*Journal of Industrial and Systems Engineering*
Vol. 12, No. 4, pp. 172 - 197
Autumn (November) 2019

# Multi-objective optimization of population partitioning problem under interval uncertainty

**Foroogh ghollasi mood[1], Hasan Hosseini-Nasab[1*], Mohammad Bagher Fakhrzad[1], javad Tayyebi[2]**

*Industrial Engineering Department, Yazd University, Yazd, Iran*
*Industrial Engineering Department, Birjand University of Technology, Birjand, Iran*

*strong>forooghghollasi@gmail.com, hhn@yazd.ac.ir, mfakhrzad@yazd.ac.ir, javadtayyebi@birjand.ac.ir*

## Abstract

This paper addresses a bi-objective mixed integer optimization model under uncertainty for population partitioning problem. The objective functions are to minimize the number of communications between partitions and to balance their population. The main constraints are defined for creating contiguous and compact partitions as well as assigning uniquely each basic unit to one partition. To deal with the uncertainty of parameters, a robust programming method is proposed that causes the uncertainty parameters lie between the interval of best-case (the deterministic mode) and worst-case (the highest uncertainty level for all parameters). As the suggested method is NP-Hard, three meta-heuristic algorithms NSGAII, PESA, and SPEA are developed and, to evaluate the efficiency of the algorithms, 10 small-size examples, 10 medium-size examples and, 10 large-size examples are generated and solved. According to computational results, the SPEA has the best performance. The method is examined for a real-world application, as a case study in Iran.

**Keywords:** Partitioning, interval uncertainty, multi-objective optimization, robust programming.

## 1-Introduction

Population partitioning problem is generally defined as grouping basic units to partitions (Garfinkel & Nemhauser, 1970). Optimal partitions in a territory should have features, such as balance (population size, distance from each other and unemployment rate), contiguousness, compactness, and the absence of holes (Baqir, 2002). To solve partitioning problems in real-world applications, it is necessary to show connections between territories as a network structure (Liberatore & Camacho-Collados, 2016). This structure is an undirected graph $G = (V, E)$ with vertex set $V$ (cities or basic units) and connection set $E$. Any population partitioning problem can be regarded as a graph partitioning problem (Tran, Dinh, & Gascon, 2017).

Many studies have been done on partitioning application in different fields. One of the earliest studies belongs to Ghiggi et al. (Ghiggi, Puliafito, & Zoppoli, 1975). There, each partition is composed of a certain number of inseparable communities with centralized population. Minciardi et al. studied decomposition of a geographic territory into an indefinite number of non-overlapping partitions.

*Corresponding author
ISSN: 1735-8272, 

They introduced two heuristic algorithms to achieve compact primary areas for reducing calculations process in partitions (Minciardi, Puliafito, & Zoppoli, 1981). Pezzella et al. addressed partitioning problem with optimal allocation of services (Pezzella, Bonanno, & Nicoletti, 1981). Lin and Kao introduced a mixed integer optimization model to partition municipal solid waste collection sites (H.-Y. Lin & Kao, 2008). Chen and Yum presented a new public security criterion to define security function level (Chen & Yum, 2010). Benzarti et al. addressed partitioning problem to deal with home health care services. Their contribution was formulating home health care partitioning problem as a mixed integer programming model with some criteria, such as basic units separation, compactness, workload balance between human resources, and consistency (Benzarti, Sahin, & Dallery, 2013). De Assis et al. investigated balanced multi-criteria partitioning problem for electricity meter reading based on compactness and homogeneity criteria of partitions (De Assis, Franca, & Usberti, 2014). Butsch et al. proposed a heuristic algorithm for arc routing districting problem (Butsch, Kalcsics, & Laporte, 2014). Camacho-Collados et al. studied multi-criteria police districting problem for the first time. They considered some area criteria, such as risk, compactness, and mutual support (Camacho-Collados, Liberatore, & Angulo, 2015). García-Ayala et al. solved road network partitioning problem with a certain number of partitions by an integer mathematical model..

**Table 1.** Studies in the field of graph partitioning with emphasize on solution methods

| **Approach type** | | | **Type of model** | **Application** | **Year** | **Author(s)** |
|---|---|---|---|---|---|---|
| Meta-heuristic | Heuristic | Exact | | | | |
| | ■ | | Single objective | Bus transportation problem | 2012 | **(Shirabe, 2012)** |
| ■ | | | Multi-objective | Health care | 2013 | **(D Datta, Figueira, Gourtani, & Morton, 2013)** |
| | | ■ | Multi-objective | Home health care | 2013 | **(Benzarti et al., 2013)** |
| | | ■ | Single objective | Distribution problem | 2013 | **(Ríos-Mercado & López-Pérez, 2013)** |
| ■ | | | Multi-objective | Electricity meter reading | 2014 | **(De Assis et al., 2014)** |
| | | ■ | Single objective | Urban land use | 2014 | **(Li, Church, & Goodchild, 2014)** |
| ■ | | | Multi-objective | Health care | 2015 | **(Steiner, Datta, Neto, Scarpin, & Figueira, 2015)** |
| ■ | | | Single objective | Health care | 2017 | **(M. Lin, Chin, Fu, & Tsui, 2017)** |
| ■ | | | Single objective | Health care | 2017 | **(Tran et al., 2017)** |
| ■ | | | Single objective | Distribution system | 2018 | **(Kong, Zhu, & Wang, 2018)** |
| | ■ | | Single objective | Rail transportation system | 2018 | **(Zhao, Wang, & Peng, 2018)** |
| ■ | | | **Bi-objective** | **Distribution system** | **-** | **The current study** |

This problem was taken into consideration by many organizations including post offices, municipalities for urban and winter services, road maintenance, and urban waste disposal sections (García-Ayala, González-Velarde, Ríos-Mercado, & Fernández, 2016)

To solve partitioning problems, many methods have been developed mainly based on meta-heuristic algorithms. In addition to the well-known genetic algorithm, some other meta-heuristic algorithms have been used to solve partitioning problem, such as simulated annealing (Brooks & Morgan, 1995), tabu search (Bozkaya, Erkut, & Laporte, 2003), hybrid simulated annealing and tabu-search (Baños, Gil, Paechter, & Ortega, 2007), particle swarm (Wang, Wu, & Mao, 2007), and differential evolution (Dilip Datta & Figueira, 2011).

In this paper, population partitioning problem is proposed as a bi-objective problem with the aim of minimizing connections between basic units outside a partition and balancing the population in partitions under uncertainty conditions. To deal with uncertainty of parameters, the optimization method provided by Sim and Bertsimas (Bertsimas & Sim, 2004) is applied.

The remainder of this paper is organized as follows. The problem definition is given in section 2. Solution methods and the related explanations are presented in section 3. In section 4, the proposed algorithms are compared with the mathematical model. The efficiency of algorithms is assessed in section 5. The case study is investigated in section 6. Finally, conclusions are presented in section 7.

## 2-Problem statement

Assume $G = (V, E)$ is an undirected graph with vertex set $V$ (basic units) and edge set $E$. The vertex set $V$ contains $N$ vertices $v_1, v_2, \ldots, v_N$. Each vertex $v_i$ is represented by a pair $(x_i, y_i)$ of vertical and horizontal coordinates. Any edge of *G* with two endpoints $v_i$ and $v_j$ is denoted by $(v_i, v_j)$. Each vertex $v_i$ has a weight $w_i \geq 0$, and it can be considered as $w_{ij} = w_i + w_j$ for each pair of vertices $v_i$ and $v_j$. These weights represent some characteristics, such as population and demand for each vertex. Since the adjacency matrix corresponding to *G* is symmetric, it follows that$w_{ij} = w_{ji}$. We can also consider $W = (w_{ij})$ as an adjacency matrix. If $w_{ij} > 0$, then there is an edge between vertices $v_i$ and $v_j$. If $w_{ij} = 0$, there is not any edge joining vertices $v_i$ and $v_j$. In addition, another parameter $p_{ij}$ represents the population difference between vertices $v_i$ and $v_j$. In fact, this serves as a parameter to balance the population in partitions.

Suppose that the goal is to divide vertices into the set of partitions $P$. We denote by $|P|$ the number of elements belonging to $P$. So $P = \{1,2, \ldots, |P|\}$. Let the values $C_{min}$ and $C_{max}$ respectively represent the minimum and maximum number of vertices that can be placed in a partition. It is clear that $P \in \{2, \ldots, N-1\}$, $C_{min}, C_{max} \in \{1, \ldots, N\}$ and $C_{min} \leq C_{max}$. Let $X_{ip}$ be a binary variable that equals to one if vertex $v_i$ is assigned to partition $p \in \{1, \ldots, |P|\}$, and otherwise $X_{ip} = 0$. Another binary variable $Y_{ij}$ equals to one if vertices $v_i$ and $v_j$ are not assigned to the same partition, and otherwise $Y_{ij} = 0$. The objective function of partitioning problem is to minimize the total weight of edges that are considered as connections between two partitions. Since the adjacency matrix corresponding to *G* is symmetric, the objective function can be defined as $\min \frac{1}{2} \sum_{i,j} Y_{ij} w_{ij}$ or $\min \sum_{i \leq j} Y_{ij} w_{ij}$. According to these definitions, the mathematical model of the problem is expressed in equations (1) to (7).

Input parameters

$w_{ij}$ Number of required transfers between vertices $v_i$ and $v_j$ (the transferred population between vertices).

$p_{ij}$ Difference between populations of vertices $v_i$ and $v_j$.

$M$ A positive and large enough scalar

### 2-1-Mathematical formulation

***Model 1***

$$Min \sum_{i \in V} \sum_{j=i+1} Y_{ij} w_{ij}, \tag{1.a}$$

$$Min \sum_{i \in V} \sum_{j=i+1} Y_{ij} p_{ij}, \tag{1.b}$$

$s.t.$

$$\sum_{p \in P} X_{ip} = 1, \qquad \forall i \in V, \tag{2}$$

$$C_{min} \leq \sum_{i \in V} X_{ip} \leq C_{max}, \qquad \forall p \in P, \tag{3}$$

$$-Y_{ij} - X_{ip} + X_{jp} \leq 0, \qquad \forall i, j \in V, p \in P, \tag{4}$$

$$-Y_{ij} + X_{ip} - X_{jp} \leq 0, \qquad \forall i, j \in V, p \in P, \tag{5}$$

$$\sum_{r \in H_{ij}} X_{rp} \geq SP_{ij} - M\left(2 - \left(X_{jp} + X_{ip}\right)\right), \qquad \forall i, j \in V, p \in P, \tag{6}$$

$$X_{ip} \in \{0,1\}, Y_{ij} \in \{0,1\}, \qquad \forall i, j \in V, p \in P, \tag{7}$$

The first objective function is to minimize the total weight of connections between partitions. The second objective function balances the population in partitions as much as possible. Constraint (2) ensures that each vertex is assigned to only one partition. Constraint (3) guarantees that the number of vertices in each partition is between the upper and lower bounds. Constraints (4) and (5) state that if vertices $v_i$ and $v_j$ are not in the same partition, then $Y_{ij} = 1$, otherwise $Y_{ij} = 0$. Constraint (6) is the first provided mathematical formulation maintaining continuity, compactness, and the absence of holes. $SP_{ij}$ is the set of vertices on the shortest path between vertices $v_i$ and $v_j$, and $|SP_{ij}|$ is the number of its vertices. Constraint (7) specifies the range of variables.

One of the most prominent characteristics of this model is that it considers the fundamental constraints of the partitioning problem (contiguity and compactness partitions). No integrated, specific mathematical model has been presented yet since the constraints are difficult to design (Kalcsics. J, 2015). The constraint (6) states that the shortest path between each two basic units belonging to a single partition must be located inside that partition. Thus, all of the points on the shortest path between the two basic units are inside that partition as well. This constraint not only assures contiguity and avoidance of unusual partition allocations but also causes partitions with the feature of compactness to be generated. In the final partitioning, therefore, the generated partitions are expected to be convex as far as possible, and no unusual allocation is expected to exist. However, points on shortest path between each two basic units can be specified using some common algorithms like Dijkstra's algorithm. In this algorithm, the shortest-path tree will be formed if the algorithm is run for all the points in the area under investigation. It should be noted that there is not one communication path between each of the two points (corners) in the graph network used in this research, unlike in communication networks between populated areas. The relations between points will be in the form of a graph network. This can be observed more clearly in the case study of the research. However, a question that may be raised is how path selection will work if there is more than one shortest path between two different vertices. To respond to this question, one can consider the structure of the algorithm used for finding the shortest path (Dijkstra's algorithm in this research). As stated, the shortest-path tree will be generated if the algorithm is run for all the vertices in the area. Therefore, two shortest paths are never identified between two specific vertices, since the eventual structure would then involve a cycle, and would no longer be a tree, and this would be a contradiction in the algorithm structure. It can, therefore, be assured that the points on the shortest path between two vertices are identified by the algorithm that is used.

A point to must be clarified is the difference between the parameters $p_{ij}$ and $w_{ij}$ and their computational structure in the mathematical model. Let us illustrate it by one example. Assume that the network contains 5 vertices with populations of 10, 20, 15, 25, and 30 shown in figure 1.

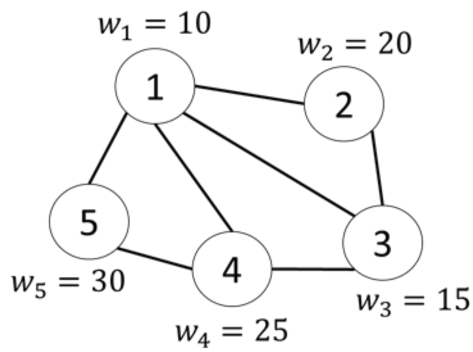



**Fig 1.** A graphic structure of problem

Therefore, one can say that $p_{12} = p_{21} = 10$, $p_{23} = p_{32} = 5$, $p_{34} = p_{43} = 10$, $p_{45} = p_{54} = 5$, $p_{14} = p_{41} = 15$, $p_{13} = p_{31} = 5$, and $p_{15} = p_{51} = 20$. Moreover, parameter $w_{ij}$ is randomly specified according to the following numbers.

$w_{12} = w_{21} = 20$
$w_{23} = w_{32} = 15$
$w_{34} = w_{43} = 15$
$w_{45} = w_{54} = 20$
$w_{14} = w_{41} = 10$
$w_{13} = w_{31} = 15$
$w_{15} = w_{51} = 10$

As stated before, $p_{ij}$ can be calculated through the population of vertices; however, $w_{ij}$ is a parameter that is independently entered by the decision maker. For example, it can indicate the amount of demands that should be moved between adjacent vertices. This work has several practical applications in health care system management and supply chain management. In fact, $p_{ij}$ is used to create population balance in partitions while $w_{ij}$ aims to minimize the number of trips between partitions. For example, in healthcare system, it is assumed that some people from a certain basic unit have to go to adjacent basic units to receive healthcare services. If vertices 1, 2, and 3 are supposed to be in a partition and vertices 4 and 5 in another partition, value of the first and the second objective functions will be 45 and 35 based on objective functions (a.1) and (b.1), respectively. If the partition structure is changed and vertices 1 and 2 are located in one partition and vertices 3, 4, and 5 in the other partition, then the first and the second objective values will be 35 and 50, respectively. As can be seen, different partition structures are non-dominated; therefore, one cannot choose an ideal solution definitely. In fact, this example demonstrates that the provided objective functions are generally contradictory and can be considered as different objective functions.

### 2-2- Robust mathematical formulation

In this paper, the investigated uncertainty is related to matrices $W = (w_{ij})_{N\times N}$ and $P = (p_{ij})_{N\times N}$. Each component $w_{ij}$ is modeled randomly as an indeterminate symmetric distribution parameter bounded to $\hat{w}_{ij}$ which varies in the interval $[w_{ij} - \hat{w}_{ij}, w_{ij} + \hat{w}_{ij}]$. It is noteworthy that parameter $\hat{w}_{ij}$ is the constant part and $w_{ij}$ is the variable part. Therefore, one can state $\widehat{w}_{ij} = \widehat{w}_{ji}$ for $i, j = 1, \dots, N$. Similarly, the same definitions can be proposed for parameter $p_{ij}$. Based on Sim and Bertsimas robust programming (Bertsimas, 2004), the uncertainty conditions for parameter $w_{ij}$ is first described, and then it is considered similarly for parameter $p_{ij}$.

Robust optimization is proposed to investigate uncertainty of weight matrix $W$ by means of $\widetilde{w}_{ij} = [w_{ij} - \widehat{w}_{ij}, w_{ij} + \widehat{w}_{ij}]$, where $w_{ij}$ is the nominal value of edge $(v_i, v_j)$. $J$ is the set of indices related to $W$ with uncertain changes i.e. $J = \{(i,j): \widehat{w}_{ij} > 0, i = 1, .., N, j = i+1, \dots, N\}$. It is supposed that $\Gamma$ is a parameter that is not necessarily integer and gets value in the interval $[0, |J|]$. This parameter was introduced by Bertsimas and Sim (Bertsimas & Sim, 2003) to adjust robustness of the proposed method against conservative level of introduced solution. The number of coefficients $w_{ij}$ and $w_{i_t j_t}$are allowed to be changed at most $\lfloor \Gamma \rfloor$ and $(\Gamma - \lfloor \Gamma \rfloor)$, respectively. The subscript of *i* and *j* is *t* which is explained in the following. In Bertsimas robust programming, the number of uncertain parameters varies proportional to value of the robustness parameter, so there should be a counter in the set of main counters that counts the uncertain parameters. For instance, if we have 10 basic units, there will be 100 number of *w*. Assume that based on value of the robustness parameter, only 10 of them are uncertain, and the rest remain at the upper bound. Hence, a counter is required to count those 10. The index *t* does the same as explained. Therefore, robust partitioning problem can be formulated as follows.

***Model 2***

$$\min_{(X_{ip}, Y_{ij})} \left( \sum_{i \in V} \sum_{j=i+1} Y_{ij} w_{ij} + \max_{\substack{\{S: S \subseteq J, |S| \le \Gamma\} \\ (i_t j_t) \in J/S}} \left( \sum_{(i,j) \in S} Y_{ij} \widehat{w}_{ij} + (\Gamma - \lfloor \Gamma \rfloor)\, \widehat{w}_{i_t j_t} Y_{i_t j_t} \right) \right), \quad (8)$$

$s.t.$

$Constraints\ (2) - (6).$

Notice that there are different conditions based on the selected value $\Gamma$.

- If $\Gamma = 0$, no change is allowed and the problem is decreased to a nominal one like to Model 1.
- If $\Gamma$ is selected as an integer number, the value of the objective function (8) will equal to $\max_{\{S | S \subseteq J, |S| \le \Gamma\}} \sum_{(i,j) \in S} Y_{ij} \widehat{w}_{ij}$ at most.
- If $\Gamma = |J|$, the problem can be solved by the Swister method (Fan, Zheng, & Pardalos, 2012). As stated in (Fan et al., 2012), the objective function (8) can be equivalently formulated as a mixed binary linear programming.

The method used in the proof of the following theorem was proposed by (Bertsimas & Sim, 2004) for the first time.

**Theorem**: Model 2 is equivalent to the following mixed binary linear programming formulation.

***Model 3***

$$Min \sum_{i \in V} \sum_{j=i+1}^{N} Y_{ij} w_{ij} + \Gamma\, U_0 + \sum_{(i,j) \in J} U_{ij}, \quad (9)$$

$s.t.$

$$\sum_{p \in P} X_{ip} = 1, \quad \forall i \in V, \quad (10)$$

$$C_{min} \leq \sum_{i \in V} X_{ip} \leq C_{min}, \quad \forall p \in P, \quad (11)$$

$$-Y_{ij} - X_{ip} + X_{jp} \leq 0, \quad \forall i,j \in V, p \in P, \quad (12)$$

$$-Y_{ij} + X_{ip} - X_{jp} \leq 0, \quad \forall i,j \in V, p \in P, \quad (13)$$

$$U_0 + U_{ij} - Y_{ij}\widehat{w}_{ij} \geq 0, \quad \forall (i,j) \in J, \quad (14)$$

$$\sum_{r \in H_{ij}} X_{rp} \geq SP_{ij} - M\left(2 - \left(X_{jp} + X_{ip}\right)\right), \quad \forall i,j \in V, p \in P, \quad (15)$$

$$U_{ij} \geq 0, \quad \forall (i,j) \in J, \quad (16)$$

$$U_0 \geq 0, \quad (17)$$

$$X_{ip} \in \{0,1\}, Y_{ij} \in \{0,1\}, \quad \forall i,j \in V, p \in P. \quad (18)$$

**Proof**: For any given value of $\left(Y_{ij}\right)_{i=1,\dots,N\ ,j=i+1,\dots,N}$ in Model 2, $\max_{\left\{\substack{S:S\subseteq J,|S|\leq \Gamma \\ (i_t j_t)\in J/S}\right\}} \left(\sum_{(i,j)\in S} Y_{ij}\widehat{w}_{ij} + (\Gamma - \lfloor \Gamma \rfloor)\,\widehat{w}_{i_t j_t} Y_{i_t j_t}\right)$ can be linearized by introducing $z_{ij}$ for all $(i,j) \in J$ subject to constraints $\sum_{(i,j)\in J} z_{ij} \leq \Gamma$ and $0 \leq z_{ij} \leq 1$, as shown in Model 4.

***Model 4***

$$min \sum_{(i,j)\in S} Y_{ij}\widehat{w}_{ij}\, z_{ij}, \quad (19)$$

$s.t.$

$$\sum_{(i,j)\in J} z_{ij} \leq \Gamma, \quad (20)$$

$$0 \leq z_{ij} \leq 1, \quad \forall (i,j) \in J. \quad (21)$$

This formulation is a fractional knapsack problem with bound constraints. The optimal solution of this formulation should have $\lfloor \Gamma \rfloor$ variables $z_{ij} = 1$ and one $z_{ij} = \Gamma - \lfloor \Gamma \rfloor$ that is equivalent to the optimal solution in maximization part of Model 2. Model 4 is linear for the given values of $\left(Y_{ij}\right)_{i=1,\dots,N\ ,j=i+1,\dots,N}$. Its duality can be formulated as follows:

***Model 5***

$$Min\ \Gamma\ U_0 + \sum_{(i,j)\in J} U_{ij}, \quad (22)$$

$s.t.$

$$U_0 + U_{ij} - Y_{ij}\widehat{w}_{ij} \geq 0, \quad \forall (i,j) \in J, \quad (23)$$

$$U_{ij} \geq 0, \quad \forall (i,j) \in J. \quad (24)$$

$$U_0 \geq 0, \quad (25)$$

Model 3 can be obtained by combining models 5 and 2. This completes the proof.

On account of the fact that the proposed model has a linear structure, it can be solved as a mixed binary programming model by CPLEX solver. The final structure of the model is as follows:

***Model 6***

$$Min\ \sum_{i\in V}\sum_{j=i+1} Y_{ij}w_{ij} + \Gamma 1\ U1_0 + \sum_{(i,j)\in J} U1_{ij}, \tag{26}$$

$$Min\ \sum_{i\in V}\sum_{j=i+1} Y_{ij}p_{ij} + \Gamma 2\ U2_0 + \sum_{(i,j)\in J} U2_{ij}, \tag{27}$$

$s.t$

$$\sum_{p\in P} X_{ip} = 1, \quad \forall i \in V, \tag{28}$$

$$C_{min} \le \sum_{i\in V} X_{ip} \le C_{max}, \quad \forall p \in P, \tag{29}$$

$$-Y_{ij} - X_{ip} + X_{jp} \le 0, \quad \forall i,j \in V, p \in P, \tag{30}$$

$$-Y_{ij} + X_{ip} - X_{jp} \le 0, \quad \forall i,j \in V, p \in P, \tag{31}$$

$$\sum_{r\in H_{ij}} X_{rp} \ge SP_{ij} - M\left(2 - \left(X_{jp} + X_{ip}\right)\right), \quad \forall i,j \in V, p \in P, \tag{32}$$

$$U1_0 + U1_{ij} - Y_{ij}\widehat{w}_{ij} \ge 0, \quad \forall (i,j) \in J, \tag{33}$$

$$U1_{ij} \ge 0, \quad \forall (i,j) \in J, \tag{24}$$

$$U1_0 \ge 0, \tag{34}$$

$$U2_0 + U2_{ij} - Y_{ij}\widehat{P}_{ij} \ge 0, \quad \forall (i,j) \in J, \tag{35}$$

$$U2_{ij} \ge 0, \quad \forall (i,j) \in J, \tag{36}$$

$$U2_0 \ge 0, \tag{37}$$

$$X_{ip} \in \{0,1\}, Y_{ij} \in \{0,1\}, \quad \forall i,j \in V, p \in P. \tag{38}$$

## 3-Solution method

In this paper, small-size instances are solved by the epsilon-constraint method using CPLEX solver, Version 12.1. On account of the fact that partitioning problem is NP-hard, meta-heuristic algorithms including NSGAII, PESA, and SPEA are developed to solve large-size instances. The key point in using meta-heuristic algorithms is design of solution representation. Therefore, algebraic structure of the solution representation and the proposed algorithms are presented in the next section.

### 3-1-Solution representation

The solution representation for the present partitioning problem is an array of basic units. The value of a member of the solution is equal to the number of partitions that the basic unit belongs to it. An example of solution structure is shown in Figure 2.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 4 | 2 | 2 | 3 | 2 | 4 | 3 |

**Fig 2.** An example of solution structure

Each number represents a partition, and position of each number represents a basic unit. The length of each solution indicates the number of basic units. For example, the above solution shows a four-partition vector with the following basic units in each partition.

Partition 1: Basic units 1 and 4.
Partition 2: Basic units 3 and 6.
Partition 3: Basic units 5 and 8.
Partition 4: Basic units 2 and 7.

Since some meta-heuristic algorithms require continuous representation, the above solution representation can be converted to continuous form. Firstly, a real random number is generated in $[0, |P| - 1]$. Each generated number is rounded to an integer number greater or equal to itself. For instance, the solution representation related to $|P| = 5$ is shown in figure 3.

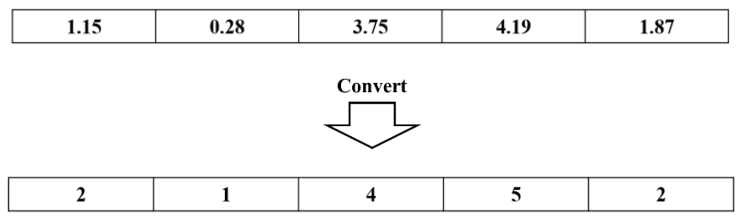



**Fig 3.** An example of solution representation

Since it is hard to determine a feasible solution for the partitioning problem by random assignment, a greedy algorithm is applied to initialize solutions for algorithms. If partition contiguity constraints are not satisfied by the cross-over operator, it will be corrected by a labeling procedure in a way that an unconnected component of a partition would be labeled as a new partition. Other constraints may not be satisfied at any step of initialization and solution production processes. Therefore, a constructive/repair mechanism is applied to those constraints according to the algorithm presented in (Steiner et al., 2015).

## 3-2-Structure of NSGAII algorithm

The multi-objective genetic algorithm is one of most widely used and powerful algorithms to solve multi-objective optimization problems and has been proven to be effective in solving various problems. Deb et al. developed the second version of bi-objective genetic algorithm (Deb, Agrawal, Pratap, & Meyarivan, 2000). They studied both quality and variety of Pareto optimal solutions to eliminate the defects of the first version. In this algorithm, two main criteria of quality and order of solutions are followed. Qualified solutions are first selected; if two identical solutions exist, the solution having more order will be selected. The NSGA-II algorithm has two known phases. The first phase uses the ranking criterion and the concept of domination. The second phase which is related to solutions order uses the congestion distance. In the first phase, solutions are ranked, and two values are calculated for each solution: the number of times that a solution is dominated and the number of solutions dominated by the current solution. To determine the two values, all solutions must be compared to each other. If there are solutions with zero number of dominations, these solutions are non-dominated, and they are Pareto optimal.

*Initialize Population*

*Generate N feasible solution and insert them into Population*

***While** Stopping criteria is not met **Do***

*Generate ChildPopulation of Size N*

*Select Parents from Population*

*Create Children from Parents*

*Mutate Children*

*Repair Solution using repair mechanism*

*Merge Population and ChildPopulation with size 2N*

***For** each individual in CurrentPopulation **Do***

*Assign rank based on Pareto-Fast non-dominates sort*

***end***

*Generate sets of non-dominated vector along* $PF_{known}$

***Loop** (inside) by adding solution to next generation of Population starting from the best front*

***Until** N solution found and determine crowding distance between points on each front*

*end*

*Report results*

**Fig 4.** Multi-objective genetic algorithm pseudo code

Figure 5 shows the flowchart of the NSGAII that is extended in this paper.

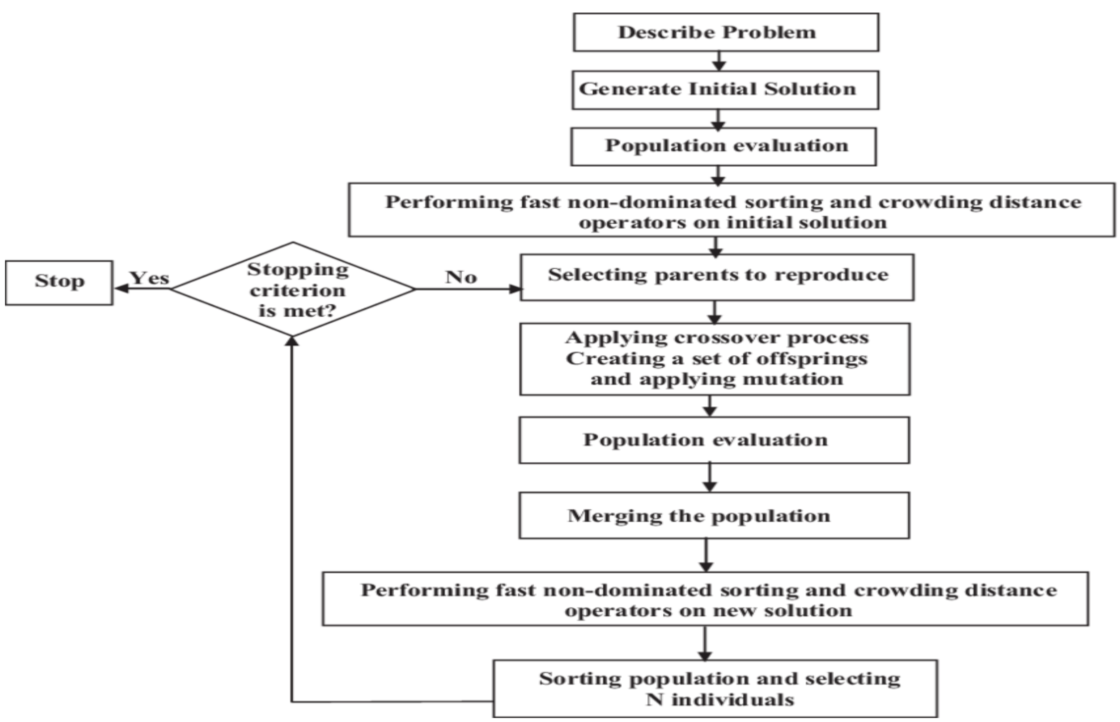



**Fig 5.** Flowchart of the NSGAII

## 3-3- SPEA-II algorithm

SPEA and SPEA-II are efficient algorithms that use an external archive to store non-dominated solutions found during the search algorithm. SPEA algorithm has some weaknesses in the calculation of strength and fitness, and there was not any secondary criterion for comparison of non-dominated solutions. Hence, Zitzler et al. presented the second version of the algorithm that solved the mentioned weaknesses. The framework of SPEA-II algorithm is described below (Zitzler, Laumanns, & Thiele, 2001).

| | |
|---|---|
| $N_E$ | Maximum archive size of non-dominated solutions E |
| $N_F$ | Population size |
| $K$ | Density computation parameter. $\left(K = \sqrt{N_E + N_F}\right)$ |

Step 1: Create initial solutions population $P_0$ and let $E_0 = \emptyset$ and $t = 0$.

Step 2: Calculate fitness of each solution $i$ in $P_t \cup E_t$ as follows.

Sub-step 2-1: Firstly, calculate raw fitness of solution *i* as follows.

$$R(i) = \sum_{j \in P_t} s(j)\,, \qquad \forall\, j > i \in P_t, \tag{39}$$

Where $j > i$ means that solution *j* dominates solution $i$. Moreover, *s(j)* shows strength value of solution, which is the number of solutions that are dominated by solution *j*.

Sub-step 2-2: Calculate fitness of solution *i* as follows.

$$D(i) = \frac{1}{\sigma_i^k + 2}, \qquad \forall\, i \in P_t, \tag{40}$$

Where $\sigma_i^k$ is the distance between solution *i* and the kth nearest neighbor to it.

Sub-step 2-3: Obtain the fitness value by the sum of the raw fitness and the density of solution *i*, i.e.,

$$F(i) = R(i) + D(i), \qquad \forall\, i \in P_t. \tag{41}$$

Step 3: Copy all non-dominated solutions in $P_t \cup E_t$ to $E_{t+1}$. Two possible states may occur.

State 1: If $|E_{t+1}| > N_E$, $|E_{t+1}| - N_E$ number of solutions are eliminated by the repetitive method of deleting the response with the criterion $\sigma^k$. In fact, the solution that has the minimum distance of $\sigma^k$ from other solutions is first eliminated. However, if more than one solution has the minimum distance, the second lowest distance can be determined and thus the additional solutions will be deleted similarly (this criterion will cause to delete similar or closely related solutions that do not care about the solutions density).

State 2: If $|E_{t+1}| \le N_E$ , $N_E - |E_{t+1}|$ number of dominated solutions are moved from $P_t \cup E_t$ to $E_{t+1}$ in order of their fitness value.

Step 4: If the stop condition is provided, the algorithm will stop, and it will return $|E_{t+1}|$ solutions.

Step 5: Use the Dual Competition Method to choose parents from set $E_{t+1}$.

Step 6: Apply cross-over and mutation operators on parents and produce $N_P$ children. The children are added to $P_{t+1}$, and one unit will added to the counter $(t = t + 1)$. Then return to step 2.

It should be noted that this algorithm also uses the same method of cross-over and mutation used in the NSGAII algorithm.

Figure 6 shows the flowchart of the SPEAII that is extended in this paper.

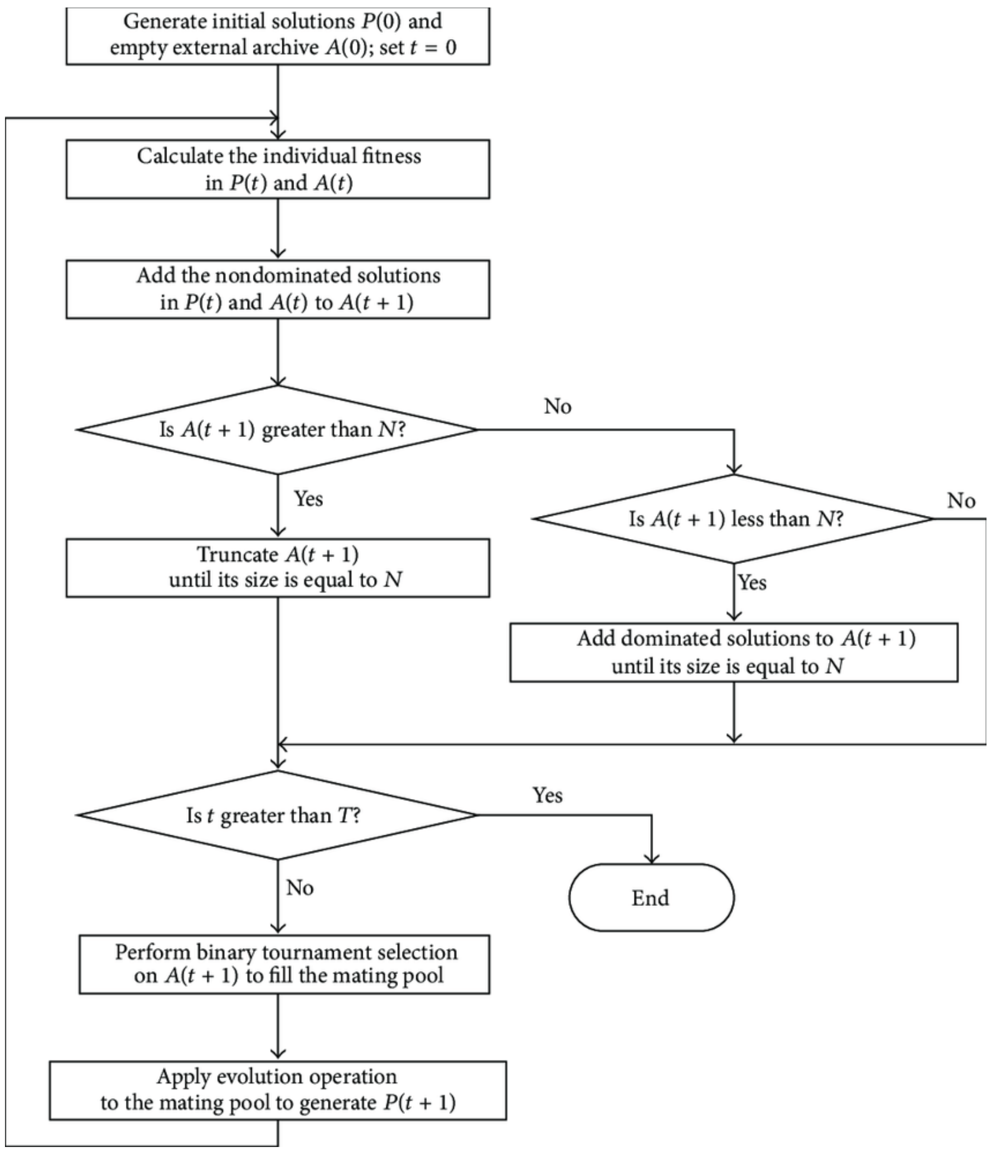



**Fig 6.** Flowchart of the SPEAII

### 3-4-PESA-II algorithm

Another of most well-known multi-objective algorithms is the second version of the Pareto-based selection algorithm (PESA-II) which uses genetic algorithm functions to generate new solutions. The first version of this algorithm, presented by (Corne, Jerram, Knowles, & Oates, 2001), had some weaknesses in the selection phase. The developed version of the algorithm, PESA-II, was presented in 2001. Steps of PESA-II algorithm are as follows.

$N_E$ The largest archive of undesirable solutions E.

$N_P$ Population size.

$N$ Number of networks in each axis of the objective function.

Step 1: Start with a random initial population $P_0$, set the external archive $E_0$ to null, and let $t = 0$.

Step 2: Divide the space into $n^k$ cloud cubicles where $n$ is the number of networks in each axis of the objective function, and $k$ is the number of objectives.

Step 3: Combine non-dominated solutions archive $E_t$ with new solutions of $P_t$. Three possible states may occur.

State 1: If a new solution is dominated by at least one of the solutions in archive $E_t$, delete the new solution.

State 2: If a new solution dominates several solutions in $E_t$, delete the dominated solutions from the archive, add the new solution to archive $E_t$, and update the cloud cube members.

State 3: If a new solution is not dominated by any solution in $E_t$ and does not dominate any solution in $E_t$, add the solution to $E_t$. If $|E_t| = N_{E+1}$, choose an arbitrary cube randomly (the selection is done using the roulette wheel so that the busy arbitrary cube would be more likely to be selected). Select an available solution randomly and delete it. Finally, update the arbitrary cube members.

Step 4: If the stop criterion is met, stop and show the final $E_t$.

Step 5: Let $P_t = \emptyset$, combine some solutions of $E_t$, and select the mutation based on the information density of the arbitrary cubes. Use the cross-over and mutation to generate $N_P$ children and copy it to $P_{t+1}$.

Step 6: Set $t$ to $t + 1$ and go to step 3.

It should be noted that this algorithm also uses the same method of cross-over and mutation used in the NSGAII algorithm.

Figure 7 shows the flowchart of the SPEAII that is extended in this paper.

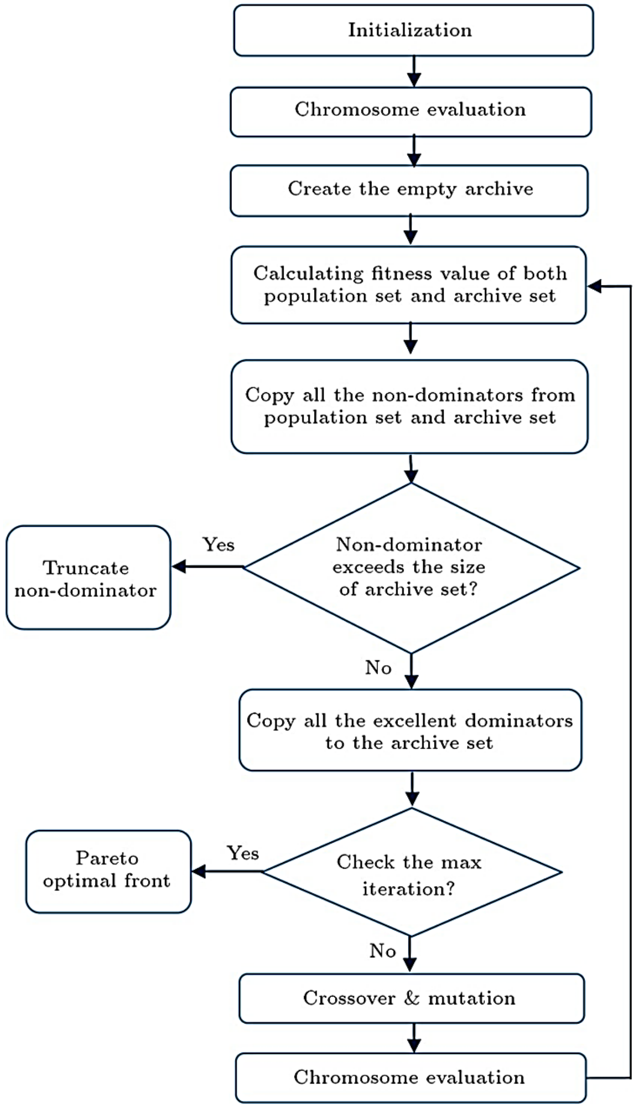



**Fig 7.** Flowchart of the PESAII

## 4-Comparison of proposed algorithm with mathematical model

After solving the problem by the algorithms coded in Visual C ++ using the Visual Studio system with a 3.2GHz processor, the random-access memory of 4GB in Windows 10 operating system, the obtained results are presented. In this section, to examine the results of the proposed algorithms in comparison with the results of the mathematical model, a number of numerical examples are generated, and the efficiency of the proposed algorithms is evaluated using the MID index. In this index, the Euclidean distance between the final non-dominated solutions generated by the algorithm and the optimal Pareto set produced by CPLEX is calculated as follows.

$$MID = \frac{\sum_{i=1}^{|Q|} \left( \sqrt{\sum_{j=1}^{n_{obj}} \left( \frac{f_i^j - f_{best}^j}{f_{\max}^j - f_{\min}^j} \right)^2} \right)}{|Q|}, \tag{42}$$

Where $f_i^j$ represents the *jth* objective value of the *ith* solution. In addition, $f_{best}^j$ is the ideal point of the *jth* objective function. $f_{max}^j$ and $f_{min}^j$ are respectively the highest and the lowest values of all Pareto solutions for the *jth* objective function. $|Q|$ and $n_{obj}$ are respectively the number of points in the Pareto optimal front and the number of objective functions. Since the optimal front is different from that of the algorithm, there is no regularity. Therefore, in this formulation, each member of the algorithm front is calculated with that of the optimal front.

**Table 2.** Comparison of results of the mathematical model and the proposed algorithm

| Instances Size | Instances Number | $\Gamma 1 = \frac{\lvert J1 \rvert}{2}$ and $\Gamma 2 = \frac{\lvert J2 \rvert}{2}$ | | | | | $\Gamma 1 = \frac{\lvert J1 \rvert}{3}$ and $\Gamma 2 = \frac{\lvert J1 \rvert}{3}$ | | | | | $\Gamma 1 = \frac{\lvert J1 \rvert}{5}$ and $\Gamma 2 = \frac{\lvert J1 \rvert}{5}$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Run time (second) | | | | $MID = \max(NSGAII, SPEA, PESA)$ | Run time (second) | | | | $MID = \max(NSGAII, SPEA, PESA)$ | Run time (second) | | | | $MID = \max(NSGAII, SPEA, PESA)$ |
| | | Cplex | NSGAII | SPEA | PESA | | Cplex | NSGAII | SPEA | PESA | | Cplex | NSGAII | SPEA | PESA | |
| Small | SM1 | 153 | 50 | 31 | 50 | 3.41 | 162 | 25 | 48 | 41 | 3.76 | 167 | 27 | 26 | 72 | 1.55 |
| | SM2 | 201 | 53 | 37 | 77 | 3.76 | 192 | 46 | 79 | 54 | 4.90 | 323 | 32 | 43 | 73 | 1.71 |
| | SM3 | 302 | 72 | 62 | 84 | 5.74 | 309 | 121 | 113 | 63 | 6.52 | 357 | 69 | 83 | 97 | 2.61 |
| | SM4 | 330 | 78 | 170 | 88 | 8.07 | 326 | 125 | 175 | 70 | 6.64 | 341 | 99 | 98 | 100 | 3.67 |
| | SM5 | 335 | 89 | 186 | 94 | 9.75 | 348 | 133 | 182 | 81 | 7.46 | 343 | 108 | 128 | 151 | 4.43 |
| | SM6 | 337 | 91 | 200 | 167 | 9.81 | 363 | 181 | 197 | 251 | 9.04 | 358 | 148 | 165 | 168 | 4.46 |
| | SM7 | 361 | 117 | 208 | 192 | 12.83 | 368 | 187 | 218 | 280 | 12.08 | 362 | 155 | 168 | 199 | 5.83 |
| | SM8 | 383 | 162 | 230 | 192 | 13.2 | 357 | 200 | 221 | 305 | 13.12 | 311 | 163 | 247 | 202 | 6.00 |
| | SM9 | 394 | 257 | 233 | 217 | 13.29 | 481 | 243 | 228 | 320 | 15.18 | 322 | 165 | 277 | 208 | 6.04 |
| | SM10 | 447 | 297 | 237 | 238 | 18.85 | 484 | 246 | 236 | 329 | 15.40 | 422 | 178 | 296 | 216 | 8.57 |
| Medium | ME1 | 549 | 304 | 248 | 265 | 20.5 | 501 | 331 | 304 | 341 | 17.32 | 427 | 224 | 306 | 241 | 9.32 |
| | ME2 | 582 | 315 | 255 | 331 | 21.93 | 519 | 347 | 350 | 346 | 17.46 | 432 | 358 | 343 | 251 | 9.97 |

According to the above table, increasing the denominators of $\boldsymbol{\Gamma 1}$ and $\boldsymbol{\Gamma 2}$ fractions results in significant changes of running time. Its reason is that the number of constraints is reduced if the level of uncertainty parameters is decreased. Figures 8, 9 and 10 show these changes.

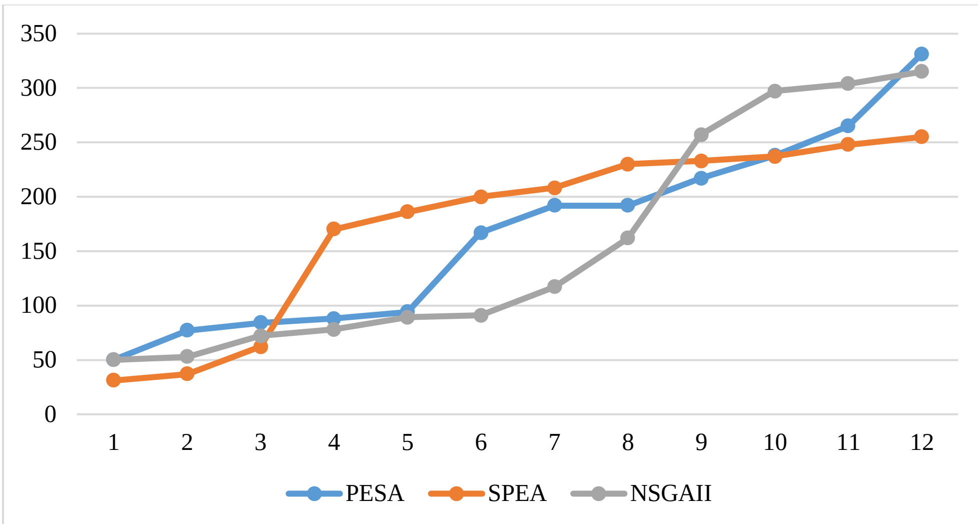



**Fig 8.** The comparison of running time in Γ1=|J1|/2 and Γ2=|J2|/2

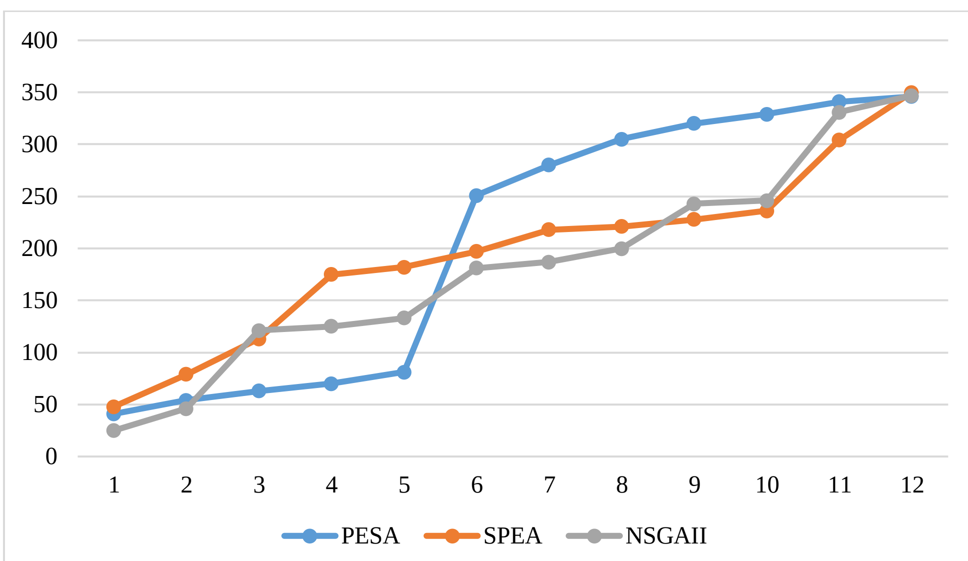



**Fig 9.** The comparison of running time in Γ1=|J1|/3 and Γ2=|J1|/3

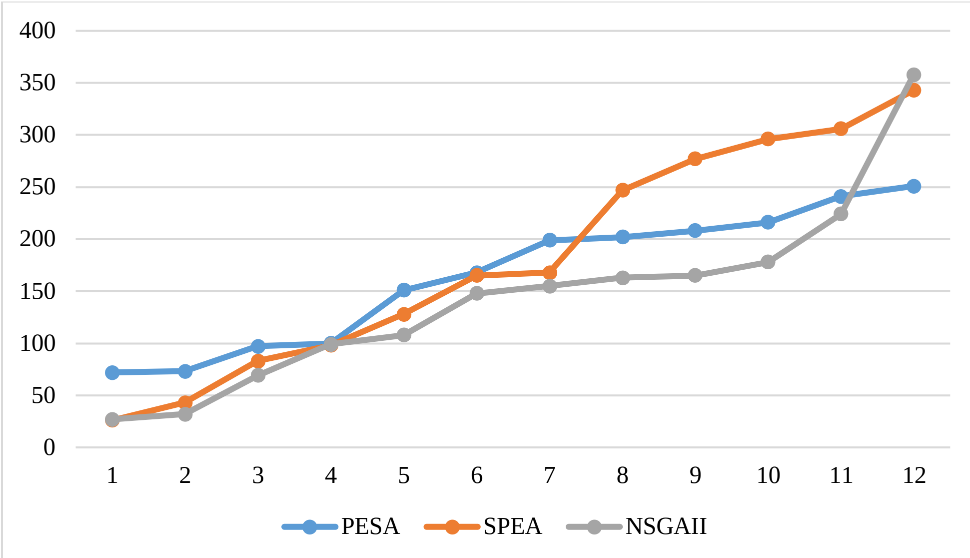



**Fig 10.** The amount of solving time for Γ1=|J1|/5 and Γ2=|J1|/5

As can be seen, the reduction of uncertainty level in the above figures yields that the algorithms' times are closer to each other, and the reason is reducing the solving space and consequently, decreasing the number of computations in different repetitions. Therefore, one cannot rank the algorithms from the aspect of their running time; however, decreasing the amount of uncertainty leads to the reduction of MID value, and consequently the obtained solutions of different instances become closer to the exact solutions produced by CPLEX. Figure 8 displays the trend of these changes.

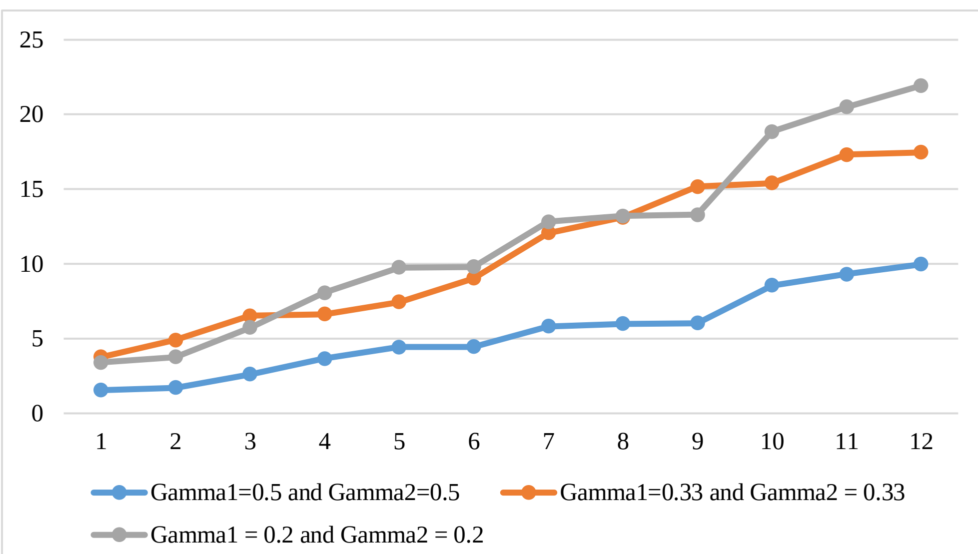



**Fig 11.** The change of MID values for different values of Γ1 and Γ2

According to figure 11, it can be seen that solutions produced with $\Gamma 1 = \frac{|J1|}{5}$ and $\Gamma 2 = \frac{|J1|}{5}$ are closer to the solutions produced by CPLEX which is due to the small number of uncertain parameters, solving space reduction, and decreasing the number of calculations per iteration. In conclusion, one cannot choose an algorithm as the best to solve real-world problems. In other words, obtained results of small-size instances are similar to CPLEX results. To investigate behavior of the proposed algorithms in larger-dimensional problems, a number of numerical examples with different uncertainty level are generated, and the obtained results are examined in the next section.

## 5-Evaluation of the proposed algorithms efficiency

To evaluate the efficiency of the proposed algorithms regarding generation of appropriate solutions, 10 small-size examples, 10 medium-size examples, and 10 large-size examples are generated and compared using the Spread of Non-Domination Solution (SNS) (Maghsoudlou, Kahag, Niaki, & Pourvaziri, 2016) and Maximum Spread (MS) (Samadi, Mehranfar, Fathollahi Fard, & Hajiaghaei-Keshteli, 2018). The SNS and MS indices are calculated as follows:

$$SNS = \sqrt{\frac{\sum_{i=1}^{|Q|}\left(MID - \sum_{j=1}^{n_{obj}} f_i^j\right)^2}{|Q|-1}}, \tag{43}$$

$$MS = \sqrt{\sum_{j=1}^{n_{obj}}\left(f_{\max}^j - f_{\min}^j\right)^2}. \tag{44}$$

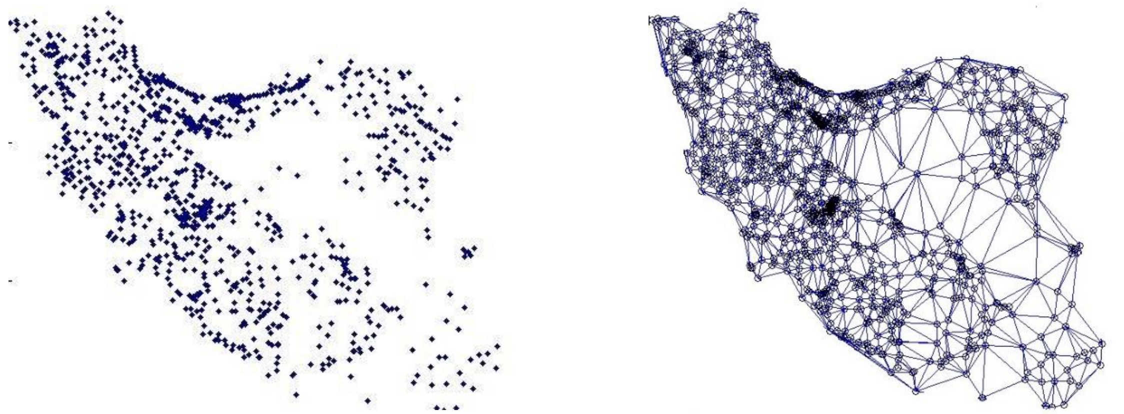
According to table 3, it can be clearly seen that the SPEA algorithm has larger values of SNS and MS criteria which indicates the higher efficiency of this algorithm than the other proposed algorithms; therefore, to solve large-scale problems, it is convenient to use this Algorithm.

**Table 3.** Results of sensitivity analysis of the robustness parameter

| Problem Size | Robust Parameters | | | | Results | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $\lvert J1 \rvert$ | $\Gamma 1$ | $\lvert J2 \rvert$ | $\Gamma 2$ | NSGAII | | PESA | | SPEA | |
| | | | | | SNS | MS | SNS | MS | SNS | MS |
| 1 | 54 | 13 | 60 | 22 | 900646 | 5447 | 1350969 | 7625.8 | 2026454 | 9150.96 |
| 2 | 66 | 19 | 105 | 31 | 684040 | 5823 | 889252 | 7569.9 | 889252 | 9840.87 |
| 3 | 72 | 18 | 111 | 14 | 601712 | 5476 | 782225.6 | 8214 | 938670.7 | 12321 |
| 4 | 92 | 11 | 112 | 31 | 235710 | 6213 | 235710 | 6213 | 329994 | 9319.5 |
| 5 | 130 | 19 | 115 | 18 | 642520 | 7444 | 899528 | 11166 | 1349292 | 16749 |
| 6 | 144 | 21 | 133 | 27 | 84891 | 9459 | 110358.3 | 14188.5 | 110358.3 | 14188.5 |
| 7 | 146 | 17 | 190 | 43 | 993107 | 7292 | 1191728 | 10208.8 | 1787593 | 11229.68 |
| 8 | 156 | 45 | 269 | 67 | 129627 | 9796 | 168515.1 | 13714.4 | 168515.1 | 20571.6 |
| 9 | 186 | 24 | 304 | 42 | 308880 | 7871 | 401544 | 9445.2 | 401544 | 10389.72 |
| 10 | 199 | 59 | 339 | 71 | 367890 | 6124 | 515046 | 8573.6 | 618055.2 | 9430.96 |
| 11 | 233 | 30 | 347 | 79 | 312993 | 5956 | 312993 | 5956 | 375591.6 | 6551.6 |
| 12 | 284 | 28 | 355 | 78 | 290586 | 6970 | 348703.2 | 9061 | 488184.5 | 12685.4 |
| 13 | 292 | 49 | 360 | 93 | 668889 | 6095 | 802666.8 | 6095 | 1043467 | 9142.5 |
| 14 | 339 | 101 | 369 | 70 | 342566 | 5714 | 479592.4 | 6856.8 | 479592.4 | 8228.16 |
| 15 | 339 | 40 | 392 | 101 | 916787 | 9059 | 1100144 | 9059 | 1650217 | 9059 |
| 16 | 385 | 127 | 394 | 114 | 789653 | 6526 | 868618.3 | 9136.4 | 955480.1 | 10050.04 |
| 17 | 391 | 148 | 400 | 108 | 641945 | 9729 | 641945 | 13620.6 | 834528.5 | 16344.72 |
| 18 | 420 | 163 | 413 | 74 | 643529 | 8533 | 900940.6 | 11946.2 | 1171223 | 14335.44 |
| 19 | 435 | 143 | 436 | 87 | 480899 | 9413 | 673258.6 | 10354.3 | 875236.2 | 14496.02 |
| 20 | 466 | 121 | 485 | 155 | 881236 | 9629 | 1145607 | 14443.5 | 1489289 | 15887.85 |
| 21 | 482 | 106 | 489 | 185 | 293192 | 7465 | 439788 | 7465 | 527745.6 | 9704.5 |
| 22 | 487 | 165 | 493 | 69 | 556841 | 7130 | 612525.1 | 7843 | 673777.6 | 11764.5 |
| 23 | 493 | 108 | 494 | 128 | 385585 | 6658 | 385585 | 9987 | 578377.5 | 12983.1 |
| 24 | 502 | 190 | 518 | 67 | 205631 | 5815 | 267320.3 | 7559.5 | 320784.4 | 10583.3 |
| 25 | 521 | 67 | 532 | 133 | 189897 | 6563 | 208886.7 | 8531.9 | 229775.4 | 11944.66 |
| 26 | 532 | 164 | 540 | 210 | 401992 | 5115 | 562788.8 | 6138 | 562788.8 | 6751.8 |
| 27 | 556 | 150 | 546 | 98 | 672824 | 9182 | 1009236 | 11018.4 | 1009236 | 13222.08 |
| 28 | 562 | 89 | 558 | 200 | 581954 | 9197 | 640149.4 | 11956.1 | 704164.3 | 17934.15 |
| 29 | 569 | 85 | 574 | 103 | 13254 | 9717 | 14579.4 | 11660.4 | 21869.1 | 16324.56 |
| 30 | 590 | 171 | 594 | 130 | 869273 | 5944 | 1216982 | 7132.8 | 1582077 | 8559.36 |

## 6-Investigation of case study

In this section, a case study in Iran is investigated. Table 4 and figure 12 show the population and geographical coordinates of the considered basic units in Iran.



***a.Basic units structure*** ***b.Graph structure of basic units.***

**Fig 12.** Graphical structure of the case study

Since connections between basic units are considered as complete graphs in partitioning problem, the basic units in (a) are converted to the structure of the graph (b). The distance between basic units based on the connections and population of each basic unit are available through the data sets of the Iranian Center for Statistics and other legal geographical information sites. Some data used to solve the problem is presented in table 4.

**Table 4.** Some information about population, coordinates, and connections between vertices

| Municipality ID | Centroid Coordinates (x,y) | Number of inhabitants | Connectivity | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | 1 | 2 | 3 | ... | 1021 | 1022 |
| 1 | (47.754331,39.042881) | 1780 | 0 | 1 | 1 | ... | 0 | 0 |
| 2 | (48.53135,37.624719) | 41165 | 1 | 0 | 1 | ... | 0 | 0 |
| 3 | (48.71865,37.388331) | 2841 | 1 | 1 | 0 | ... | 0 | 0 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 1021 | (49.906411,36.888306) | 5417 | 0 | 0 | 0 | ... | 0 | 0 |
| 1022 | (50.174514,37.152006) | 18756 | 0 | 0 | 0 | ... | 0 | 0 |

The following results are the best 10 results of independent implementation of each algorithm. The number of iterations is 200. It is also considered that $\Gamma 1 = \frac{|J1|}{2}$, $\Gamma 2 = \frac{|J2|}{2}$, and $|P| = 10$.

After solving the problem with the SPEA algorithm, which has the best performance, variation interval of the obtained Pareto front are (120, 2506423) and (242, 1031562). Interval of the objective function values are (120, 242) and (2506423, 1031562). The Pareto front structure is considered as figure 13.

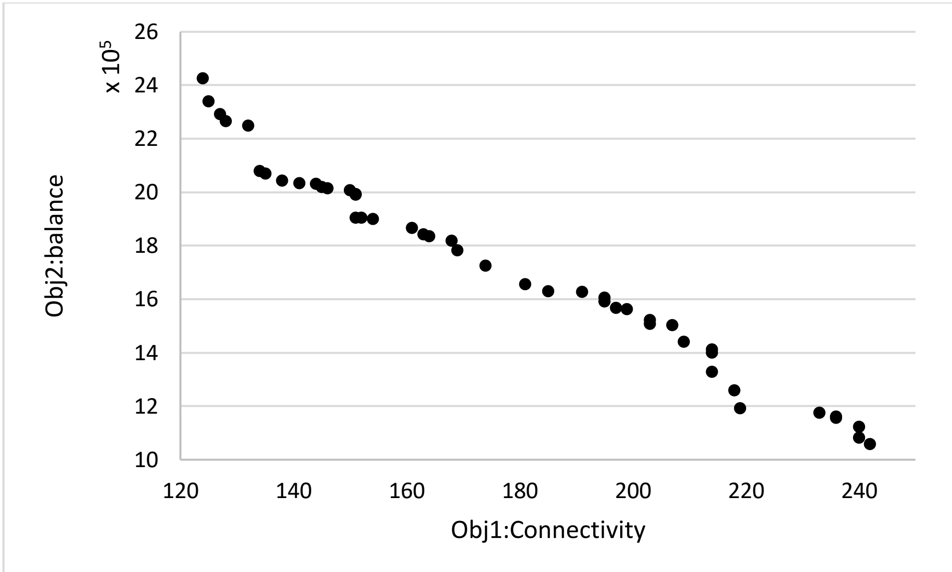



**Fig 13.** The Pareto front obtained from solving the SPEA algorithm for the case study

As can be seen, 46 non-dominated Pareto members are generated. To implement the final results in the case study structure, the Pareto member with the first objective function value of 146 and the second objective function value of 2015013 is considered, and the final partitioning is demonstrated in figure 14.

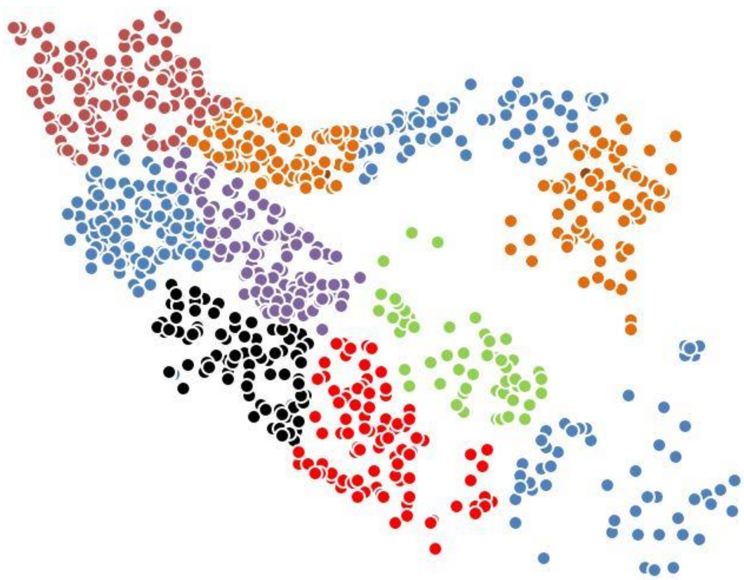


**Fig 14.** The final partitioning structure of the case study

To analyze sensitivity of the proposed algorithm performance in solving the case study, numerical results for different values of $\Gamma 1 = \frac{|J1|}{2}$ $\Gamma 2 = \frac{|J1|}{2}$, $\Gamma 1 = \frac{|J1|}{3}$ $\Gamma 2 = \frac{|J1|}{3}$, $\Gamma 1 = \frac{|J1|}{4}$ $\Gamma 2 = \frac{|J1|}{4}$, and $\Gamma 1 = \frac{|J1|}{5}$ $\Gamma 2 = \frac{|J1|}{5}$ are reported. Table 5 shows the objective function values for each created partitions.

**Table 5.** The objective function values for the partitions

| پهنه | $\Gamma 1 = \frac{\lvert J1 \rvert}{2}$ | $\Gamma 2 = \frac{\lvert J1 \rvert}{2}$ | $\Gamma 1 = \frac{\lvert J1 \rvert}{3}$ | $\Gamma 2 = \frac{\lvert J1 \rvert}{3}$ | $\Gamma 1 = \frac{\lvert J1 \rvert}{4}$ | $\Gamma 2 = \frac{\lvert J1 \rvert}{4}$ | $\Gamma 1 = \frac{\lvert J1 \rvert}{5}$ | $\Gamma 2 = \frac{\lvert J1 \rvert}{5}$ |
|---|---|---|---|---|---|---|---|---|
| | F1 | F2 | F1 | F2 | F1 | F2 | F1 | F2 |
| 1 | 185 | 1522488 | 153 | 1268490 | 172 | 1251046 | 177 | 1327309 |
| 2 | 198 | 1450048 | 149 | 1336178 | 152 | 1299656 | 197 | 1374532 |
| 3 | 288 | 933640 | 246 | 847842 | 243 | 880579 | 266 | 711917 |
| 4 | 275 | 1855679 | 202 | 1441249 | 210 | 1346297 | 237 | 1570229 |
| 5 | 207 | 790832 | 178 | 654728 | 152 | 635829 | 186 | 722323 |
| 6 | 301 | 1507125 | 286 | 1367168 | 258 | 1288178 | 259 | 1181550 |
| 7 | 283 | 930131 | 261 | 829630 | 260 | 846349 | 255 | 863209 |
| 8 | 282 | 660395 | 268 | 564261 | 275 | 541937 | 255 | 645863 |
| 9 | 283 | 1081343 | 260 | 988316 | 223 | 893226 | 248 | 918347 |
| 10 | 209 | 726486 | 197 | 663846 | 186 | 644481 | 169 | 711672 |
| | $Max - Min = 1195284$ | | $Max - Min = 876988$ | | $Max - Min = 804360$ | | $Max - Min = 924366$ | |

As can be seen, the more the uncertainty level, the greater the balance in the partitions. It indicates that decreasing the uncertainty level leads to better results in solving the problem. In fact, decision making process becomes more appropriate, and the algorithm will be able to find more qualified solutions. Figure 15 shows the change trend of objective function values for different uncertainty level in all partitions.

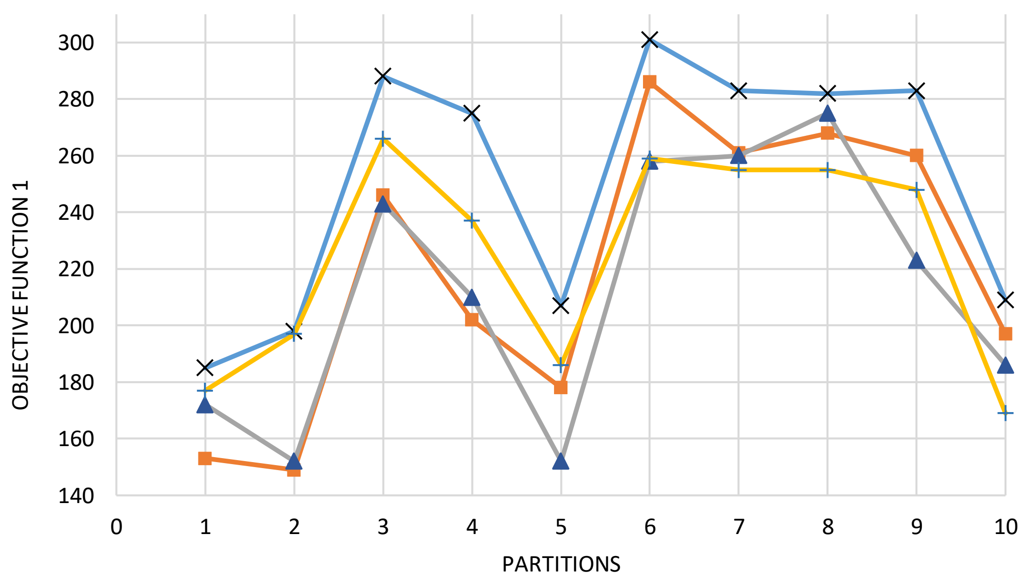



**Fig 15.** The first objective function value obtained from the SPEAII algorithm

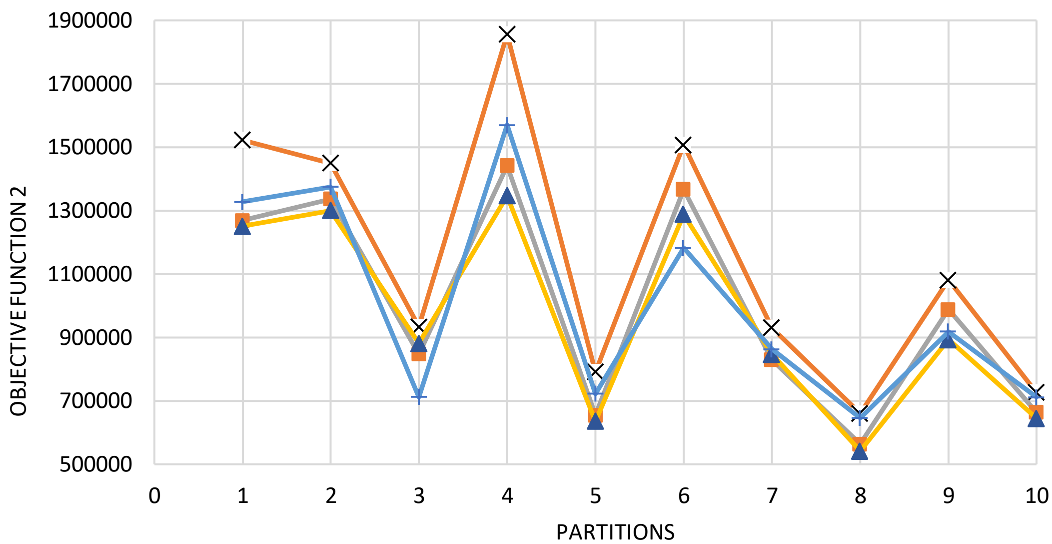



**Fig 16.** The second objective function value obtained from the SPEAII algorithm

In figure 16, different signs of ×, the square, the triangle, and + are respectively considered for uncertainty levels of $\Gamma 1 = \frac{|J1|}{2}$ $\Gamma 2 = \frac{|J1|}{2}$, $\Gamma 1 = \frac{|J1|}{3}$ $\Gamma 2 = \frac{|J1|}{3}$, $\Gamma 1 = \frac{|J1|}{4}$ $\Gamma 2 = \frac{|J1|}{4}$, and $\Gamma 1 = \frac{|J1|}{5}$ $\Gamma 2 = \frac{|J1|}{5}$. The uncertainty level $\Gamma 1 = \frac{|J1|}{5}$ $\Gamma 2 = \frac{|J1|}{5}$ and $\Gamma 1 = \frac{|J1|}{2}$ $\Gamma 2 = \frac{|J1|}{2}$ have respectively the greatest values in the first and the second objective functions. It can be clearly seen that the lowest difference between the largest and the lowest population of partitions is related to $\Gamma 1 = \frac{|J1|}{5}$ $\Gamma 2 = \frac{|J1|}{5}$ which indicates that better results are obtained from lower level of uncertainty.

## 7-Conclusions and future suggestions

In this paper, a bi-objective mathematical model is proposed for population partitioning problem under uncertainty conditions. This model can be used as a supporting tool for managers to make final decisions. Since the proposed model is in the category of non-deterministic polynomial hard (NP-hard) problems, meta-heuristic algorithms should be applied to solve real-world problems. Hence, three meta-heuristic algorithms PESA, SPEAII, and NSGAII are proposed. To evaluate the efficiency of the proposed algorithms, suitable comparison criteria are considered.

After investigation of the results, it is found that the SPEAII algorithm has the highest performance level than the other algorithms. The obtained Pareto members are non-dominated. A thorough investigation of one of the produced members confirmed that the optimal structure has complete feasible conditions. Therefore, the results of this model can be used for implementation in real-world conditions. After comparing the proposed algorithms, it can be seen that the NSGAII algorithm results has more distance from the generated front by CPLEX with comparison to the others. This matter suggests that using operators of SPEAII and PESA algorithms can lead to more suitable results.

Comparison of results of different numerical examples reveals that SPEAII algorithm has the highest level of performance and can be used as the final algorithm. In addition, increasing robustness parameter affects the obtained results, and the criterion of the sum of objective functions increases. In other words, more uncertainty results higher level of objective functions and worsen the results. To expand the scope of the research, some suggestions are presented below.

- The first suggestion can be implementation of results in larger real-world environments. Investigation of the obtained results can clearly show the performance range of the model and algorithms.

- Using other new meta-heuristic algorithms and comparing the final results can be considered as another research proposal. This can provide a suitable field for generating better solutions by other algorithms as well as comparing the functionality of different algorithms in this problem.
- Since some parameters of the problem cannot be estimated exactly, using appropriate approaches to deal with uncertainty will expand the scope of the problem. One of the most credible approaches to deal with uncertainty is robust programming which leads to constructive solutions to changes.
- Proposing exact algorithms such as branch and bound and branch and cut can also provide a guarantee for obtaining exact solutions in medium-size and occasionally large-size problems.

## References:


Baños, R., Gil, C., Paechter, B., & Ortega, J. (2007). A hybrid meta-heuristic for multi-objective optimization: MOSATS. *Journal of Mathematical Modelling and Algorithms, 6*(2), 213-230.

Baqir, R. (2002). Districting and government overspending. *Journal of political Economy, 110*(6), 1318-1354.

Benzarti, E., Sahin, E., & Dallery, Y. (2013). Operations management applied to home care services: Analysis of the districting problem. *Decision Support Systems, 55*(2), 587-598.

Bertsimas, D., & Sim, M. (2004). The price of robustness. *Operations Research, 52*(1), 35-53.

Bertsimas, D., & Sim, M. J. M. p. (2003). Robust discrete optimization and network flows. *98*(1-3), 49-71.

Bozkaya, B., Erkut, E., & Laporte, G. (2003). A tabu search heuristic and adaptive memory procedure for political districting. *European Journal of Operational Research, 144*(1), 12-26.

Brooks, S. P., & Morgan, B. J. (1995). Optimization using simulated annealing. *Journal of the Royal Statistical Society: Series D (The Statistician), 44*(2), 241-257.

Butsch, A., Kalcsics, J., & Laporte, G. (2014). Districting for arc routing. *INFORMS Journal on Computing, 26*(4), 809-824.

Camacho-Collados, M., Liberatore, F., & Angulo, J. M. (2015). A multi-criteria police districting problem for the efficient and effective design of patrol sector. *European Journal of Operational Research, 246*(2), 674-684.

Chen, X., & Yum, T.-S. P. (2010). *Patrol districting and routing with security level functions.* Paper presented at the 2010 IEEE International Conference on Systems, Man and Cybernetics.

Corne, D. W., Jerram, N. R., Knowles, J. D., & Oates, M. J. (2001). *PESA-II: Region-based selection in evolutionary multiobjective optimization.* Paper presented at the Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation.

Datta, D., Figueira, J., Gourtani, A., & Morton, A. (2013). Optimal administrative geographies: an algorithmic approach. *Socio-Economic Planning Sciences, 47*(3), 247-257.

Datta, D., & Figueira, J. R. (2011). Graph partitioning by multi-objective real-valued metaheuristics: A comparative study. *Applied Soft Computing, 11*(5), 3976-3987.

De Assis, L. S., Franca, P. M., & Usberti, F. L. (2014). A redistricting problem applied to meter reading in power distribution networks. *Computers & Operations Research, 41*, 65-75.

Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2000). *A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II.* Paper presented at the International Conference on Parallel Problem Solving From Nature.

Fan, N., Zheng, Q. P., & Pardalos, P. M. (2012). Robust optimization of graph partitioning involving interval uncertainty. *Theoretical Computer Science, 447*, 53-61.

García-Ayala, G., González-Velarde, J. L., Ríos-Mercado, R. Z., & Fernández, E. (2016). A novel model for arc territory design: promoting Eulerian districts. *International Transactions in Operational Research, 23*(3), 433-458.

Garfinkel, R. S., & Nemhauser, G. L. (1970). Optimal political districting by implicit enumeration techniques. *Management Science, 16*(8), B-495-B-508.

Ghiggi, C., Puliafito, P. P., & Zoppoli, R. (1975). *A combinatorial method for health-care districting.* Paper presented at the IFIP Technical Conference on Optimization Techniques.

Kong, Y., Zhu, Y., & Wang, Y. (2018). A center-based modeling approach to solve the districting problem. *International Journal of Geographical Information Science*, 1-17.

Li, W., Church, R. L., & Goodchild, M. F. (2014). An extendable heuristic framework to solve the p-compact-regions problem for urban economic modeling. *Computers, Environment and Urban Systems, 43*, 1-13.

Liberatore, F., & Camacho-Collados, M. (2016). A comparison of local search methods for the multicriteria police districting problem on graph. *Mathematical Problems in Engineering, 2016*.

Lin, H.-Y., & Kao, J.-J. (2008). Subregion districting analysis for municipal solid waste collection privatization. *Journal of the Air & Waste Management Association, 58*(1), 104-111.

Lin, M., Chin, K.-S., Fu, C., & Tsui, K.-L. (2017). An effective greedy method for the Meals-On-Wheels service districting problem. *Computers & Industrial Engineering, 106*, 1-19.

Maghsoudlou, H., Kahag, M. R., Niaki, S. T. A., & Pourvaziri, H. (2016). Bi-objective optimization of a three-echelon multi-server supply-chain problem in congested systems: Modeling and solution. *Computers & Industrial Engineering, 99*, 41-62.

Minciardi, R., Puliafito, P., & Zoppoli, R. (1981). A districting procedure for social organizations. *European Journal of Operational Research, 8*(1), 47-57.

Pezzella, F., Bonanno, R., & Nicoletti, B. (1981). A system approach to the optimal health-care districting. *European Journal of Operational Research, 8*(2), 139-146.

Ríos-Mercado, R. Z., & López-Pérez, J. F. (2013). Commercial territory design planning with realignment and disjoint assignment requirements. *Omega, 41*(3), 525-535.

Samadi, A., Mehranfar, N., Fathollahi Fard, A., & Hajiaghaei-Keshteli, M. (2018). Heuristic-based metaheuristics to address a sustainable supply chain network design problem. *Journal of Industrial and Production Engineering, 35*(2), 102-117.

Shirabe, T. (2012). Prescriptive modeling with map algebra for multi-zone allocation with size constraints. *Computers, Environment and Urban Systems, 36*(5), 456-469.

Steiner, M. T. A., Datta, D., Neto, P. J. S., Scarpin, C. T., & Figueira, J. R. (2015). Multi-objective optimization in partitioning the healthcare system of Parana State in Brazil. *Omega, 52*, 53-64.

Tran, T.-C., Dinh, T. B., & Gascon, V. (2017). *Meta-heuristics to Solve a Districting Problem of a Public Medical Clinic.* Paper presented at the Proceedings of the Eighth International Symposium on Information and Communication Technology.

Wang, T., Wu, Z., & Mao, J. (2007). *A new method for multi-objective tdma scheduling in wireless sensor networks using pareto-based pso and fuzzy comprehensive judgement.* Paper presented at the International Conference on High Performance Computing and Communications.

Zhao, J., Wang, D., & Peng, Q. (2018). *Optimizing the Train Dispatcher Desk Districting Problem in High-Speed Railway Network* (No. 18-03607).

Zitzler, E., Laumanns, M., & Thiele, L. (2001). SPEA2: Improving the strength Pareto evolutionary algorithm. *TIK-report, 103.*