

A permutation decision making method with multiple weighting vectors of criteria using NSGA-II and MOPSO

Mahdi Bashiri* and Ehsan AliAskari

Department of Industrial Engineering, Shahed University, Persian Gulf Highway, Tehran, Iran

CHRONICLE

ABSTRACT

Article history:

Received January 15, 2013
Accepted October 12, 2013
Available online
November 25 2013

Keywords:

Permutation Method
Different criteria weight vectors
Non-dominated permutations
NSGA-II
MOPSO

In decision making when multiple criteria are determined, the best choice depends on having complete information and proper decision-making technique. The permutation method is one of the popular techniques used in the context of multiple criteria decision making (MCDM). In this paper, a method is presented where there is more than one vector of weights for the criteria and there are uncertainties associated with criteria weights or there are multiple decision makers. We first take different weight vectors to create a multi-objective problem and then we solve them simultaneously to achieve appropriate Pareto solutions of the permutation method. Therefore, MOPSO and NSGA-II algorithms are utilized to find non-dominated solutions. Some examples in different sizes are considered to compare the efficiency of the proposed methods. Results show that by increasing the number of options and considering the computational time, the proposed methods perform better compared with the exact method. Moreover, NSGA-II is more efficient than MOPSO for the considered problem.

© 2014 Growing Science Ltd. All rights reserved.

1. Introduction

In the real world problems, most decision making problems face multiple criteria instead of dealing with only one criterion and there have been tremendous efforts to handle such cases. Tabriz et al. (2011) used Pareto multiple criteria decision making (MCDM) model in polyethylene industry to perform concurrent design. Athawale and Chakraborty (2011) implemented MCDM methods for industrial robot selection and ranked them based on their performance. Momeni et al. (2011) proposed a model for clustering economies based on MCDM techniques. Chatterjee and Bose (2012) applied a complex proportional assessment based on MCDM methodology under fuzzy environment for site selection of wind farm. Yen and Xu (2013) proposed a fuzzy MCDM algorithm to evaluate alternative recycling activities of an e-waste recycling job.

* Corresponding author.

E-mail addresses: bashiri@shahed.ac.ir (M. Bashiri)

In recent years, many researchers have concentrated on using evolutionary algorithms in multi-criteria decision-making. Dehuri and Cho (2009) presented a multi-objective Pareto based particle swarm optimization (MOPPSO) to minimize the architectural complexity and to maximize the classification accuracy of a polynomial neural network. Moslehi and Mahnam (2010) proposed an approach based on a hybridization of the particle swarm and local search algorithm to solve the multi-objective flexible job-shop scheduling problem. Ding et al. (2011) introduced a co-evolutionary immune algorithm for MCDM model and used the model to solve the large scale garment matching problem. Fallah-Mehdipour et al. (2011) applied two evolutionary algorithms, MOPSO and NSGA-II to solve two time-cost trade-off problems in the field of project management. Xu and Li (2012) used a MOPSO algorithm with permutation-based representation to solve dynamic construction site layout planning problem in fuzzy random environment.

“The Permutation” is a kind of decision-making method for multi attribute decision making (MADM) problems in which the ranking of alternatives is fundamentally described by the decision-making matrix and the weights of indices. In this method, the practical (applied) possible priorities (preferences) of alternatives are evaluated (assessed) to select the best alternatives order, which acquires the highest score. Additionally, in permutation method, with a slight increase in the number of alternatives, duration of finding the optimized (optimum) order demonstrates remarkable (exponential) rise. Rinnooy (1976) confirmed that if the number of alternatives increases, the solution of permutation problems is converted to an NP-hard one. There were several applications for this method in previous studies. For instance, Paelinck (1977) used the permutation method for the localization in air plants for the first time. Ancot and Paelink (1982) used the mentioned method in the management of municipal services such as water supplies, garbage collection and transportation. Blair and Karinsky (1994) analyzed some records of brain waves by applying the permutation method. Chen and Wang (2009) defined a kind of interval-valued permutation method for solving MADM problems with fuzzy complexes and demonstrated how this method can be applied in group-based decision making. Bashiri and Jalili (2010) defined an approach in which the best permutation of alternatives is extended by improved genetic algorithms. Karimi and Rezaenia (2011) demonstrated the modified permutation method for solving decision-making problems. Zavadskas et al. (2011) applied permutation technique to select the most preferable construction enterprises management strategy. Bashiri et al. (2012) presented a fuzzy permutation technique for MCDM problems. Tavana and Zandi (2012) utilized the permutation method in evaluating the scenarios, which are associated with the journeys to Mars. In all of the mentioned studies, it was assumed that there was only one decision maker in addition to one weight vector for defined criteria but in real world there are problems with multiple decision makers and associated weight matrices then it is necessary to consider whole decision makers' preferences to select preferred alternative.

This paper has been organized according to the following structure; in the second part, at first the permutation method for the decision making with existence of multiple criteria are explained and then it continues by determining the stages of proposed method. In the third and fourth part of this paper, the MOPSO and NSGA-II algorithms are evaluated for identifying of non-dominated solutions. Additionally, the fifth part contains results of solving some numerical examples by using the described algorithms and then the results of exact method have been compared to the results of the proposed algorithm. Conclusively, the sixth part has defined the final result.

2. Permutation method in decision making with multiple criteria

One of the methods applied for solving the problems of decision making and it is not totally identified as a kind of compensatory method, is the permutation explained, subsequently. In this method, all of the permutations related to the ranking of alternatives are considered and the most preferable order (arraignment) is selected. In other words, if $P_i = (\dots, A_k, \dots, A_l, \dots)$ is assumed as the i^{th} permutation from the whole ones, the alternative of A_k is considered higher than the A_l . In this way, C_{kl} is considered as the collections of criteria in which the A_k is higher than the A_l and D_{kl} is considered as

the collections of criteria in which the A_k has not higher value than the A_l . The permutation method, selects a permutation which can acquire the highest score according to the Eq. (1):

$$\begin{cases} R_i = \sum_{j \in C_{kl}} W_j - \sum_{j \in D_{kl}} W_j \\ \forall k, l \end{cases} \quad (1)$$

This scoring system practically shows the conformance of considered order with the decision maker's opinion while W_j indicates the weight of j^{th} criterion and R_i is the i^{th} permutation score. If the problem has two or more decision makers or the decision maker cannot define a unique weight for the criteria, in this situation, there is practically more than one weight vector for various criteria in the decision making, so the equation (1) is converted to the Eq. (2):

$$\begin{cases} R_{mi} = \sum_{j \in C_{kl}} W_{mj} - \sum_{j \in D_{kl}} W_{mj} \quad m = 1, \dots, M \\ \forall k, l \end{cases} \quad (2)$$

where m shows the number of weight vectors and W_{mj} indicates the weight of j^{th} criteria in the m^{th} weighted vector. Therefore, the decision-making permutation problem is converted to a kind of m -objective problem while each objective function value indicates the score of a permutation in accordance with related weight vector.

As mentioned before, solving the problems of decision making with permutation method is considered as a kind of NP-hard problems and because of that increasing the number of alternatives will need a significant amount of computational time. Indeed, using the meta-heuristic algorithms seems to be more useful. In a case of multiple weight vectors the problem will be as the multi objectives problem, so, it seems that a multi-objective meta-heuristic algorithm will be more appropriate for identifying the non-dominated permutations. Accordingly, in the following part of this study two multi-objective meta-heuristic methods which are called multi objective particle swarm optimization (MOPSO) and NSGA-II are discussed and explained.

3. MOPSO Algorithm for the problems with multiple weight vectors

Before defining MOPSO algorithm, the particle swarm optimization (PSO) must be explained, shortly. In PSO, initially some predefined particles are produced incidentally in the solution space. The situation of each particle indicates an order of alternatives. In addition, the fitness function value for algorithm is known identically as the determined score for the considered permutation. In each iteration, each particle should be moved according to other particles situations. Two kinds of solutions should be updated in each iteration of this algorithm. The first variable which is shown by $PBest$ is the best permutation which has been experienced by each particle while the second one called as $GBest$ is the best experienced permutation by all particles. Particles movement direction and their final position in each iteration will be calculated according to the following equations:

$$V_i(t+1) = w \cdot V_i(t) + c_1 r_1 (PBest_i(t) - X_i(t)) + c_2 r_2 (GBest(t) - X_i(t)), \quad (3)$$

$$X_i(t+1) = X_i(t) + V_i(t+1). \quad (4)$$

where, $V_i(t)$ is the velocity vector of i^{th} particle in the t^{th} iteration moment, $X_i(t)$ shows the i^{th} particles situation in t^{th} iteration, w , c_1 and c_2 are the constant numbers which demonstrate the share (importance factors) of the particle's present situation, the best experienced position of a particle and the best experienced permutation of the whole particles, respectively. Finally, r_1 and r_2 are the incidental numbers between zero and one.

Generally, the pseudo code of PSO based algorithm can be illustrated as following:

1. Create some permutation randomly as the initial population
2. Evaluate the score of each permutation according to Eq. (1)
3. Update $PBest$ and $GBest$ for each permutation
4. Move each permutation toward the selected positions using Eq. (3) and Eq. (4)
5. Calculate the fitness value of new permutations using Eq. (1)
6. Repeat the algorithm while the stopping condition has not been met.

If there are several weight vectors or decision makers, it is possible that a permutation acquires a high value according to the first vector while contains less score according to the other one. In such cases it more preferable that we extract non dominated permutations. It would be mentioned that in this method, the decision makers only face a few non-dominated permutations instead of facing a lots of permutations which facilitates the process of decision making.

3.1 Concept of domination in a problem with different weight vectors

If the R_{mi} is assumed as the score of the P_i permutation with weight vector m according to the equation (2) and the R_{mj} is anticipated as the score of P_j permutation with the same weight vector, the P_i permutation dominates the P_j if the following condition is satisfied:

$$R_{mi} \text{ dom } R_{mj} \Leftrightarrow \begin{cases} \forall m: R_{mi} \geq R_{mj} \\ \exists m: R_{mi} > R_{mj} \end{cases} \quad (5)$$

The group of solutions which are not dominated by the others is known as the *Pareto Front* and finding of Pareto solutions is the primary objective of multiple objectives optimization algorithms. The MOPSO algorithm demonstrated by Coello Coello (2004) identifies Pareto in multi objective problems. In this algorithm, during calculating the velocity vector for each particle, one permutation from the Pareto Front should be selected as the leader. The leader permutation in this algorithm plays identically as the $GBest$ in PSO algorithm. It would be mentioned that because of the change in Pareto Front or in conditions in different repetitions, the selected leader permutation is repeatedly altered. The process of selecting leader permutation is explained in the next section.

3.2 Selecting process of leader permutation

Selecting the leader permutation should be operated in a way to increase the variety of permutations created in Pareto Front. Indeed, the PESA-II technique proposed by Corne et al. (2001) is used. According to this technique, the spaces for answers are divided into equal parts and selecting leader method is related to the regional base instead of individual base. This process can be illustrated briefly as following:

1. Divide the solution space into equal regions
2. Select regions which contain at least one non-dominated permutation
3. Give a chance to selected regions according to their respective number of non-dominated permutations using Eq. (6)
4. Choose one of the selected regions according its calculated chance
5. Choose one of the non-dominated permutation in the selected region randomly as the leader permutation for the considered particle

$$P_i = \frac{e^{-\beta n_i}}{\sum_{j=1}^{n_r} e^{-\beta n_j}}, \quad (6)$$

where P_i shows the probability of selecting i^{th} region, n_r is the number of regions in the solution space and β is the algorithm parameter between zero and one which must be tuned during the algorithm operation.

3.3 Updating $PBest$ and archive of non-dominated permutations

After creating a new permutation, the $PBest$ and non-dominated permutation's archive should be updated. Updating process includes following stages:

1. If the new-created permutation by each particle, dominates its $PBest$, should be replaced as the new $PBest$. If the original $PBest$ dominates new-created permutation, it remains unchanged. If the new-created permutation and original $PBest$ cannot dominate each other, one of them is selected as the $PBest$ of the newer particle randomly.
2. If the $PBest$ permutation is changed during the previous stage, it must be added to the non-dominated permutations archive. Then the archive should be screened.

3.4 General steps of MOPSO algorithm with multiple weight vectors

The general steps of MOPSO algorithm with multiple weight vectors by using the permutation method can be declared as follows:

1. Create some permutations randomly as the initial population,
2. Calculate the score of each permutation with each weight vector (R_{mi}) based on the Eq. (2),
3. Identify non-dominated permutations and update the archive, simultaneously,
4. Update $PBest$ for each particle (permutation),
5. Select a leader permutation according to PESA-II technique for each particle,
6. Calculate velocity vector for each particle according to Eq. (3),
7. Move the particles to the new positions according to the Eq. (4),
8. If the frustration condition doesn't exist, algorithm must be repeated from step (2). Otherwise report the final archive as Pareto front.

4. NSGA-II algorithm for solving the decision-making problem with multiple weight vectors

For identifying the appropriate algorithm in solving the considered problem, NSGA-II algorithm as a kind of meta-heuristic methods has been applied to the problem. This algorithm has been introduced by Deb et al. (2002). New population creation procedure is its efficiency reason. In this algorithm, for selecting the members of new principle population, quality-base criterion is considered as the first criterion and diversity is considered as the second one. For evaluating the quality of answers, the non-domination rank is applied as a comparative criteria while a mounts of diversity is detected by a criterion called crowding distance. The definitions for these two criteria have been illustrated in the following paragraphs, respectively:

4.1 Non-domination rank of a permutation

The whole permutations in the population are compared with each other and the number of times in which the p permutation is dominated by the other one is computed and called n_p . Additionally, a group of population members which are dominated by p permutation should be called S_p . Other

permutations which have not been dominated anyway create the first front. For detecting the members of the second front, the first front members should be excluded from the solutions to achieve the second front. Continuing this procedure divides the whole solutions into appropriate fronts. The non-dominated sorting algorithm comprises:

1. For each two members of the population such as p and q , if the p permutation dominates the q one, the q permutation joins to the S_p . On the other hand the q permutation shows dominancy, then one unit should be added to the n_p .
2. All of the solutions with their n_p is equal to zero must be added to F_1 (first front). After excluding the mentioned solutions, the S_p and n_p should be updated again and the Pareto answers are identified for the next ranks. This procedure will be continued until no solution remains.

For example, in a population with six permutations the objective value of R for each permutation has been calculated considering of two weight vectors for two decision makers, so the objective values of the first decision maker is 12, 10, 3, 8, 6 and 6, while the values for the second one will be 4, 13, 16, 3, 10 and 5. The ranking procedure of the permutations have been depicted in the Fig.1.

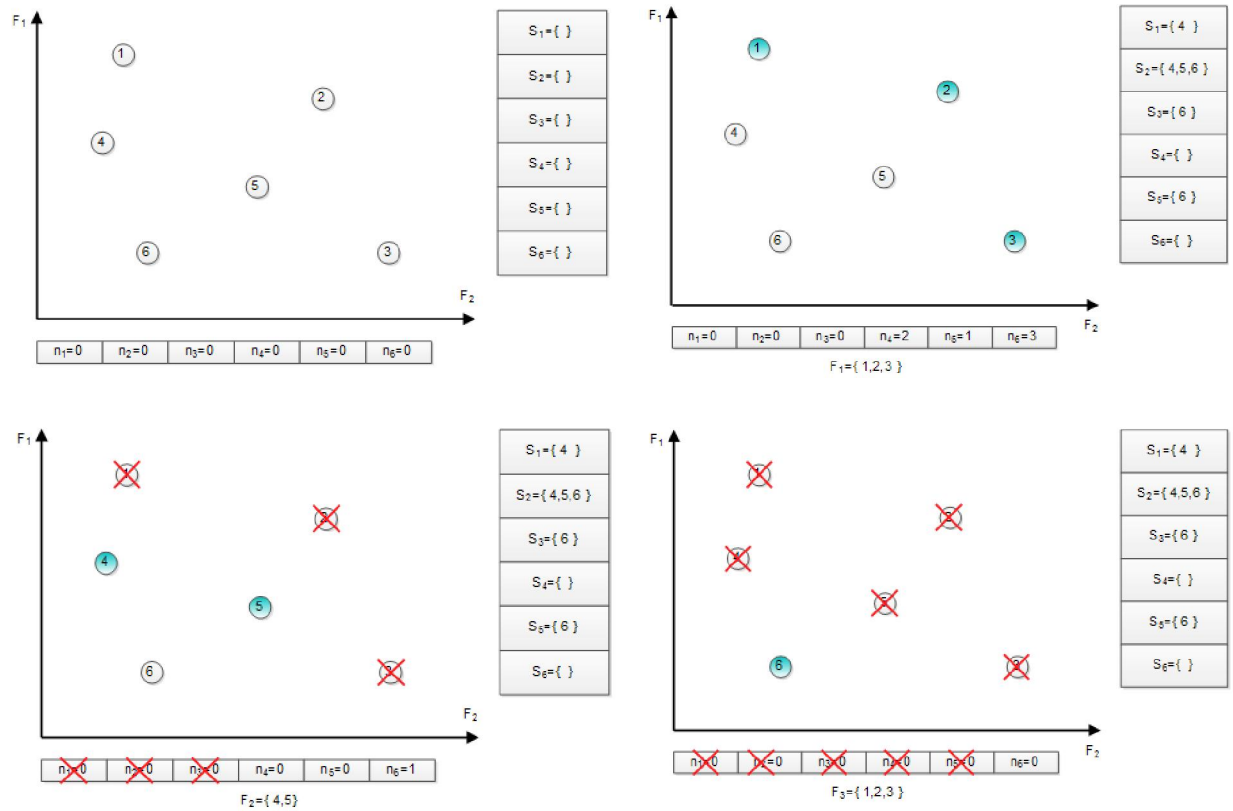


Fig. 1. Stages of determining Pareto answers with various ranks in the six permutation example

4.2 Defining the crowding distance

The crowding distance is a criterion which demonstrates the amounts of compression around each permutation. The larger amount of the mentioned criterion means the less severity of compression around the counterpart permutation. This distance is calculated by the Eq. (7) and Eq. (8):

$$d_i^j = \frac{|f_j^{i+1} - f_j^{i-1}|}{f_j^{max} - f_j^{min}} \quad (7)$$

$$d_i = \sum_j d_i^j. \quad (8)$$

In this case, the d_i^j shows the crowding distance of i^{th} solution on the axis of j^{th} objective function while the d_i is the crowding distances for the i^{th} permutation on the whole objective functions.

The NSGA-II algorithm for selecting the members of principal population initially sorts all of the permutations according to their quality criteria and then each permutation is categorized in accordance with the diversity criteria for the same ranked solutions. The flow chart of this applied algorithm for solving permutation problems is illustrated in the Fig. 2.

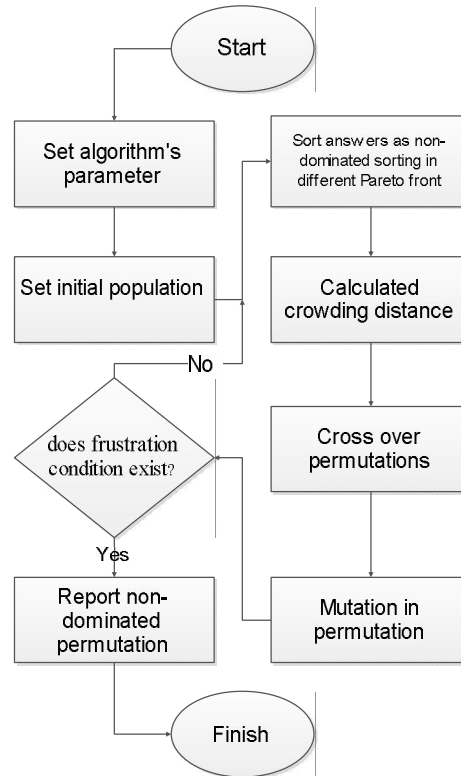


Fig. 2. NSGA-II algorithm for solving permutation problems with multiple

5. Numerical examples

As it has been mentioned, solving decision making problem with the permutation method with the multiple weighted vectors of the criteria needs large duration of calculating time with the exact method as the number of alternatives increases. Indeed, using meta-heuristic multiple objective methods seems to be more appropriate. For evaluating the efficiency of two algorithms in problem solving, simulated numerical examples are studied. For instance, suppose there are two decision making committees in the municipal council of a city to decide about replacing an alternative car type for old taxi cabs. The first is transportation committee and the other one is environment and beautification committee. For selecting the best model among fifteen types of cars, six criteria have been considered in which the fourth criteria has negative direction. These criteria comprise: 1. The passenger capacity, 2. GPS equipment, 3. Beautifulness, 4. Pollution rate, 5. Safety and 6. Hybrid fuel system. In this case, two decision makers (committees) are interested in deciding independently for selecting the best permutation according to the criteria importance while they have different opinions

about the priority of importance, so the non-dominated permutations should be extracted. Two vectors of $w_1 = [0.47 \ 0.1 \ 0.1 \ 0.08 \ 0.22 \ 0.03]$ and $w_2 = [0.05 \ 0.06 \ 0.14 \ 0.41 \ 0.1 \ 0.24]$ are the weights of transportation and environmental committees, respectively. The decision matrix has been reported in Table 1. In the following part, the results extracted from the three methods of exact, NSGA-II and MOPSO have been evaluated according to the computational time, and quality measures. In this paper some quality measures have been analyzed which are reported in Table 1. One of mentioned measure is the Generational Distance (GD) which has been defined by Van Veldhuizen and Lamont (1998) and accounts the closeness of determined non dominated solutions to the real Pareto front. It is calculated according to the Eq. (9):

$$GD = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{n} \quad (9)$$

In this case, n shows the number of total determined non-dominated solutions while the d_i demonstrates Euclidean distance between each non-dominated answer and the nearest point in the real Pareto front. In this article, the Pareto front extracted by an exact solution approach is assumed as the real one and for large scale examples the best frontier achieved by all approaches is assumed as the real front. Other quality measures for extracted non-dominated solutions which are used in this study can be mentioned as follows:

1. The number of non-dominated permutations known by the each solution algorithm,
2. The similarity percentage of determined non-dominated permutations by an algorithm comparing to other algorithms,
3. The number of determined non-dominated permutations in each algorithm, which have been dominated by another non-dominated permutation in the other algorithms.

As mentioned before, some numerical examples were simulated by considering two decision making weights with different number of alternatives. For the numerical examples, we assumed that there are 4 to 15 alternatives. The results of mentioned measures have been shown in Table 1. Moreover in Fig. 3 to Fig. 14 extracted non dominated solutions by two approaches of NSGA-II and MOPSO have been compared. In Fig. 3 to Fig. 5 the Pareto fronts have been compared with the exact front because for medium and large sized problems the mentioned solutions could not be determined by exact solution approaches. Analyzing of mentioned figures in larger sizes shows that not only the function of NSGA-II algorithm is better than the MOPSO in identifying non-dominated permutations but also the Table 1 results emphasizes that the NSGA-II algorithm has more appropriate function in all of the evaluated indices in comparison with MOPSO. The computational time slopes according to the problem size for three solution approaches have been compared in Fig. 15. It shows that NSGA-II is more efficient than others for the permutation problem.

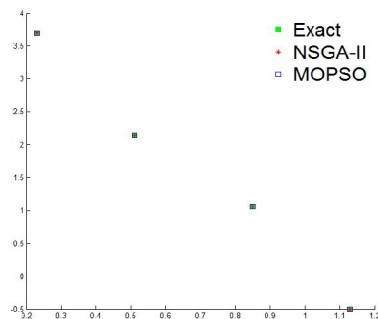


Fig. 3. Extracted non-dominated solutions by different approaches for permutation problem with 4 alternatives

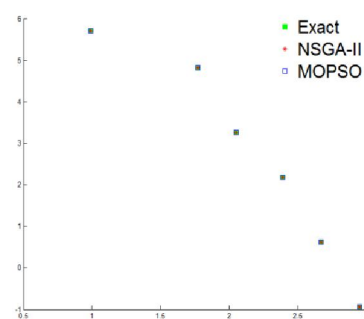


Fig. 4. Extracted non-dominated solutions by different approaches for permutation problem with 5 alternatives

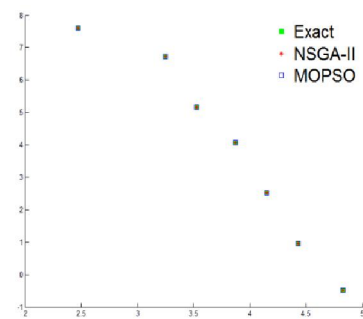


Fig. 5. Extracted non-dominated solutions by different approaches for permutation problem with 6 alternatives

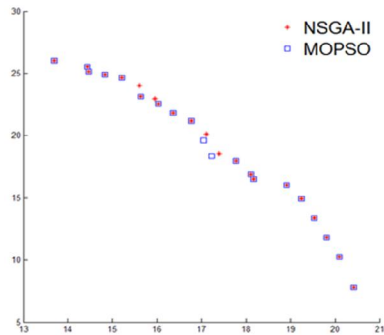


Fig. 6. Extracted non-dominated solutions by different approaches for permutation problem with 7 alternatives

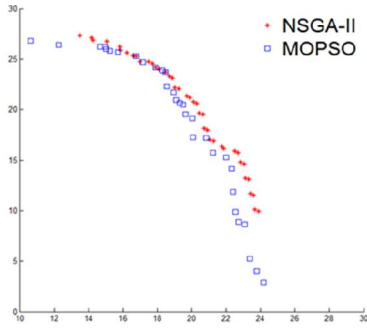


Fig. 7. Extracted non-dominated solutions by different approaches for permutation problem with 8 alternatives

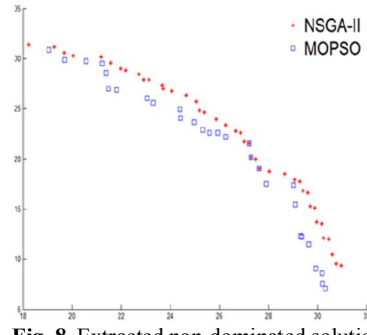


Fig. 8. Extracted non-dominated solutions by different approaches for permutation problem with 9 alternatives

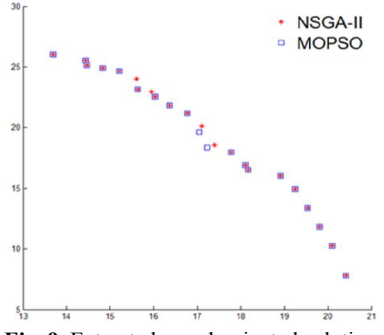


Fig. 9. Extracted non-dominated solutions by different approaches for permutation problem with 10 alternatives

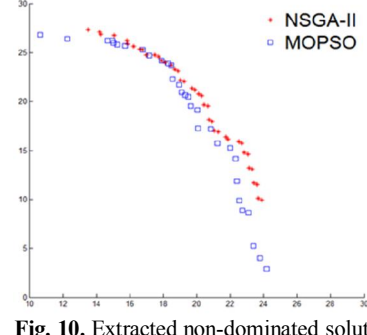


Fig. 10. Extracted non-dominated solutions by different approaches for permutation problem with 11 alternatives

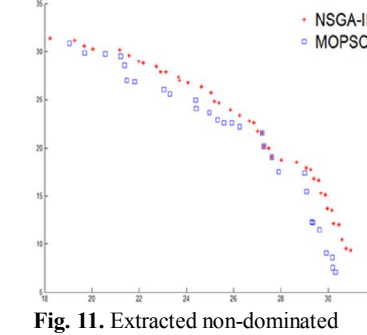


Fig. 11. Extracted non-dominated solutions by different approaches for permutation problem with 12 alternatives

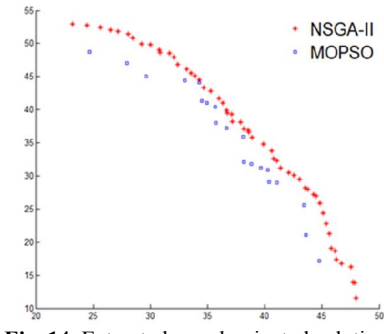


Fig. 14. Extracted non-dominated solutions by different approaches for permutation problem with 15 alternatives

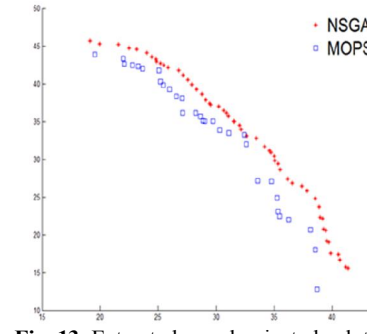


Fig. 13. Extracted non-dominated solutions by different approaches for permutation problem with 14 alternatives

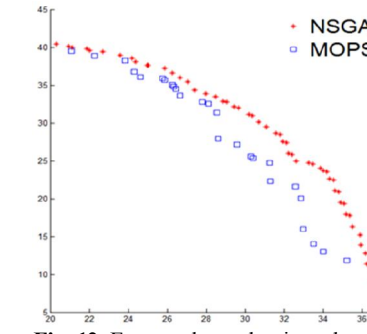


Fig. 12. Extracted non-dominated solutions by different approaches for permutation problem with 13 alternatives

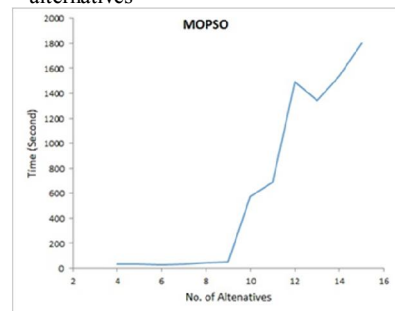
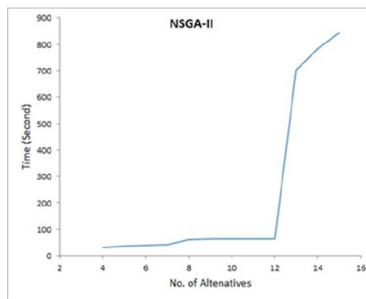
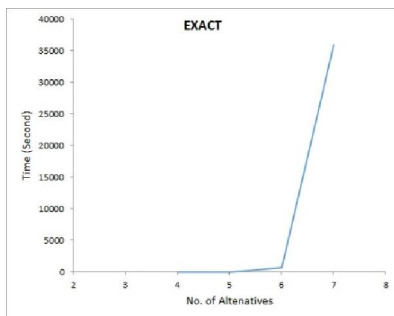


Fig. 15. Computational time slope by increasing of problem size in Exact, NSGA_II and MOPSO solution approaches

Table 1

Quality measures of extracted results by three solution approaches for different problem sizes

Method	Number of opinions	Number of extracted NDS	Computational time (second)	Similarity percentage of extracted NDS to the known best NDS	Similarity percentage of extracted NDS to the real NDS	Number of extracted NDS that dominated by the known best NDS	Number of extracted NDS that dominated by the real NDS	GD distance from the extracted known best NDS	GD distance from the extracted real NDS
Exact	4	4	0.079	100	100	0	0	0	0
NSGA-II		4	30.756	100	100	0	0	0	0
MOPSO		4	32.043	100	100	0	0	0	0
Exact	5	6	0.105	100	100	0	0	0	0
NSGA-II		6	36.812	100	100	0	0	0	0
MOPSO		6	30.944	100	100	0	0	0	0
Exact	6	7	700	100	100	0	0	0	0
NSGA-II		7	39.19	100	100	0	0	0	0
MOPSO		7	29.25	100	100	0	0	0	0
Exact	7	7	9h<	100	-	0	-	0	-
NSGA-II		7	40.577	100	-	0	-	0	-
MOPSO		7	35.463	100	-	0	-	0	-
NSGA-II	8	9	59.516	100	-	0	-	0	-
MOPSO		9	45.215	100	-	0	-	0	-
NSGA-II	9	14	63.654	100	-	0	-	0	-
MOPSO		14	52.1	100	-	0	-	0	-
NSGA-II	10	22	63.654	100	-	0	-	0	-
MOPSO		20	575.48	81.8	-	2	-	0.027368	-
NSGA-II	11	41	63.654	100	-	0	-	0	-
MOPSO		31	690.36	7.3	-	28	-	0.73243	-
NSGA-II	12	42	63.654	100	-	0	-	0	-
MOPSO		29	1492.35	7.14	-	26	-	0.83599	-
NSGA-II	13	75	700.24	100	-	0	-	0	-
MOPSO		26	1345.55	0	-	26	-	0.54927	-
NSGA-II	14	64	784.82	100	-	0	-	0	-
MOPSO		31	1542.6	0	-	29	-	0.58646	-
NSGA-II	15	61	844.02	100	-	0	-	0	-
MOPSO		20	1803.4	0	-	19	-	0.44031	-

6. Conclusion

In this study, the problem of multiple criteria decision making with the existence of various weighted vectors by considering multiple decision makers has been studied for the permutation problem. Existing of different weight vectors transform the problem into a multiple objective permutation problem. We have explained that by increasing the problem size, exact solution approach for determining of non-dominated solutions would not operate efficiently. Therefore, two metaheuristics; namely NSGA-II and MOPSO were applied to solve the multiple objectives permutation problem. Some numerical examples with different sizes were analyzed in a case where there were two different decision makers. The results have indicated that in a problem with more than 7 alternatives the exact solution approach needs significant amount of computational time while the proposed algorithms could find the non-dominated solutions, more efficiently. Moreover, the results have confirmed that the NSGA-II has better performance considering the computational time and other presented quality measures. As a future research this study can be extended for another state while decision makers have interactions to each other and their objective values will depend on the other one.

References

- A. Tabriz, A., Ahmadi, A., Maleki, M. H., Afshari, M. A. & S. Moradi, J. (2011). Applying Pareto multi-criteria decision making in concurrent engineering: A case study of polyethylene industry. *Management Science Letters*, 1, 289–294.

- Ancot, J., & Paelinck, J. H. P. (1982). Recent experiences with the Qualiflex multicriteria method. *Qualitative and Quantitative Mathematical Economics. Martinus Nijhoff Publishers*, 217-266.
- Bashiri, M., Jalili, M. (2010). Interactive permutation decision making based on genetic algorithm. *Proceedings of the IEEE IEEM*, 84-88.
- Blair, R., & Karnisky, W. (1994). Distribution-free statistical analysis of surface and volumetric maps. *Brain Topography*, 6, 19-28.
- Coello Coello, C. A., Pulido, G. T. & Lechuga, M. S. (2004). Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 8, No. 3, 256-279.
- Chen, T., & Wang, J. (2009). Interval-valued fuzzy permutation method and experimental analysis on cardinal and ordinal evaluations. *Journal of Computer and System Sciences*, 75, 371-387.
- Yeh, C. H. & Xu, Y. (2013). Sustainable planning of e-waste recycling activities using fuzzy multicriteria decision making. *Journal of Cleaner Production*, 52, 194-204.
- Van Veldhuizen D. A. & Gary B. Lamont, G. B. (1998). Multi-objective evolutionary algorithm research: A history and analysis. *Technical Report, Air Force Institute of Technology, Wright-Patterson AFB, OH*, TR-98-03.
- Deb, K., Pratap, A., Agarwal, S. & Meyarivan, T. (2002). A fast and elitist multi-objective genetic algorithm. NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6, 182-197.
- Zavadskas, E.K., Turskis, Z. & Tamosaitiene, J. (2011). Selection of construction enterprises management strategy based on the SWOT and multi-criteria analysis. *Archives of Civil and Mechanical Engineering*, 11, Pages 1063-1082.
- Fallah-Mehdipour, E., Bozorg Haddad, O., Rezapour Tabari, M.M. & Marino, M. A., (2012). Extraction of decision alternatives in construction management projects: Application and adaptation of NSGA-II and MOPSO. *Expert Systems with Applications*, 39, 2794-2803.
- Moslehi, G. & Mahnam, M. (2011). A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search. *International Journal of Production Economics*, 129, 14-22.
- Xu, J. & Zongmin Li, Z. (2012). Multi-objective dynamic construction site layout planning in fuzzy random environment. *Journal of Automation in Construction*, 27 155-169.
- Karimi, H. & Rezaeinia, A. (2011) Adjusted permutation method for multiple attribute decision making with meta-heuristic solution approaches. *International Journal of Industrial Engineering Computations*, 2, 369-384.
- Bashiri, M., Koosha, M. & Karimi, H. (2012). Permutation based decision making under fuzzy environment using Tabu search. *International Journal of Industrial Engineering Computations*, 3, 301-312.
- Momenia, M., Mehrabioun Mohammadia, M., Malekia, M. H. & Afsharib, M. A. (2011). Clustering economies based on multiple criteria decision making techniques. *Journal of Management Science Letters*, 1, 487-492.
- Chatterjee, N. C. & Bose, K. B. (2012). A COPRAS-F base multi-criteria group decision making approach for site selection of wind farm. *Decision Science Letters*, 2, 1-10.
- Paelinck, J. (1977), Qualitative multiple criteria analysis: an application to airport location. *En'liironment and Planning*, 9, 695-893.
- Rinnooy Kan, A. H. G. (1976). Machine scheduling problems, classification, complexity and computations. *Nijhoff, The Hague*.
- Dehuri, S. & Cho, S. B. (2009). Multi-criterion Pareto based particle swarm optimized polynomial neural network for classification: A review and state-of-the-art. *Computer Science Review*, 3, 19-40.
- Tavana, M. & Zandi, F. (2012). Applying fuzzy bi-dimensional scenario-based model to the assessment of Mars mission architecture scenarios. *Journal of Advances in Space Research*, 49, 629-647.

- Athawalea, V. M. & Chakraborty, S. (2011). A comparative study on the ranking performance of some multi-criteria decision-making methods for industrial robot selection. *International Journal of Industrial Engineering Computations*, 2, 831–850.
- Ding, Y. S., Hu, Z. H. & Zhang, W. B. (2011). Multi-criteria decision making approach based on immune co-evolutionary algorithm with application to garment matching problem. *Expert Systems with Applications*, 38, 10377–10383.