

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/305380696>

# A Genetic Algorithm for the p-Median Facility Location Problem

Conference Paper · July 2016

CITATIONS

2

READS

1,198

4 authors:



[Sule Itir Satoglu](#)

Istanbul Technical University

64 PUBLICATIONS 319 CITATIONS

[SEE PROFILE](#)



[Mehmet Kursat Oksuz](#)

Istanbul Technical University

17 PUBLICATIONS 29 CITATIONS

[SEE PROFILE](#)



[Gulgun Kayakutlu](#)

Istanbul Technical University

98 PUBLICATIONS 788 CITATIONS

[SEE PROFILE](#)



[Kadir Buyukozkan](#)

Karadeniz Technical University

11 PUBLICATIONS 48 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Intelligence in Energy [View project](#)



Design of the Hybrid Manufacturing Systems in Erratic Demand Conditions [View project](#)

# A Genetic Algorithm for the P-Median Facility Location Problem

Mehmet Kursat Oksuz

Sule Itir Satoglu

Gulgun Kayakutlu

## Abstract

The p-median problem is one of the most well-known facility location problem and have several applications in transportation, distribution, location of public, warehouses etc. The objective is to locate p facilities (medians) such that the sum of the distances from each demand point to its nearest facility is minimized. The p-median problem is well known to be NP-hard and several heuristics have been developed in the literature, but there are few applications of genetic algorithms for this problem. In this study, a new genetic algorithm approach to solve uncapacitated p-median problem is proposed. The parameters of the genetic algorithm are tuned using design of experiments approach. The proposed algorithm is tested on several instances of benchmark data set and evaluated with optimal solutions of the problems.

**Keywords:** P-median problem, facility location, genetic algorithm, heuristics

## 1. Introduction

P-median problem is a well-known discrete optimization problem aiming to locate p number of facilities that satisfies the demand of multiple places with minimum cost. Besides, the p-median problem is a network problem that was originally designed for, and has been extensively applied to facility location. The search for p-median nodes on a network is a classical location problem. In the supply chain context, distribution of goods from decentralized warehouses is more beneficial than that from a central warehouse (Satoglu et al., 2006). Besides, p-median problem has been studied for solving the cell formation problem (Behret and Satoglu, 2012).

The p-median problem is an NP-hard combinatorial optimization problem, because of this reason, if the problem size is increase, it is getting harder to obtain optimum solution via the mathematical models. There is a large number of studies on the p-median problem in the literature. Reese (2006) reviewed the past studies according to the problem type and the solution methods employed. Later, Mladenovic et al. (2007) assessed the meta-heuristic studies that intended to solve the p-median problem. Summary of the p-median studies are summarized in the literature review section of this study.

The aim of this study is to develop a new Genetic Algorithm (GA) to solve the p-median problem that can reach optimal or near optimal solutions. The unique aspect of the study is that first an Initial Solution Algorithm is employed to reach good beginning solutions. Thus, the algorithm can reach solutions equal to or very close to the optimum. Moreover, a  $3^5$  Full Factorial Design is performed where three levels are selected for the factors of the probability of mutation, the population size and the number of iterations, and parameter tuning is performed to reach a better performance. The objective values and the CPU times are considered as response variables. For each parameter level, the proposed GA is run five times. By using the GA solution results, MANOVA and Post-Hoc Tests are performed, to identify whether the performance difference between selected parameter levels are statistically significant, for each problem. Hence, significant parameter levels are determined and input into the algorithm. The proposed GA is tested on the well-known data set presented in the *OR-Library* which consist of 15 instances with up to 100 medians and 300 demand points. In addition, the results are compared with those of another GA in the literature which is presented by Alp et al. (2003).

The paper is organized as follows: The p-median studies and those that used GA for the p-median are reviewed in the literature review section. Later, the proposed GA that is integrated with the Initial Solution Algorithm is explained in Section 3. In Section 4, parameter tuning and experimental design stages are explained for the selected data set. The computational results are presented and discussed in Section 5. Finally, the conclusion and future research are presented.

## 2. Literature Review

Over the past 10 years, there has been a dramatic increase in the amount of literature on solution methods for the p-median problem. Several heuristic and metaheuristic methodologies were developed to solve the p-median problems by the researches. In the field of the heuristics, Rolland et al. (1996) proposed a tabu search algorithm for the p-median problem. In the algorithm, long term and short term memory, strategic oscillation and random tabu list sizes were used. Results of the algorithm were compared with two other heuristics to show its performance. Beltran et al. (2006) proposed a Semi-Lagrangian relaxation approach to the p-median problem. It was tested by solving large-scale instances and the best known dual bounds for five of the six non solved difficult problems were improved. Avella et al. (2012) developed an aggregation heuristic for the large size p-median problems. The authors introduced a new heuristic for large-scale p-median problem instances based on Lagrangian relaxation. Dzator and Dzator (2013) proposed a new heuristic for the medium size p-median problems and applied to an ambulance location problem. A reduction and an exchange procedure are used in the heuristic and 400 randomly generated problems and 6 well known test problems are used to test the proposed methodology. Sevkli et al. (2014) developed a new discrete particle swarm optimization (PSO) algorithm for the p-median problem. The PSO algorithm was tested on benchmarking problem instances from *OR-Library* and its performance was compared with other algorithms in the literature such as neural model, reduced variable neighbourhood search, simulated annealing and other existing discrete PSO algorithms in the literature.

In the field of the metaheuristics, Chiyoshi and Galvão (2000) presented a statistical analysis of simulated annealing for the p-median problem. Elements of the vertex substitution method of Teitz and Bart combined with the general methodology of simulated annealing. The cooling schedule adopted includes the notion of temperature adjustments rather than just temperature reductions. Computational results were given for test problems ranging from 100 to 900 vertices, retrieved from *OR-Library*. Optimal solutions were found for 26 of the 40 problems and high optimum hitting rates were obtained for only 20 of them. Besides, Resende and Werneck (2004) presented a multistart hybrid heuristic that combines elements of several traditional metaheuristics to find near-optimal solutions to p-median problem. The robustness of the algorithm is demonstrated in the experimental study and better result in terms of both running time and solution quality is obtained. Senne et al. (2005) proposed a branch-and-price algorithm to solve the large scale p-median problems. The traditional column generation process was compared with a stabilized approach that combines the column generation and Lagrangian/surrogate relaxation. The combined use of Lagrangian/surrogate relaxation and subgradient optimization in a primal-dual viewpoint was found to be a good solution approach.

In more recent studies, Al-Khedhairi (2008) proposed a simulated annealing metaheuristic to find optimal or near optimal solution for the p-median problem. The proposed metaheuristic was tested on 40 well-known problems in *OR-Library* and results were reported. Berman and Drezner (2008) proposed an integer programming model and a heuristic for the p-median problem under uncertainty. The p-median problem under uncertainty is to find the location of p facilities such that the expected value of the objective function in the future is minimized. The problem was formulated on a graph and an integer programming formulation was constructed, also heuristic algorithms were suggested for its solution. Lim and Ma (2013) proposed a GPU-based parallel vertex substitution (PVS) algorithm for the p-median problem using the CUDA architecture by NVIDIA. PVS is developed based on the best profit search algorithm that shown to produce reliable solutions for the p-median problems. In this approach, each candidate solution in the entire search space is allocated to a separate thread, rather than dividing the search space into parallel subsets. Antamoshkin and Kazakovtsev (2013) studied on the p-median location problem on networks and proposed a heuristic algorithm which is based on the probability changing method (a special case of the genetic algorithm) for an approximate solution to the problem. The ideas of the algorithm are proposed under the assumption that, in the large scale networks with comparatively small edge lengths, the p-median problem has features similar to the Weber problem. The efficiency of the proposed algorithm and its combinations with the known algorithms are proved by the experiments.

The genetic algorithms are slightly used for the p-median problems in the literature. Bozkaya et al. (2002) proposed a genetic algorithm for the p-median problem and the algorithm was tested on randomly generated problems. It was shown that good results can be obtained by using this algorithm. Besides, Alp et al. (2003) proposed a new genetic algorithm that uses a greedy selection heuristic instead of the classical crossover operator. The algorithm was tested on 80 problems in the literature and compared with other heuristics. In addition, Fathali (2006) proposed a genetic algorithm for solving the p-median problem with positive and negative weights. Computational results were compared with those obtained by a variable neighborhood search method and showed that for almost all examples the proposed GA has better performance.

In the latest studies, Basti and Sevkli (2015) proposed an artificial bee colony algorithm which is a recently developed population-based optimization algorithm for the combinatorial problems. The algorithm was tested on several benchmark instances by comparing several metaheuristics in the literature and competitive results were obtained by using the algorithm. Janáček and Kvet (2016) presented a sequential approximate approach for solving the large scale p-median problem instances. It was used for the public service system design problem which is related to the p-median problem and efficiency of the proposed approach was tested on several test problems in the literature.

The summary of the p-median studies are presented in Table 1. For further investigation, Mladenovic et al. (2007) presented a survey of metaheuristic approaches for solving the classical p-median problems. In addition, Reese (2006) summarized the literature on solution methods for the uncapacitated and capacitated p-median problems and presented annotated bibliography of different solution methods.

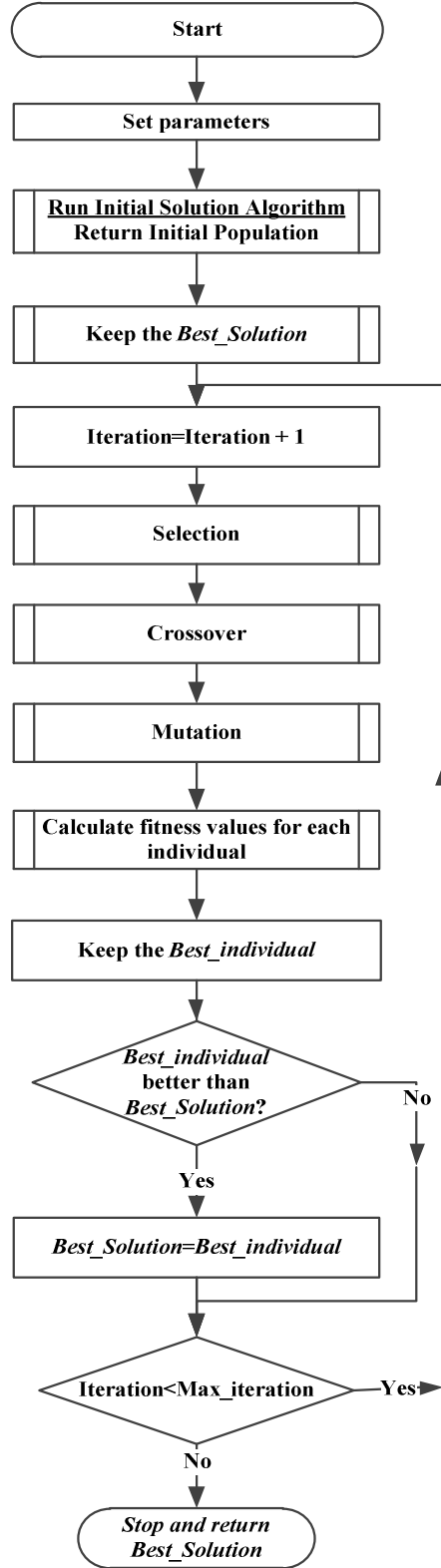
**Table 1.** Summary of the p-median studies.

Study	Method
Rolland et al. (1996)	Tabu Search
Chiyoshi and Galvão (2000)	Simulated Annealing
Bozkaya et al. (2002)	Genetic Algorithm
Alp et al. (2003)	Genetic Algorithm
Resende and Werneck (2004)	Hybrid Metaheuristic
Senne et al. (2005)	Branch and Price Algorithm
Beltran et al. (2006)	Semi-Lagrangian relaxation
Fathali (2006)	Genetic Algorithm
Al-khedhairi (2008)	Simulated Annealing
Berman and Drezner (2008)	Integer programming and a heuristic
Avella et al. (2012)	Lagrangian relaxation based heuristic
Dzator and Dzator (2013)	Heuristic Algorithm
Lim and Ma (2013)	Parallel Vertex Substitution Algorithm
Antamoshkin and Kazakovtsev (2013)	Random Search Algorithm
Sevkli et al. (2014)	Particle Swarm Optimization
Basti and Sevkli (2015)	Artificial bee colony algorithm
Janáček and Kvet (2016)	Sequential approach

### 3. Proposed Genetic Algorithm

Genetic Algorithm is a metaheuristic search method that has been inspired by biological progression. Firstly, it is proposed as a problem-solving method in the 1960s. It has been intensively used as an effective and robust search method for many optimization problems ever since the 1990s. In the algorithm each solution point is represented by a chromosome. The chromosome structure will vary depending on the problem considered.

For the p-median problem, solution is obtained by assigning all demand points to the selected medians. Therefore, the chromosomes will be an array with elements as the number of median. If we have a four median problem, 14-21-10-8 could be a sample chromosome structure. Basic procedure of the proposed GA could be shown as Figure 1.



**Figure 1.** Basic procedure of the proposed GA.

The algorithm starts with parameters set. Thus, population size ( $p\_size$ ), mutation probability ( $mp$ ), and maximum iteration ( $max\_iter$ ) values are determined. Later, the initial solution algorithm is run for generate high quality initial population. These procedures are described briefly as following.

### Initial Solution Algorithm

The initial solution algorithm is adapted from Mulvey and Beck (1984). The pseudo code for the algorithm is shown below.

#### Begin

*For each individual*

*Randomly select p-medians*

*Assign all demand points to nearest median*

*For each median*

*Determine the center point which has minimum distance to all demand points that assigned this median*

*Replace the median with center point*

*Calculate fitness value for the individual*

*End*

*End*

*Return the population*

**end**

### Fitness value calculation

Fitness value is calculated for an individual by using equation 1. In order to perform calculation, it is necessary to determine the  $x_{ij}$  values by assigning all demand points to its nearest median.

$$\min z = \sum_i^n \sum_j^p d_{ij} x_{ij} \quad (1)$$

$d_{ij}$ : distance between demand point  $i$  and candidate median  $j$ .

$x_{ij}$ :  $\begin{cases} 1, & \text{if demand point } i \text{ is served by median } j \\ 0, & \text{otherwise} \end{cases}$

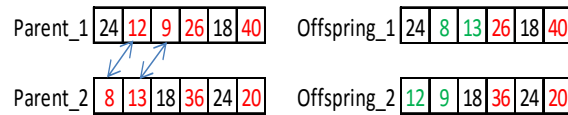
### Selection

Ranking-based selection method is used for selection operator which is adapted from Correa et al. (2004). The basic idea of this method is select high quality solution more than the low quality solution. The equation 2 is used to obtain a sequence number corresponding to the random number generated. In this equation,  $R$  is the list of individuals which is ranked in an ascending order according to the fitness value.  $L$  represents the number of individuals.  $Rnd$  is a random number generated between 0 and 1.  $[b]$  symbol used in Eq. (2) represents the largest integer which is smaller or equal to  $b$ . The Eq. (2) gives the sequence number ( $j$ ) of the individual which will be selected from the list  $R$ .

$$Select(R) = \left\{ r_j \in R \mid j = L - \left\lfloor \frac{-1 + \sqrt{1 + 4rnd(L^2 + L)}}{2} \right\rfloor \right\} \quad (2)$$

### Crossover

In the basic structure of the genetic algorithm, a random value is generated to decide crossover. Here we perform crossover for all individual pairs but not the same ones. The number of  $k$  genes is replaced for each pair. “ $k$ ” is generated randomly between 1 and non-identical number of genes. An example for crossover is illustrated in Figure 2. In this example, non-identical gene number is four while  $k$  is two.



**Figure 2.** An example for the crossover operation.

### Mutation

The mutation operator is employed by considering the mutation probability “ $mp$ ”. For each individual, a random number is generated. If the generated random number is smaller than the  $mp$  value, the mutation process is

performed for the concerned individual. In this process, a randomly selected median is replaced with a randomly selected demand point.

#### 4. Parameter tuning

The parameters of a heuristic or metaheuristic algorithm may have a great influence on the desired output. Moreover, the time required to the parameter setting of an algorithm sometimes far exceeds the development time (Adenso-Diaz and Laguna, 2006). Despite this fact, parameter tuning is usually neglected in most of the heuristic studies.

In this study, a statistical design of experiments (DOE) was conducted to determine the parameter levels of the proposed GA for chosen data set and thus to obtain better results. A  $3^3$  Full Factorial Design is performed where three levels are selected for the factors of the probability of mutation, the population size and the number of iterations. The objective (fitness) value and the CPU time are considered as response variables. Five runs are conducted for each combination of the factor levels. MANOVA and Post-hoc tests (Duncan and Tukey) are performed by using the GA solution results to identify whether the performance difference between selected parameter levels are statistically significant, for each problem set. Hence, significant parameter levels are determined and input into the algorithm. The selected parameters are presented in the Appendix A.

#### 5. Computational Results

The proposed GA is tested on the well-known data set presented in the *OR-Library* which consist of 15 instances with up to 100 medians and 300 demand points. In addition, the results are compared with those of another GA in the literature which is presented by Alp et al. (2003).

The proposed GA was coded and implemented in Matlab® and the computational tests were made on i7-4500U CPU 2.0 Ghz personal computer. The results of the GA for the 15 test problems are presented in Table 2 and compared with the optimum values reported in the literature. Moreover, the CPU times and gap between the optimum solutions and solutions obtained by the GA are reported.

**Table 2.** Summary of the results for the 15 test problems of the OR-Library.

Problem	N	p	Optimum	GA		ADE		Best dev. (%)	
				Obj. val.	Time(s)	Obj. val.	Time(s)	GA	ADE
pmed1	100	5	5819	5819	0,1	5819	0,1	0,000	0,000
pmed2	100	10	4093	4093	0,9	4093	0,1	0,000	0,000
pmed3	100	10	4250	4250	0,2	4250	0,2	0,000	0,000
pmed4	100	20	3034	3034	1,2	3034	0,2	0,000	0,000
pmed5	100	33	1355	1355	3,2	1355	0,3	0,000	0,000
pmed6	200	5	7824	7824	2,6	7824	0,4	0,000	0,000
pmed7	200	10	5631	5631	3,9	5631	0,5	0,000	0,000
pmed8	200	20	4445	4445	14,2	4445	0,7	0,000	0,000
pmed9	200	40	2734	2734	30,9	2734	1,2	0,000	0,000
pmed10	200	67	1255	1255	39,6	1256	2,0	0,000	0,080
pmed11	300	5	7696	7696	27,4	7696	1,7	0,000	0,000
pmed12	300	10	6634	6634	45,8	6634	1,2	0,000	0,000
pmed13	300	30	4374	4374	75,1	4374	2,1	0,000	0,000
pmed14	300	60	2968	2968	289	2968	4,4	0,000	0,000
pmed15	300	100	1729	1731	329	1733	6,3	0,116	0,230

As shown in Table 2, optimum solutions were found in 14 out of 15 problems by using the proposed GA. Besides, the gap is about 0.12 percent for the 15<sup>th</sup> test problem. According to the CPU times, the algorithm showed promising performance. This result shows efficiency of the proposed GA with respect to both solution quality and the CPU performance. Moreover, the results are compared with those of another GA in the literature

called as “ADE” which is presented by Alp et al. (2003). Computational results showed that our algorithm superior to ADE with respect to the objective values.

## 6. Conclusion

In this study, a new GA is developed for the uncapacitated p-median problem that uses an Initial Solution Algorithm to get better results. Thus, good beginning solutions are obtained and the computational time of the GA is reduced considerably. In addition, a  $3^3$  Full Factorial Design is performed where three levels are selected for the factors of the probability of mutation, the population size and the number of iterations, and parameter tuning is performed to reach a better performance. The objective values and the CPU times are considered as response variables. For each parameter level, the proposed GA was run five times. By using the GA solution results, MANOVA and Post-Hoc Tests are performed to identify whether the performance difference between selected parameter levels are statistically significant, for each problem. Hence, significant parameter levels are determined and input into the algorithm.

The proposed GA is solved for the 15 test problems presented in the *OR-Library*. The results show that the parameter tuning and the proposed Initial Solution Algorithm improved the performance of our GA. Moreover, the result showed that promising solutions can be obtained for larger problems by using this algorithm. However, the algorithm can be expanded for the capacitated p-median problem for the more practical implications.

In the future studies, the performance of the proposed GA for solving larger problems can be also examined. Besides, different heuristics or hybrid meta-heuristics can be employed for solving the uncapacitated p-median problem and compared with the proposed GA. The algorithm can be also implemented for a real case problem to show its efficiency and applicability.

## References

- Adenso-Diaz, B., & Laguna, M. (2006). Fine-tuning of algorithms using fractional experimental designs and local search. *Operations Research*, 54(1), 99-114.
- Al-Khedhairi, A. (2008). Simulated annealing metaheuristic for solving p-median problem. *International Journal of Contemporary Mathematical Sciences*, 3(28), 1357-1365.
- Alp, O., Erkut, E., & Drezner, Z. (2003). An Efficient Genetic Algorithm for the p-median problem. *Annals of Operation Research*, 122, 21-42.
- Antamoshkin, A. N., & Kazakovtsev, L. A. (2013). Random Search Algorithm for the p-median problem. *Informatica*, 37, 267-278.
- Avella, P., Boccia, M., Salerno, S., & Vasilyev, I. (2012). An aggregation heuristic for large scale p-median problem. *Computers and Operation Research*, 39(7), 1625-1632. <http://doi.org/10.1016/j.cor.2011.09.016>.
- Basti, M., & Sevkli, M. (2015). An artificial bee colony algorithm for the p-median facility location problem. *International Journal of Metaheuristics*, 4(1), 91-113.
- Behret, H., & Satoglu, S. I. (2012). Fuzzy Logic Applications in Cellular Manufacturing System Design. *Computational Intelligence Systems in Industrial Engineering*, Atlantis Press, pp. 505-533.
- Beltran, C., Tadonki, C., & Vial, J. P. (2006). Solving the p-median problem with a semi-Lagrangian relaxation. *Computational Optimization and Applications*, 35(2), 239-260.
- Berman, O., & Drezner, Z. (2008). The p-median problem under uncertainty. *European Journal of Operation Research*, 189, 19-30. <http://doi.org/10.1016/j.ejor.2007.05.045>.
- Bozkaya, B., Zhang, J. & Erkut E. (2002). An efficient genetic algorithm for the p-median problem. *Facility location: Applications and theory*, Springer Verlag, Berlin, pp. 179-205.
- Chiyooshi, F. & Galvão, R.D. (2000). A statistical analysis of simulated annealing applied to the p-median problem. *Annals of Operation Research*, 96, 61-74.
- Correa, E. S., Steiner, M. T. A., Freitas, A. A., & Carnieri, C. (2004). A genetic algorithm for solving a capacitated p-median problem. *Numerical Algorithms*, 35(2-4), 373-388.
- Dzator, M., & Dzator, J. (2013). An effective heuristic for the p-median problem with application to ambulance location. *OPSEARCH*, 50(1), 60-74. <http://doi.org/10.1007/s12597-012-0098-x>.
- Fathali, J. (2006). A genetic algorithm for the p-median problem with pos/neg weights. *Applied Mathematics and Computation*, 183(2), 1071-1083.
- García, S., Labbé, M., & Marín, A. (2011). Solving large p-median problems with a radius formulation. *INFORMS Journal on Computing*, 23(4), 546-556.



- Janáček, J., & Kvet, M. (2016). Sequential approximate approach to the p-median problem. *Computers & Industrial Engineering*, 94, 83-92.
- Lim, G. J., & Ma, L. (2013). GPU-based parallel vertex substitution algorithm for the p-median problem. *Computers & Industrial Engineering*, 64(1), 381–388. <http://doi.org/10.1016/j.cie.2012.10.008>.
- Mladenović, N., Brimberg, J., Hansen, P., & Moreno-Pérez, J. A. (2007). The p-median problem: A survey of metaheuristic approaches. *European Journal of Operational Research*, 179(3), 927-939.
- Mulvey, J. M., & Beck, M. P. (1984). Solving capacitated clustering problems. *European Journal of Operational Research*, 18(3), 339-348.
- OR-Library. <http://people.brunel.ac.uk/~mastjjb/jeb/info.html>. Last access date: April 15<sup>th</sup> 2016.
- Reese, J. (2006). Solution methods for the p-median problem: An annotated bibliography. *Networks*, 48(3), 125-142.
- Resende, M. G. C., & Werneck, R. F. (2004). A Hybrid Heuristic for the p-median Problem. *Journal of Heuristics*, 10, 59–88.
- Rolland, E., Schilling D. A. & Current J. R. (1996). An efficient tabu search procedure for the p-median problem. *European Journal of Operation Research*, 96, 329-342.
- Satoglu, S. I., Durmusoglu, M. B., & Dogan, I. (2006). Evaluation of the conversion from central storage to decentralized storages in cellular manufacturing environments using activity-based costing. *International Journal of Production Economics*, 103(2), 616-632.
- Senne, E. L. F., Lorena, L. A. N. & Pereira, M. A. (2005). A branch-and-price approach to p-median location problems. *Computers & Operations Research*, 32(6), 1655–1664. <http://doi.org/10.1016/j.cor.2003.11.024>.
- Sevcli, M., Mamedsaidov R. & Camci F. (2014). A novel discrete particle swarm optimization for p-median problem. *Journal of King Saud University-Engineering Sciences*, 26(1), 11–19. <http://doi.org/10.1016/j.jksues.2012.09.002>.

## Acknowledgements

This study has been financially supported by Turkish National Science Foundation (TUBITAK), with the project number 215M143.

## Appendix A. Parameter values for the test problems according to DOE.

Problem	Parameters		
	P_size	mp	max_iter
pmed1	10	0,6	20
pmed2	20	0,6	50
pmed3	20	0,3	20
pmed4	20	0,6	50
pmed5	20	0,3	150
pmed6	20	0,6	50
pmed7	20	0,3	50
pmed8	30	0,6	100
pmed9	20	0,3	150
pmed10	20	0,3	150
pmed11	20	0,6	150
pmed12	20	0,6	150
pmed13	20	0,6	150
pmed14	20	0,1	250
pmed15	40	0,3	250