that parallel method outperforms the existing serial methods, with same costs, because hardware configuration is the same. In future work, algorithm for segmentation will be improved, for optimal calculation sizes and number of segments, according to the number and position of clusters. Using this algorithm, maximum number of segments will be used for every clustering. In that way, the number of parallel processes is maximised and limited by hardware configuration. Furthermore, the presented method will be used for clustering sensor networks data, and network nodes clustering.

5 References

[1] Kao, B.; Lee, S. D.; Lee, F. K. F.; Cheung, D. W. I.; Ho, W. S. Clustering Uncertain Data using Voronoi Diagrams and R-Tree Index. // Knowledge and Data Engineering, IEEE Transactions, Sept. 2010, pp. 1219-1233.

[2] Chau, M.; Cheng, R.; Kao, B.; Ng, J. Uncertain data mining: An example in clustering location data. // In PAKDD Singapore 2006, pp. 199–204.

[3] Cheng, R.; Kalashnikov, D.; Prabhakar, S. Querying imprecise data in moving object environments. // IEEE TKDE 2004, 16(9), pp. 1112–1127.

[4] Cheng, R.; Xia, X.; Prabhakar, S.; Shah, R.; Vitter J. Efficient indexing methods for probabilistic threshold queries over uncertain data. // In Proc. of VLDB Conference 2004, pp. 876-887.

[5] Dehne, F. K. H. A.; Noltemeier, H. Voronoi trees and clustering problems. // Inf. Syst. 1987, 12(2), pp. 171–175.

[6] Ester, M.; Kriegel, H. P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. // In Proc. of ACM SIGKDD Conference 1996, pp. 226-231.

[7] Ichino M.; Yaguchi H. Generalized Minkowski metrics for mixed feature type data analysis. // IEEE TSMC 1994, 24(4), pp. 698V–708.

[8] Kao, B.; Lee, S. D.; Cheung, D. W.; Ho, W. S.; Chan, K. F. Clustering Uncertain Data using Voronoi Diagrams. // Data Mining, 2008. ICDM '08. Eighth IEEE International Conference 2008, pp. 333 – 342.

[9] Korn, F.; Muthukrishnan, S. Influence sets based on reverse nearest neighbor queries. // Proceedings of the 2000 ACM SIGMOD International conference on Management of data, 2000, pp. 201-212.

[10] Kriegel, H. P.; Pfeifle, M. Density-based clustering of uncertain data. // In KDD 2005, pp. 672–677.

[11] Kriegel, H. P.; Pfeifle, M. Hierarchical density-based clustering of uncertain data. // In Proc. of IEEE ICDM Conference 2005, pp. 689-692.

[12] Xiao, L.; Hung, E. An Efficient Distance Calculation Method for Uncertain Objects. // Computational Intelligence and Data Mining, CIDM 2007, pp. 10–17.

[13] Lukić, I.; Köhler, M.; Slavek, N. Improved Bisector Pruning for Uncertain Data Mining. // Proceedings of the 34th International Conference on Information Technology Interfaces, ITI 2012, pp. 355-360.

[14] Lukić, I.; Köhler, M.; Slavek, N. Segmentation of Data Set Area Method in Clustering of Uncertain Data, Proceedings of the Jubilee 35th International ICT Convention – MIPRO 2012, pp. 420-425.

[15] Macqueen, J. Some methods for classification and analysis of multivariate observations. // In Proc. 5th Berkeley Symposium on Math. Stat. and Prob. 1967, pp. 281–297.

[16] Ngai, W. K.; Kao, B.; Chui, C. K.; Cheng, R. Efficient clustering of uncertain data. // In ICDM 2006, pp. 436–445.

[17] Nilesh, N. D.; Suciu, D. Efficient query evaluation on probabilistic databases. // In Proc. of VLDB Conference 2004, pp. 864–875.

[18] Ruspini, E. H. A new approach to clustering. // Information and Control 1969, 15(1), pp. 22–32

[19] Wolfson, O.; Sistla, P.; Chamberlain, S.; Yesha, Y. Updating and querying databases that track mobile units. // Distributed and Parallel Databases, 7, 3(1999), pp. 257-288.

Authors' addresses

Dr. sc. Ivica Lukić, dipl. ing. el.
J. J. Strossmayer University of Osijek
Faculty of Electrical Engineering
Cara Hadrijana bb
31000 Osijek, Croatia
E-mail: ivica.lukic@etfos.hr

Doc. dr. sc. Ninoslav Slavek, dipl. ing. el.
J. J. Strossmayer University of Osijek
Faculty of Electrical Engineering
Cara Hadrijana bb
31000 Osijek, Croatia
E-mail: ninoslav.slavek@etfos.hr

Mirko Köhler, dipl. ing. el.
J. J. Strossmayer University of Osijek
Faculty of Electrical Engineering
Cara Hadrijana bb
31000 Osijek, Croatia
E-mail: mirko.kohler@etfos.hr