ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)

https://doi.org/10.17559/TV-20151109144645
Original scientific paper

Improved Algorithm for Distributed Points Positioning Using Uncertain Objects Clustering

Ivica LUKIĆ, Mirko KÖHLER, Tomislav GALBA

Abstract: Positioning of mobile objects that require communication with some kind of online service application is a very challenging task. Proper positioning with minimal deviation is an important mobile service system (MSS), e.g. taxi service used in this paper. It will perform all tasks for the users and reduce the overall travel distance. This paper is focused on the development of an algorithm that will find the optimal position for an MSS object and upgrade the system quality using uncertain data clustering. If the best position for the MSS is found, then the response time is short, and the system tasks could also be performed in usable time. The improved bisector pruning method is used for clustering stored data of mobile service system objects to provide the best position of system objects. As the best position of MSS objects, we use cluster centres. Using clustering, the total expected distance from end users to the service system is minimal. Therefore, the MSS is more efficient and has more time to fulfil additional tasks at the same time.

Keywords: clustering; data mining; Euclidian distance; information service; positioning

1 INTRODUCTION

The Mobile Service System (MSS) stores large amount of data during time. However, it is not always on time and can therefore be uncertain. To extract certain data from a large amount of uncertain data is a challenging task [1, 2]. Various methods are proposed to extract useful information from uncertain data such as improved bisector pruning [3], MinMax pruning [4, 5] and Voronoi pruning [6, 7]. Clustering is used to find mutually similar objects which are near to the cluster centre and add them to the same cluster. All objects in the same cluster will have a minimized total expected distance to the cluster centre. It can be concluded that MSS will locate its objects near to or in the cluster centre. Many clustering algorithms are used to track mobile devices and objects [8, 9].

Object location uncertainty is represented as an uncertainty region [10]. Mathematically, it is explained as a Probability Density Function (PDF). In this paper, a taxi service location is presented in 2D dimensions with a 2D uncertainty. We could represent uncertainty with Gaussian distributions [8] whose density function is exponentially dropping and, finally, the probability density outside the observed region will be zero. In the mobile system, each object can be bounded by a finite 2D region. The 2D bounding region size is defined using the maximum speed of the object and time from the last data update. Clustering time of MSS data should be short to perform real time applications. Expected distances (ED) calculation is a challenging task and takes most of the computational time in the clustering process [11, 12]. Each PDF is represented by numerous sample points [10] and the expected distance calculation requires a lot of computational power. For each sample, the distance to cluster centre is calculated, which makes the computational costs high [13]. In [4-6], different algorithms for cluster pruning are presented. These pruning algorithms avoid many ED calculations and reduce the execution time. In this paper, we use pruning algorithms to eliminate some clusters as candidate clusters without ED calculations. To eliminate such clusters as candidates for an object, some other clusters should be closer.

2 SURVEY OF EXISTING METHODS

An object location can pose existential uncertainty and value uncertainty. An object is existentially uncertain if it is uncertain whether that object exists. Mobile service system object has a probability value that shows the confidence of its existence [9]. Query evaluation and database management on probabilistic databases is explained in [1]. In value uncertainty, the object is known to exist, but the object's value is uncertain. Because of value uncertainty, the object's location is not precise and such objects are called uncertain objects. Uncertain objects are modelled as a minimum bounding region (MBR), which bounds all possible location values. In [2], [8, 9], MBR is described by a PDF. In this paper, clustering objects with value uncertainty, such as location uncertainty, are studied. Cluster analysis is used to minimize the total squared distance from objects to cluster centres. Distance can be measured, for example a city block distance [15], Minkowski distance [16], Euclidian distance, etc.

Data uncertainty is represented by a probability density function, which is represented by sets of sample values, and a large number of samples that are needed to improve the accuracy. The distance is calculated between all object samples and the computational cost is higher than in the simple distance calculation where there is only one sample [10]. In the basic clustering algorithm UK-means, the expected distance (ED) is calculated from all objects to all clusters which is ineffective [13]. MinMax pruning and Voronoi pruning methods are significantly more effective than the UK-means method. In Voronoi pruning, the geometric structure of Rm is observed because of using Voronoi diagrams [17]. These methods can be combined with SDSA [18]. The SDSA method is used for segmentation of a data set area into smaller parts. Then, pruning methods are used for efficient cluster pruning in those parts. By synthesizing these methods with the SDSA method, the new method acquires the best pruning qualities taken from the two synthesized methods.

Before describing the methods, some definitions should be explained. Uncertain objects are data collections $O = \{o_1,..., o_n\}$ in space R^m with m dimensions. The distance between objects has to be more than zero. The probability density function for each point inside the MBR in an m dimensional space R^m is greater than zero. The probability density function for all points inside the MBR is calculated using the following formula:

$$\int_{x \in R^m} f_i(x)\mathrm{d}x = 1 \tag{1}$$

From object oᵢ to point y inside the bounding region, the expected distance is calculated using the following formula:

$$ED(o_i, y) = \int_{x \in A} d(x, y) f_i(x)\mathrm{d}x \tag{2}$$

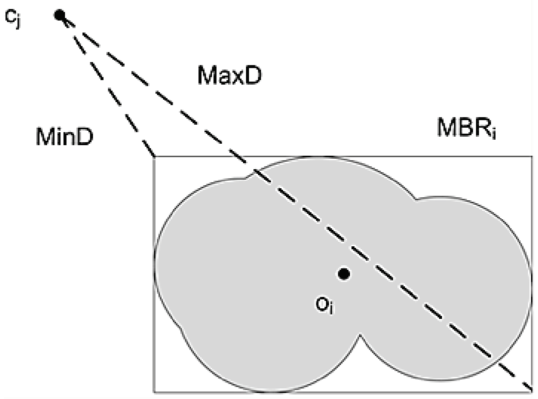

Figure 1 MBR of object, MinD and MaxD distance from object to cluster

The bounded region Ai is finite with defined borders and $f_i(x) = 0$ for all points outside the bounding region. The clustering process is used to find a set of cluster points $C = \{c_1, ..., c_m\}$ and all relations between all Mobile service system objects and clusters $h:\{1, ..., n\} \rightarrow \{1, ..., m\}$. For these relations, the total expected distance from Mobile service system objects to the assigned centre should be minimal. In the MinMax pruning method, the minimum bounding rectangle MBR is used to prune cluster candidates without the expected distance calculation. The MBR is the smallest rectangle that is equal to the finite region as shown in Fig. 1.

Using MBR and inexpensive Euclidian distance calculations, some clusters are pruned as candidates for an object. Thus, the expected distances from those clusters to the object are not computed. For each object, the minimum, maximum and smallest among all maximum distances are defined:

$$MinD(o_i, c_j) = \min_{x \in MBR_i} d(x, c_j) \tag{3}$$

$$MaxD(o_i, c_j) = \max_{x \in MBR_i} d(x, c_j) \tag{4}$$

$$MinMaxD(o_i, c_j) = \min_{c_j \in C}\{MaxD(o_i, c_j)\} \tag{5}$$

It is obvious that the minimum distance between object and cluster is smaller and the maximum distance is larger than the expected distance from an object to a cluster, as shown in the following formula:

$$MinD(o_i, c_j) \le ED(o_i, c_j) \le MaxD(o_i, c_j) \tag{6}$$

If it is satisfied:

$$MinD(o_i, c_p) \ge MaxD(o_i, c_j) \tag{7}$$

Without computing the expected distances, cluster c_p is removed from object o_i, the ED is not calculated, and the execution time is shortened. In the Voronoi pruning method, unlike MinMax pruning, the geometric structure of R^m is observed, which means that spatial relationships between clusters are considered. With a set of clusters $C = \{c_1, ..., c_k\}$, space R^m is divided into k cells with the following property:

$$d(x, c_p) \le d(x, c_q) \quad \forall x \in V(c_p), c_p \ne c_q \tag{8}$$

The next step in the iteration process after constructing Voronoi diagrams is Voronoi cell pruning as shown in Fig. 2. It is checked whether MBR$_i$ of object oi is completely inside Voronoi cell $V(c_j)$. If that condition is satisfied, object o_i can be allocated to cluster c_j and there is no need for ED computation because all other clusters are pruned. Fig. 2 shows that MBR$_2$ of object o_2 is completely inside Voronoi cell V_3, thus object o_2 is assigned to cluster c_3. However, MBR$_1$ is particularly inside Voronoi cell V_3 and object o_1 cannot be assigned to cluster c_3. For remaining objects that are not allocated to any cluster, the expected distance must be calculated.

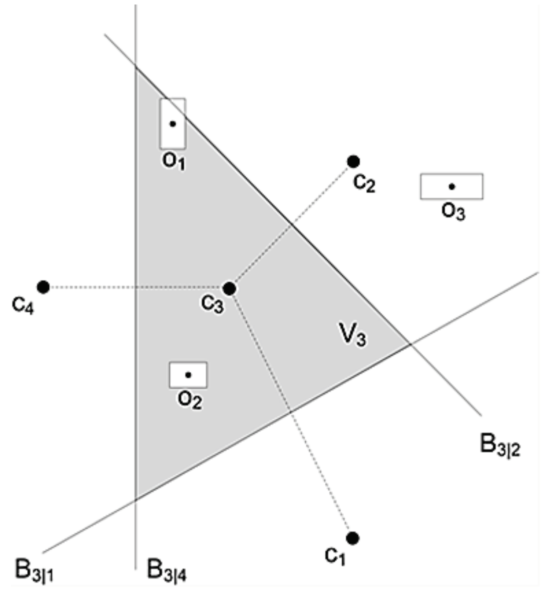

Figure 2 Voronoi cell construction for cluster c₃

3 IMPROVED BISECTOR PRUNING

The Improved Bisector Pruning algorithm has properties of Voronoi and Bisector pruning methods. It is used to upgrade these methods using their advantages and removing shortcomings [6, 7]. Bisectors are boundaries of Voronoi diagrams, and they can be calculated after Voronoi diagrams were constructed, as a small additional calculation cost. Bisector pruning properties are used together with the properties of the Voronoi method [7]. The Improved Bisector algorithm does not construct Voronoi diagrams, i.e. it only calculates bisectors using Eq. (10) which is a significant improvement to Voronoi diagrams construction. Bisector is a line which is perpendicular to and c_q line. Between each cluster pair c_p and c_q, we need to calculate $B_{p/q}$ using the following formulae:

$$a = -\left(\frac{x_{cp} - x_{cq}}{y_{cp} - y_{cq}}\right) \tag{9}$$

$$b=\frac{x_{cp}^2-x_{cq}^2+y_{cp}^2-y_{cq}^2}{2(y_{cp}-y_{cq})} \tag{10}$$

$$B_{p/q}=a\cdot x+b \tag{11}$$

Cluster points (x_{cp}, y_{cp}) and (x_{cq}, y_{cq}) are used to calculate their representative bisectors. Bisector is then used to check if both object oi and its MBR$_i$ are on the same half plain of bisector space $B_{p/q}$ as cluster c_p. If this condition is satisfied, cluster c_q is removed as a candidate cluster for object o_i. And opposite, if object o_i and its MBR*i* are on the same half plain of bisector space $B_{p/q}$ as cluster c_q, then cluster c_p is removed as a candidate cluster for object o_i. These removed clusters are not included in later calculations and are not used for bisector construction with remaining cluster candidates. In Fig. 3, the principles of cluster pruning are explained. All points on the border of MBR$_2$ are mathematically checked using bisector pruning to see whether they are on the same bisector side as candidate cluster c_3. Coordinate x_{c3} of cluster c_3 and coordinate x_{o2} from points on the border on MBR$_2$ are used in bisector pruning calculation to obtain points y_{bc3} and y_{bo2}.

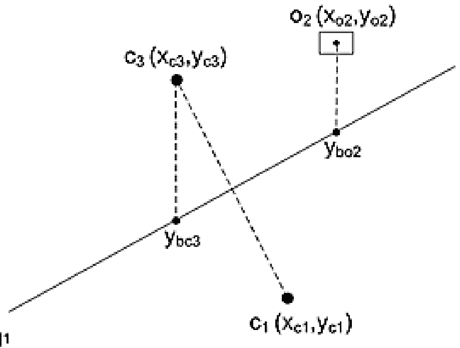

Figure 3 Bisector pruning cluster candidate removal and lower coordinates projection

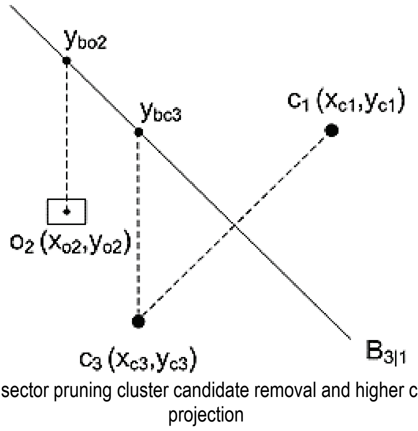

Figure 4 Bisector pruning cluster candidate removal and higher coordinates projection

In Fig. 3, we can see that the observed object o_2 and candidate cluster c$_3$ are on the same half plain of bisector $B_{3/1}$ space. The obtained projection y_{bc3} is lower than projection y_{c3}, and projection y_{bo2} is lower than projection y_{o2} of point on the border of MBR$_2$. On the other hand, in Fig. 4, projection y_{bc3} is higher than projection y_{c3}, and projection y_{bo2} is higher than projection y_{o2}. These steps are repeated for all border peak points on MBR$_2$. Cluster c_1 is removed from object o_2 only if all points on the border meet the condition. After all cluster pairs are compared with some objects, many clusters will be removed and only few clusters will stay as cluster candidates for which ED will be calculated. Voronoi diagrams methods remove all clusters apart from one. In this situation, the expected distance is calculated for all cluster pairs. To improve its properties, this method is combined with the Bisector Pruning method. These combined methods avoid many ED calculations. The Improved Bisector Pruning algorithm is described in [6].

4 EXPERIMENTS

In these experiments, the taxi mobile service system in the city of Osijek is experimented as MSS. We collected data for four months. These data are mined and pruned to establish the best position for taxi vehicles. All data are clustered using the Improved Bisector Pruning method. Without our data collection and clustering, taxi vehicles would be positioned at random locations and the arrival time to the end user would take longer. Clustering is used to save costs and reduce the distance travelled by taxi vehicles. All collected data about MSS are used to calculate proper cluster centres and locate a taxi in those centres. It is proved that vehicles, which are positioned in the cluster centre, will complete the end user task faster than vehicles positioned at random locations.

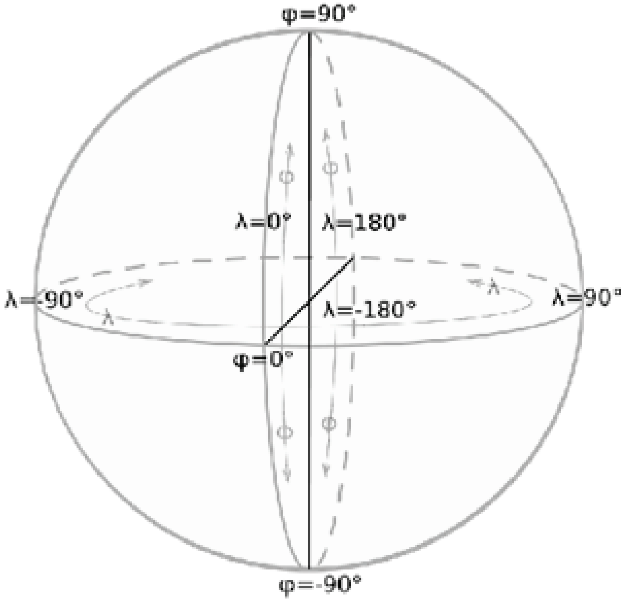

Figure 5 Spherical geo - coordinates representation

We conducted many experiments to prove that clustering existing data about the taxi service will improve the reaction time and reduce the travelled distance. In the conducted experiments, distance from cluster centres to designated tasks is measured. That distance is compared to random distance in which taxi vehicles were located without clustering the end user tasks. Taxi positions are presented in geo–coordinates system, where φ is the Earth latitude, λ is the Earth longitude and r radius of the Earth as shown in Fig. 5.

The clustering algorithm uses Cartesian coordinates which were converted from spherical coordinates [19] using the following formulae:

$$y=r\cdot\cos\varphi\cdot\cos\lambda \tag{12}$$

$$y = r \cdot \cos\varphi \cdot \sin\lambda \qquad (13)$$

Conversion to Cartesian coordinates was made on spherical coordinates of the city of Osijek [20]. The converted Cartesian coordinates are shown in Fig. 6.

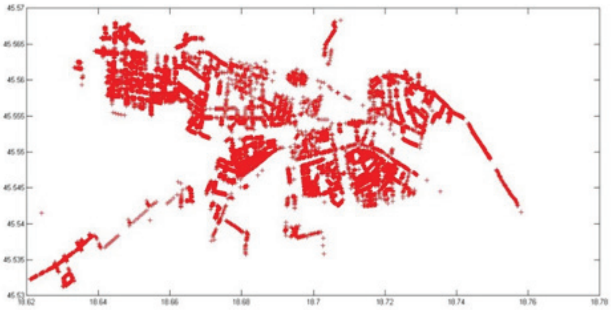

Figure 6 Converted Cartesian coordinates of the city of Osijek

4.1 Experiment with Fifteen Clusters

The experiment used data about the taxi service system in the city of Osijek. Data are clustered using fifteen clusters. Each cluster centre has been assigned a number of taxi vehicles whose number is dependent on the number of tasks around the cluster area. Fifteen cluster centres are shown in Fig. 7 using red circles.

Table 1 Number of tasks settled in each of the fifteen cluster centres

Cluster centre	Number of tasks
Centre No. 1	40
Centre No. 2	65
Centre No. 3	61
Centre No. 4	75
Centre No. 5	62
Centre No. 6	50
Centre No. 7	63
Centre No. 8	85
Centre No. 9	60
Centre No. 10	103
Centre No. 11	58
Centre No. 12	89
Centre No. 13	42
Centre No. 14	75
Centre No. 15	72

In Tab. 1, the number of tasks for each cluster centre is shown. Most of the tasks are positioned in the cluster centre No. 10, and the taxi system should position more taxi vehicles in that area than in other cluster centres as shown in Tab. 1. On the other hand, the smallest number of tasks is positioned around cluster centre No. 1, and the taxi system should position less vehicles in that area. Each cluster centre will position a suitable number of taxi vehicles depending on its load. Cluster centres are named in an orderly way from the left side of the image to the right side of the image. Efficiency of the taxi service is higher because taxi vehicles are positioned effectively.

Taxi vehicles are positioned in the cluster centres and it is certain that the total travelled distance from MSS to end users which use the taxi service is minimal. Clustering improves the efficiency and the travelled distance can be reduced by almost 50%, compared to the travelled distance by a taxi service in which clustering is not used.

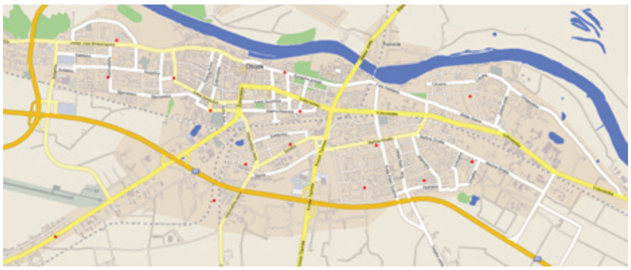

Figure 7 Taxi service of fifteen cluster centres

4.2 Experiment with Nine Clusters

The experiment used data about the taxi service system in the city of Osijek. Data are clustered using nine clusters. Each cluster centre has been assigned a number of taxi vehicles whose number is dependent on the number of tasks around the cluster area. Nine cluster centres are shown in Fig. 8 using red circles. In Tab. 2, the number of tasks for each cluster centre is shown. Most of the tasks are positioned in cluster centre No. 6, and therefore more taxi vehicles should be positioned in that area than in other cluster centres as shown in Tab. 2. On the other hand, the smallest number of tasks is positioned around cluster centre No. 1, and the taxi system should position less vehicles in that area. Each cluster centre will position a proper number of taxi vehicles depending on its load. Cluster centres are named in an orderly way from the left side of the image to the right side of the image. Efficiency of the taxi service is higher because taxi vehicles are positioned effectively. The total travelled distance in this scenario is reduced up to 40%.

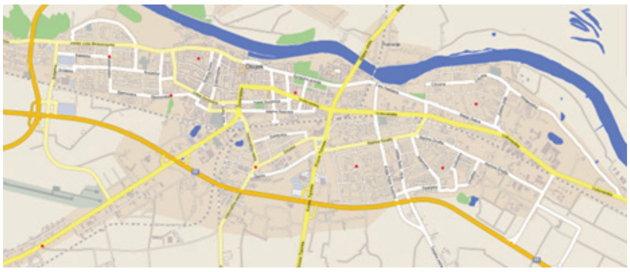

Figure 8 Taxi service of nine cluster centres

Table 2 Number of tasks settled in each of the nine cluster centres

Cluster centre	Number of tasks
Centre No. 1	42
Centre No. 2	130
Centre No. 3	82
Centre No. 4	69
Centre No. 5	117
Centre No. 6	298
Centre No. 7	202
Centre No. 8	83
Centre No. 9	77

4.3 Experiment with Five Clusters

In the experiment we used data about the taxi service system in the city of Osijek. Data are clustered using five clusters. Each cluster centre has been assigned a number of taxi vehicles whose number is dependent on the number of tasks around the cluster area. Nine cluster centres are shown in Fig. 9 using red circles. In Tab. 3, the number of tasks for each cluster centre is shown. Most of the tasks are positioned in cluster centres No. 2 and 3. Consequently, more taxi vehicles should be positioned in that area than in other cluster centres as shown in Tab. 3. In this experiment, all existing data about mobile service system responses in the city of Osijek were divided into five clusters. Depending on the number of tasks that are needed to be accomplished, and which surround the cluster area, in each cluster centre a suitable number of vehicles should be located. Cluster centres are shown in Fig. 9. The total travelled distance is reduced by almost 35%, compared to the travelled distance by a taxi service which does not use clustering.

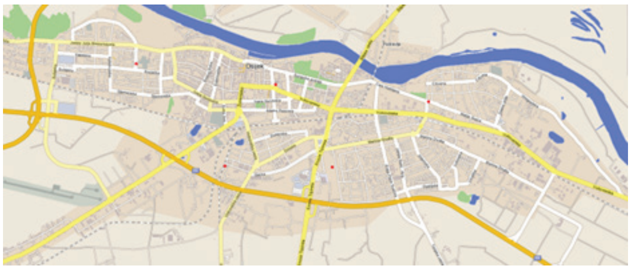

Figure 9 Taxi service of five cluster centres

Table 3 Number of tasks settled in each of the five cluster centres

Cluster centre	Number of tasks
Centre No. 1	128
Centre No. 2	235
Centre No. 3	284
Centre No. 4	203
Centre No. 5	150

The experiments show that if we use more cluster centres, the total expected distance will be more reduced. However, the number of clusters is limited because it is not an easy task to find suitable locations for cluster centres in certain locations in the city of Osijek. Some city areas are not suitable for maintaining the taxi service system. Therefore, it is necessary to find a compromise between the number of clusters and suitable areas.

5 FUTURE WORK

In the future, we will improve our data collection of taxi service data to get better calculations. In addition, we will try to collect more samples for geographical data to represent one uncertain object. These samples are used to calculate the probability density function. This will help us achieve better accuracy to locate a task in a certain sample. The database will incorporate end user tasks for the taxi service in each sample. Sample number of tasks will be stored in the database, and the probability density function will be calculated using the number of tasks. It will help us to precisely predict end user demands for the taxi service.

Prediction will reduce the time and the travelled distance to end user task. Cluster centres with the highest task density will present the most demanding challenge for the taxi service (MSS). Our prediction model will ensure that MSS can deploy taxi vehicles in cluster centres and effectively accomplish end user tasks. Tasks are dependent on various unexpected events, such as accidents, football games, concerts, etc. but unexpected situations will affect these regular predictions.

In future research, our goal is to predict unexpected events and adjust to them. We will conduct experiments and simulate the unexpected events where the end user tasks are dramatically increased and changed with respect to locations. The new prediction model should change the system behaviour in unexpected situations and reduce maintenance costs and increase reliability.

6 CONCLUSION

It is very important to choose the best position for vehicles in MSS. To find the best position, we used the Improved Bisector Pruning method. This method clusters previously collected data about MSS to find the best position for taxi vehicles. Cluster centres are used to minimise the total travelled distance which taxi vehicles should travel to the end user tasks. The first experiment used fifteen clusters, the second used nine clusters and the third experiment used five clusters. In the first experiment, extreme distance reduction of 50% was accomplished due to the larger number of clusters, and proximity of taxi vehicles to the tasks. Some city areas are not suitable for maintaining the taxi service system. Accordingly, it is necessary to find a compromise between the number of clusters and suitable areas. Our experiments proved that clustering improves the effectiveness of MSS and the time for accomplishing the tasks. MSS taxi vehicles are closer to end user tasks and can take on more tasks at the same time. Taxi vehicles are positioned in cluster centres which enables them to minimize the distance and the travel time to designated tasks. The travelled distances are reduced by almost 50% in the first experiment which is a significant improvement. Also, the MSS can deal with more tasks and offer service quality to their users. We will try to include additional parameters in our model and use the model for other mobile systems to significantly improve their quality of service.

7 REFERENCES

[1] Nilesh, D. & Suciu, D. (2004). Efficient query evaluation on probabilistic databases. In *Proc. of VLDB Conference*, 864-875.

[2] Cheng, R., Xia, X., Prabhakar, S., Shah, R., & Vitter, J. (2004). Efficient indexing methods for probabilistic threshold queries over uncertain data. In *Proc. of VLDB Conference*. https://doi.org/10.1016/B978-012088469-8.50077-2

[3] Lukić, I., Köhler, M., & Slavek, N. (2012). Improved Bisector Pruning for Uncertain Data Mining. *Proceedings of the 34th International Conference on Information Technology Interfaces, ITI 2012*, 355-360.

[4] Ngai, W. K., Kao, B., Chui, C. K., Cheng, R. et al. (2006). Efficient clustering of uncertain data. In *ICDM*, 436-445. https://doi.org/10.1109/ICDM.2006.63

[5] Kriegel, H. P. & Pfeifle, M. (2005). Density-based clustering of uncertain data, In *KDD 2005*, 672-677. https://doi.org/10.1145/1081870.1081955

[6] Kao, B., Lee, S. D., Cheung, D. W., Ho, W. S., & Chan, K. F. (2008). Clustering Uncertain Data using Voronoi Diagrams. *Data Mining, ICDM '08. Eighth IEEE International Conference on Date*: 15-19 Dec. 2008. 333-342. https://doi.org/10.1109/ICDM.2008.31

[7] Kao, B., Lee, S. D., Lee, F. K. F., & Cheung, D. W. W. S. H. (2010). Clustering Uncertain Data using Voronoi Diagrams and R-Tree Index. *Knowledge and Data Engineering, IEEE Transactions*, Sept. 2010. 1219-1233.

[8] Wolfson, O., Sistla, P., Chamberlain, S., & Yesha, Y. (1999). Updating and querying databases that track mobile units. *Distributed and Parallel Databases, 7*(3).

https://doi.org/10.1023/A:1008782710752

[9] Cheng, R., Kalashnikov, D., & Prabhakar, S. (2004). Querying imprecise data in moving object environments. *IEEE TKDE, 16*(9), 1112-1127. https://doi.org/10.1109/TKDE.2004.46

[10] Xiao, L. & Hung, E. (2007). An Efficient Distance Calculation Method for Uncertain Objects. *Computational Intelligence and Data Mining, CIDM 2007*, 10-17. https://doi.org/10.1109/CIDM.2007.368846

[11] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proc. 5th Berkeley Symposium on Math. Stat. and Prob.*, 281-297.

[12] Ichino, M. & Yaguchi, H. (1994). Generalized Minkowski metrics for mixed feature type data analysis. *IEEE TSMC, 24*(4), 698-708. https://doi.org/10.1109/21.286391

[13] Chau, M., Cheng, R., Kao, B., & Ng, J. (2006). Uncertain data mining: An example in clustering location data. In *PAKDD*, Singapore, 9–12 Apr. 2006. Springer, 199-204. https://doi.org/10.1007/11731139_24

[14] Chau, M., Cheng, R., & Kao, B. (2005). Uncertain Data Mining: A New Research Direction, InProc. *Workshop on the Sciences of the Artificial*, Hualien, Taiwan.

[15] Aggarwal, C., Wolf, J., Yu, P., Procopiuc, C., & Park, J. (1999). Fast algorithms for projected clustering. In *Proceedings of the 1999 ACM SIGMOD international conference on management of data*, Philadelphia: ACM Press, 61-72. https://doi.org/10.1145/304182.304188

[16] Ichino, M. & Yaguchi, H. (1994). Generalized minkowski metrics for mixed feature type data analysis. *IEEE TSMC, 24*(4), 698-708. https://doi.org/10.1109/21.286391

[17] Dehne, F. K. H. A. & Noltemeier, H. (1987). Voronoi trees and clustering problems. *Inf. Syst. 12*(2), 171-175. https://doi.org/10.1016/0306-4379(87)90041-X

[18] Lukić, I., Köhler, M., & Slavek, N. (2012). The Segmentation of Data Set Area Method in the Clustering of Uncertain Data. *Proceedings of 36th International Convention on Information and Communication Technology, Electronics and Microelectronics MIPRO*, Opatija, Croatia, May 2012, 420-425.

[19] Bowring, B. R. (1976). Transformation from spatial to geographical coordinates. *Survey Review XXIII*, 181. https://doi.org/10.1179/sre.1976.23.181.323

[20] Lukić, I., Köhler, M., & Slavek, N. (2014). Positioning of Public Service Systems Using Uncertain Data Clustering. *Acta Polytechnica Hungarica, 11*(1), 121-133.

Contact information:

doc. dr. sc. **Ivica LUKIĆ,** dipl. ing.
Sveučilište J. J. Strossmayera u Osijeku,
Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek,
Kneza Trpimira 2b, 31000 Osijek, Croatia
ivica.lukic@ferit.hr

doc. dr. sc. **Mirko KOHLER,** dipl. ing.
Sveučilište J. J. Strossmayera u Osijeku,
Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek,
Kneza Trpimira 2b, 31000 Osijek, Croatia
mirko.kohler@ferit.hr

dr. sc. **Tomislav GALBA,** mag. ing.
Sveučilište J. J. Strossmayera u Osijeku,
Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek,
Kneza Trpimira 2b, 31000 Osijek, Croatia
tomislav.galba@ferit.hr