Solving Multi-Objective Assignment Problem with Decision Maker's Preferences by Using Genetic Algorithm

Md. Mahbubur Rahman
Department of Industrial Engineering and Management
Khulna University of Engineering & Technology
Khulna-9203, Bangladesh
mahbub.iem@gmail.com

Md. Kutub Uddin
Department of Mechanical Engineering
Khulna University of Engineering & Technology
Khulna-9203, Bangladesh
kutubuddin@me.kuet.ac.bd

Abstract

The multi-objective assignment problem is basically the N men –N tasks problem, where a single task has to be assigned to an individual with a view of optimizing the outcomes. A common challenge is to address the conflicting objectives which produce Pareto–optimal solutions. The main feature of the work is- normalizing all the criteria into a single scale regardless of their measurement units and their demand of minimum or maximum, which reliefs us from careful attention in quantifying the quality criteria. The methodology also included the decision maker's preferences regarding the objectives. While solving the problem through a genetic algorithm, a new encoding scheme is used together with a partially matched crossover (PMX). The working principle of the proposed algorithm is illustrated with a numerical example and its effectiveness has been compared with some well-established methodologies. It is found that the proposed algorithm provides a better solution with minimal computational effort.

Keywords
Assignment problem, Multi-objective, Decision maker's preferences, and Genetic Algorithm.

1. Introduction

The assignment problem (AP) is one of the fundamental topics in combinatorial optimization in the branch of operation research. It has significant use in production planning, transportation, telecommunication, VLSI design, economics etc. (Pramanik and Biswas 2012). It deals with the allocation of the various resources to the various activities so that an optimal assignment can be made in the best possible way.

The linear assignment problem manifests the scenario where assignees are being assigned to perform tasks in one to one basis (Sahu and Thapadar 2017). It addresses the question of how to set assignee to tasks in an injective way so that the assignment cost (or profit) is minimized (or maximized). In this perspective, an assignment problem is viewed as a balanced transportation problem in which all supplies and demands equal '1' for identical numbers of rows and columns. Transportation simplex is not applied here because of a high degree of degeneracy (Lin 2009).

Kuhn (1995) proposed an algorithm for the linear assignment problem known as the Hungarian method. It was primarily designed for hand computation (Taha 2006). It has third order run time requirement that is very tedious with a large number of the task (Bertsekas 1981). Solving multiple objectives optimization problem is not suitable with such an approach.

In the real arena, management has many objectives for tasks allocation to workers. We often come in close contact with an assignment problem, where, cost and time are jointly co-related (Pramanik and Biswas 2012). Multi-objective

assignment model usually considers time, cost, safety, quality etc. simultaneously. Single objective optimization is easy to solve but the multi-objective problem is complex because of the conflicting nature of the objectives (Oliveira and Saramago 2010). These problems give rise to a set of trade-off among optimal solutions, popularly known as Pareto-optimal solutions. For the multi-objective assignment problem (MOAP), all the criteria are not equally important. Generally, decision maker imparts priority ranking among the objectives, *e.g.* time is less important than quality. By incorporating the decision maker's preferences into the problem, the problem becomes hard to solve (Acar and Aplak 2010).

Bao et al. (2007) use 0-1 programming to translate a MOAP into a linear programming problem. They take the reciprocal of quantified quality in the normalized quality, but it doesn't translate the quality criteria into a similar scale like cost and time criteria when there is no quality point '1'. Moreover, there may be multiple local optimum solutions. Linear programming has not the ability to avoid being trapped in local optimal solution as it starts searching from a single point and moves to nearby better solution point. This may lead to trapped in local optimal solution (Ishizuka and Matsuo, 2002).

Pramanik and Biswas (2012) solved the MOAP with Generalized Trapezoidal Fuzzy Numbers. It suffers the drawback from imprecise costs, time and effectiveness instead of having precise information. In addition, it uses linear programming to find the solution.

Tsai et al. (1999) proposed a new methodology to solve the problem of multi-objective fuzzy deployment of manpower. They transform the multi-objective problem into a fuzzy linear programming. In this methodology, a careful attention must be paid to determine the weights among the resources. The attention of the management may skew to an erroneous assignment for an inappropriate set of weights. Also, this approach requires expensive calculation time.

Genetic Algorithm (GA) is successfully used to solve computer science and operations research problems. GA is a stochastic search and optimization technique inspired by the process of natural selection that belongs to the larger class of evolutionary algorithms (EA) (Goldberg 1989). GA is commonly used to generate high-quality solutions for optimization and search problems by relying on bio-inspired operators such as mutation, crossover, and selection (Boussaid et al. 2013). It is applicable for both constrained and unconstrained optimization problems (Yeniay, 2005). Moreover, GA has a better ability to avoid being trapped in local optimal solution as opposed to the linear programming (Melanie, 1999).

Tailor and Dhodiya (2016) presented a solution procedure of MOAP using a genetic algorithm based hybrid approach. They convert the MOAP into 'combine objective assignment problem' and then the solution is searched by genetic algorithm. This methodology also requires allocating equivalent weight to cost, time and quality like the work of Tsai et al. (1999).

The purpose of this research is to solve MOAP using a comparatively easy and effective algorithm based on GA principles, which will solve the MOAP incorporating the decision maker's preferences.

2. The Proposed Model

The representation of notation and the mathematical model is done as follows.

2.1 Notation

Subscripts

i	*Worker number*
j	*Task number*
n	*Number of total worker/ task*
r	*Chromosome's number*
m	*Number of total solutions/ chromosomes*
c	*Cost*
t	*Time*

q	*Quality*

Parameters and matrixes

M_c	*The Cost matrix*
M_t	*The Time matrix*
M_q	*The Quality matrix*
C_{ij}	*The element of the i^{th} row and j^{th} column in the cost matrix*
T_{ij}	*The element of the i^{th} row and j^{th} column in the time matrix*
Q_{ij}	*The element of the i^{th} row and j^{th} column in the quality matrix*
b	*Gene*
L	*Number of total genes in a chromosome*
C_{max}	*Maximum cost*
T_{max}	*Maximum time*
Q_{max}	*Maximum quality*
N_c	*Normalized cost matrix*
N_t	*Normalized time matrix*
N_q	*Normalized quality matrix*
Cn_{ij}	*The element of the i^{th} row and j^{th} column in the normalize cost matrix*
Tn_{ij}	*The element of the i^{th} row and j^{th} column in the normalize time matrix*
Qn_{ij}	*The element of the i^{th} row and j^{th} column in the normalize quality matrix*
W_c	*Weightage of cost*
W_t	*Weightage of time*
W_q	*Weightage of quality*
P_m	*Probability of mutation*

2.2 Mathematical model

The MOAP deals with cost, time, quality etc. (Deb 2001). The objectives of an assignment problem are to minimize both operating cost and operating time, and to maximize quality simultaneously (Mota et al. 2015). Suppose we have to assign *n* workers to *n* tasks in such a way that the overall operation cost, labour-time, and quality level are optimized.

It is noted that the units for measuring time, cost and quality are different. Generally, the quality criteria are expressed as “good”, “fair”, and “poor”. Therefore, it is necessary to quantify this quality criterion in terms of numerical value (Bao et al. 2007). We assign 1 for “good”, 3 for “fair” and 5 for “poor” or researcher can express quality into more level in any interval. This assignment imparts the highest value to the lowest quality and the lowest value to the highest quality (Tsai et al. 1999). It converts the requirement of maximum quality in MOAP into a minimum value of quality. Now, the demand for the value of all the criteria viz. cost, time and quality is minimal. The assignment cost, time and quality are given in table 1.

Table 1. A MOAP

Criteria	Worker, i	Task, j			
		1	***2***	**..**	***n***
Cost, C_{ij}	***1***	C_{11}	C_{12}	..	C_{1n}
	2	C_{21}	C_{22}	..	C_{2n}
	..	..	..	..	..
	N	C_{n1}	C_{n2}	..	C_{nn}
Time, T_{ij}	***1***	T_{11}	T_{12}	..	T_{1n}
	2	T_{21}	T_{22}	..	T_{2n}
	..	..	..	..	..
	N	T_{n1}	T_{n2}	..	T_{nn}
Quality, Q_{ij}	***1***	Q_{11}	Q_{12}	..	Q_{1n}
	2	Q_{21}	Q_{22}	..	Q_{2n}
	..	..	..	..	..
	N	Q_{n1}	Q_{n2}	..	Q_{nn}

The problem can be stated as,

$$Minimize \sum_{i=1}^{n} \sum_{j=1}^{n} X_{ij} C_{ij} \quad (1)$$

$$Minimize \sum_{i=1}^{n} \sum_{j=1}^{n} X_{ij} T_{ij} \quad (2)$$

$$Minimize \sum_{i=1}^{n} \sum_{j=1}^{n} X_{ij} Q_{ij} \quad (3)$$

Where

$$Xij = \begin{cases} 1 \; if \; i^{th} \; worker \;\; is \; assigned \; to \; j^{th} \; task \\ 0 \qquad\qquad\qquad\qquad Otherwise \end{cases} \quad (4)$$

$$\sum_{i=1}^{n} x_{ij} = 1; \; j = 1, 2, \ldots\ldots, n \; (only \; one \; person \; is \;\; assigned \; to \; the \; j^{th} \; task) \quad (5)$$

$$\sum_{j=1}^{n} x_{ij} = 1; \;\; i = 1, 2, \ldots\ldots, n \;\; (only \; one \; task \; is \; done \; by \; the \; i^{th} \; worker) \quad (6)$$

3. A Genetic algorithm for the MOAP

In this sub-section, we present the basis and the details of the proposed genetic algorithm.

3.1 Basis of the formulation

The values of one criterion (*e.g.* cost) may be very high and for another criterion (*e.g*. time) may be very low. In the case of minimization, the criteria having high values play an important role by ignoring the criteria having a low value (Mota et al. 2015). So it requires the conversion of all the criteria into a similar scale. The process of normalization translate all the criteria into a similar scale. For the purpose of normalization, first, maximum operation cost and maximum operation time and maximum operation quality are determined. To find the normalized matrix, all the cost, time and quality are divided by the maximum operation cost, maximum operation time, and maximum operation quality respectively.

Maximum cost,

$$C_{max} \; = \; Max \; (C_{ij}) \quad (7)$$

Maximum *time,*

$$T_{max} \; = \; Max \; (T_{ij}) \quad (8)$$

Maximum *quality,*

$$Q_{max} \; = \; Max \; (Q_{ij}) \quad (9)$$

Normalized cost matrix,

$$N_c = Cn_{ij} = M_c / C_{max} = \; [C_{ij} / C_{max}] \quad (10)$$

Normalized time matrix,

$$N_t = Tn_{ij} = M_t/T_{max} = [T_{ij}/T_{max}] \quad (11)$$

Normalized quality matrix,

$$N_q = Qn_{ij} = M_q/Q_{max} = [Q_{ij}/Q_{max}] \quad (12)$$

3.2 The details of the proposed algorithm

The basic structure of the proposed GA algorithm to solve MOAP is as follows:

Step 1. Create an initial population of *m* chromosomes where $\frac{m}{2}$ is an even number (generation 0).
Step 2. Evaluate the fitness of each chromosome.
Step 3. Select parents from the current population via *proportional selection* (*i.e.* the selection probability is proportional to the fitness). The number of total parents is $\frac{m}{2}$.
Step 4. Choose at random a pair of parents for mating and apply partially mapped crossover (PMX) to create two offspring. A parent is chosen for mating for one time.
Step 5. Apply mutation operator to offsprings, and insert the resulting offsprings in the new population with their parents.
Step 6. Repeat steps 5 and 6 until all parents are selected and mated. (*i.e.* offspring are created).
Step 7. Find the best chromosome from all the parents and child of the new population. Replace the 'best chromosome so far' by the best chromosome of the new population when the later one is superior.
Step 8. Replace the old population of chromosomes by the new one.
Step 9. Go back to step 2 if the last generation does not provide a better solution for several iterations. Otherwise, the final solution is the 'best chromosome so far' created during the search.

In the forthcoming section, we will describe the details of the implementation of GA in solving MOAP.

3.2.1 Representation of solution

The representation of the solution structure of the MOAP is discussed here. Symbolic ordered gene (*i.e.* the value of two alleles can't be same (Triantaphyllou et al. 1998)) strings of length *n* (total number of tasks) are used to represent solution (chromosome), henceforth called tasks chromosome. The chromosome has one allele for each task. The position of an allele in task chromosome represent the task number while the allele value is the worker number who is assigned to that task (Sehrawat and Singh 2011). For example, let a string consists of genes (4, 3, 1, 5, 2). The allele at the first locus of the string signifies that the worker number 4 is assigned to task number 1, worker number 3 is assigned to task number 2, and so on.

3.2.2 Initial population

The encoded solution is represented as a chromosome. The initial set of solutions *i.e.* the population size of *m* is generated randomly where $\frac{m}{2}$ is an even number, allowing the entire range of possible solutions (the *search space*).
For a particular solution *i.e.* chromosome, the genes,

$$b_j = i \quad where\ i^{th}\ worker\ is\ assigned\ to\ j^{th}\ task \quad for \quad j = 1, 2, \ldots\ldots, n \quad (13)$$

And the chromosome *r*,

$$chrm_r = b_j \quad for\ r = 1, 2, \ldots\ldots, m\ and\ j = 1, 2, \ldots\ldots, n \quad (14)$$

3.2.3 Fitness function

The fitness function works as an objective function which is needed to be maximized. In this study, the function F_r is used to form fitness functions for chromosome number *r*. First, the total normalized cost, the total normalized time, and the total normalized quality are determined. It is often given different priority on cost, time and quality. The Fitness of a chromosome depends on this priority (weight). The W_c, W_t and W_q are weights of the cost, time, and

quality respectively. Since the MOAP is a minimization problem, the high fitness value is associated with minimized cost, minimized time, and minimized quality value (as the highest quality corresponds to the lowest value). It is to make the fitness choice criteria maximum, thus the inverse of sum product of a priority and corresponding total normalized values are taken in the equation (Mota et al. 2015).

For a particular chromosome *i.e.* solution,

The total normalized cost,

$$Tot_c = \sum_{j=1}^{n} Cn_{b_j\ j} \quad (15)$$

The total normalized time,

$$Tot_t = \sum_{j=1}^{n} Tn_{b_j\ j} \quad (16)$$

The total normalized quality,

$$Tot_q = \sum_{j=1}^{n} Qn_{b_j\ j} \quad (17)$$

The fitness of chromosome,

$$F_r = 1/\ (W_c\ *\ Tot_c + W_t\ * Tot_t\ + W_q\ *\ Tot_q) \quad (18)$$

3.2.4 Reproduction

In the present implementation, the *proportional selection* (*i.e.* the selection probability is proportional to the fitness) is used (Uddin and Shanker 2002). The expected number of chromosomes going from the parent generation to mating pool depends on the individual fitness values (Blickle and Thiele 1995). The probability of selection for chromosome *r* is

$$pselect_r = \frac{F_r}{\sum_{r=1}^{m} F_r} \quad (19)$$

3.2.5 Crossover

Every chromosome is an ordered list of the workers, so the direct swap is not possible. Partially Matched Crossover (PMX) and cycle crossover (CX) are widely used for the crossover of ordered chromosomes (Razali and Geraghty 2011). PMX and CX are not really competitive with the order-preserving crossover operators (Soni and Kumar 2014). Partially Matched Crossover (PMX) which was initially developed for tackling the "Travelling Salesman Problem", is chosen as the crossover operator in this model. The crossover in the proposed methodology is explained below.

Each individual in the mating pool has the same chance of being parent independent of its fitness. Two parent chromosomes from the mating pool are chosen randomly. Crossover occurs between these two parents. The locus of the cross-over points is generated randomly. For example, it is to crossover between,
$chrm_1$ = (1 8 2 4 7 6 5 3) and
$chrm_2$ = (2 7 5 3 1 6 8 4).

Two random number is generated between 1 and *L* (*L=7* in this case). Let it '3' and '5'. The locus of the crossover point is shown by 'dot' before position '3' and after position '5'.
$chrm_1$ = (1 8 . 2 4 7 . 6 5 3)
$chrm_2$ = (2 7 . 5 3 1 . 6 8 4)

Now the portion between the selected crossover points is swapped and the rest of the values are changed according to the PMX rule (Umbarkar and Sheth, 2015). After exchanging the information, the two offspring are,
$chrm_1$' = (7 8 . 5 3 1 . 6 2 4)
$chrm_2$' = (5 1 . 2 4 7 . 6 8 3)

The resulting two chromosomes, called the offspring, added to the population with their parents. The offspring cannot be chosen for crossover until the next generation. The process is repeated until the mating pool is not empty, where a parent in the mating pool take part in crossover for only one time.

3.2.6 Mutation

This mutation operator is the closest in philosophy to the biological mutation operator because it only slightly modifies the original chromosome (Potvin 1996). In this accomplishment, we have done the two alleles swapping for each chromosome, in offspring, with the probability of mutation, pm. For illustration, let us consider the chromosome, from the previous example,
$chrm_1$'= (7 8 5 3 1 6 2 4)
Suppose the locus chosen for mutation is 2 and 5. Then, after mutation, the new chromosome (offspring) will be,
$chrm_1$''= (7 1 5 3 8 6 2 4)

3.2.7 Termination

When there is no improvement of the highest fitness value attained so far, in successive five generations, it stops creating a new generation. And the chromosome having the highest fitness in all the generations is taken as a solution of the MOAP.

3.2.8 Extracting the values of decision variables from the best chromosome

Chromosomes *i.e.* solution is made of genes, like r^{th} chromosome,

$$chrm_r = b_j \quad for\ j = 1, 2, \ldots\ldots, n \quad (here\ b_j\ is\ the\ value\ of\ j^{th}\ gene) \tag{20}$$

Now, we will set the values of decision variables according to the chromosome, as follows.
For $i = 1, 2, \ldots\ldots, n$ *and* $j = 1, 2, \ldots\ldots, n$

$$X_{ij} = \begin{cases} 1 & for\ i = b_j \\ 0 & Otherwise \end{cases} \tag{21}$$

4. Numerical illustration and result analysis

In the present work, the genetic algorithm was coded in ANSI C programming language using simple array data structure and ran on a PC (core i3, Intel processor of 2.2GHz). Here, we considered an example of 6 workers and 6 tasks as in Tsai et al. (1999) as a numerical example where all the criteria needed to be minimized (Table 2).

Table 2. A numerical example of a MOAP

Criteria	Worker, i	Task, j					
		1	**2**	**3**	**4**	**5**	**6**
Cost, C_{ij}	**1**	6	3	5	8	10	6
	2	6	4	6	5	9	8
	3	11	7	4	8	3	2
	4	9	10	8	6	10	4
	5	4	6	7	9	8	7
	6	3	5	11	10	12	8
Time, T_{ij}	**1**	4	20	9	3	8	9
	2	6	18	8	7	17	8
	3	2	8	20	7	15	7
	4	12	13	14	6	9	10
	5	9	8	7	14	5	9
	6	17	13	3	4	13	7
Quality, Q_{ij}	**1**	1	3	1	1	1	5
	2	3	5	3	5	7	5
	3	1	7	5	3	5	7
	4	5	9	3	5	7	3
	5	3	9	7	5	3	3
	6	3	3	5	7	5	7

Using the current approach, with the equal priority of cost, time and quality the solution is following which corresponds to the total cost of 42 units, the total time of 41 units and the total quality of 14 units. Some parameters of this solution have been shown in table 3.

$$x_{14} = x_{23} = x_{31} = x_{46} = x_{55} = x_{62} = 1 \quad (22)$$

Table 3. Various parameter value in the solution

Trial No.	Population size, *m*	CPU Time to get the best solution (second)	No. of iteration required to get the best solution	The best fitness value attained so far
1	24	0.016	17	0.0140735
2	40	0.016	22	0.0140735
3	60	0.016	15	0.0140735
4	100	0.016	4	0.0140735
5	200	0.016	7	0.0140735
6	400	0.016	3	0.0140735

Furthermore, the convergence of 'best fitness value so far' and 'average fitness value' has been illustrated in figure 1. Where we get the optimum result at the generation number 17 for the population size of 24.

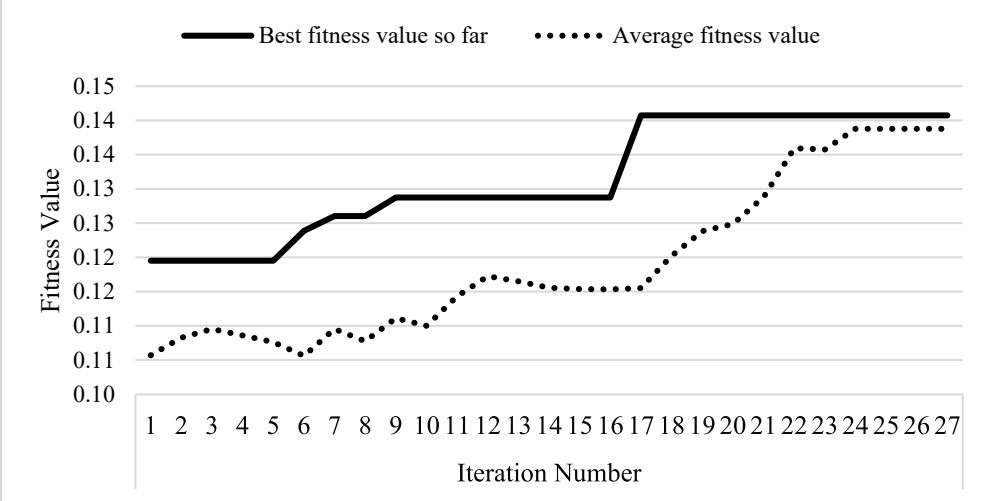

Figure 1. Convergence to the global maximum fitness for population size 24

5. Comparison of the present work

To further justify the proposed approach, the results of the developed methodology has been compared with the experimental result produced by the approach of (Bao et al., 2007) and the multi-objective fuzzy deployment methodology developed by (Tsai et al., 1999) as shown in Figure 2. Additionally, the improvement by proposed methodology relative to Bao et al. and Tsai et al. are shown in Figure 3.

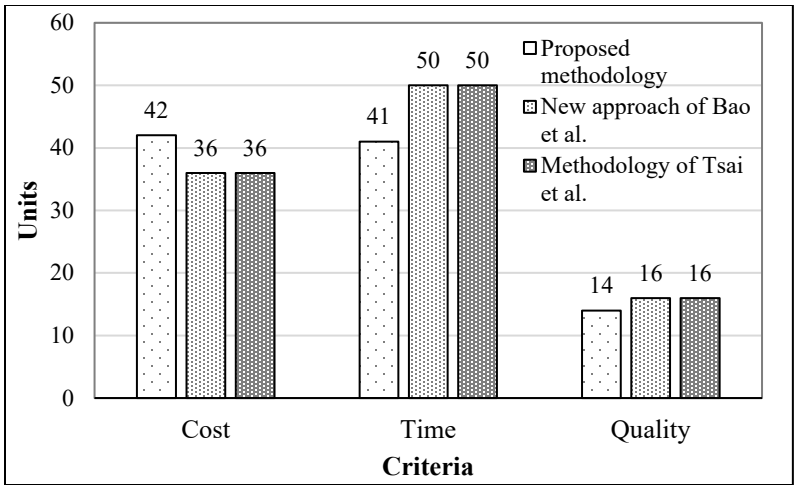

Figure 2. Results in different methodologies

The proposed methodology results in 42 units operation cost, 41 units operation time and 14 units quality. Which imply the amount of 18% improvement in time and 13% improvement in quality with an expense of 17% cost deterioration. Consequently, the result of the proposed methodology is better since the priorities of all the criteria is the same.

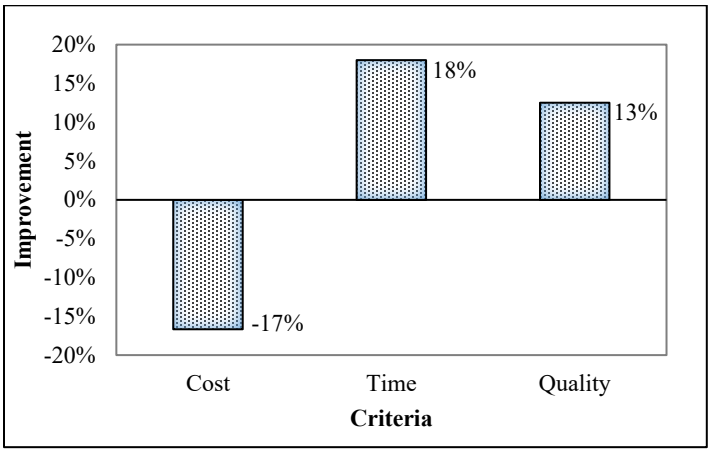

Figure 3. Improvement by proposed methodology relative to Bao et al. and Tsai et al.

The approach of Bao et al. cannot normalize quality criterion in an interval of [0,1] when there is no assignment having the quality weight of '1'. In this case, the reciprocal of quality distributes as normalized quality in an interval [0,1) while the other normalized criteria are distributed in an interval of [0,1]. However, all the criteria are distributed into a normalized value in an interval of [0,1] in the present work.

6. Conclusion

In the present paper, a methodology to solve the multi-objective assignment problem has been proposed and solved by a genetic algorithm. It is found that the algorithm is very effective to find the global optimal solution quickly. A great feature of this work is its simple calculation procedure compared to the other methods. As a whole, the proposed methodology doesn't require careful attention to the determinations of the weight among the resources. Moreover, it incorporates the priority of the resources in the decision making process.

References

Acar and Aplak, A Model Proposal for a Multi-Objective and Multi-Criteria Vehicle Assignment Problem: An Application for a Security Organization, *Mathematical and Computational Applications*, vol. 21, pp. 39-60, 2010.

Bao, Tsai and Tsai, A New Approach to Study the Multi-Objective Assignment Problem, *An Interdisciplinary Journal*, vol. 53, pp. 123-132, 2007.

Blickle and Thiele, A Comparison of Selection Schemes Used in Genetic Algorithms, *TIK Report*, vol. Nr-11, 1995.

Boussaïd, Lepagnot and Siarry, A survey on optimization metaheuristics, *Information Sciences, Elsevier,* vol. 237, pp. 82-117, 2013.

Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, 1st edition, John Wiley & Sons, New Delhi, 2001.

Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning, Reading*, 1st edition, Addison-Wesley Publishing Company, Inc., Boston, 1989.

Ishizuka, M. and Matsuo, SL method for computing a near-optimal solution using linear and nonlinear programming in cost based hypothetical reasoning, *Knowledge based system, Elsevier*, vol. 15, pp. 369-376, 2002.

Khun, The Hungarian method for assignment problem, *Naval Research Logistics*, vol. 2, no. Quarterly, pp. 83-97, 1955.

Lin, Solving the Transportation Problem with Fuzzy Coefficients using Genetic Algorithms, *Proceedings of the IEEE International Conference on Fuzzy Systems*, Korea, Aug 20-24, 2009.

Melanie, *An Introduction to Genetic Algorithms*, 5th edition, MIT Press, Boston, 1999.

Mota, Miguel, Mota, Serrano and Daniel, *Applied Simulation and Optimization*, 1st edition, Springer International Publishing, Switzerland, 2015.

Oliveira and Saramago, Multiobjective optimization techniques applied to engineering problems, *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 32, pp. 94-105, 2010.

BERTSEKAS, A New Algorithm for the Assignment Problem, *Mathematical Programming,* North-Holland Publishing Company, vol. 21, pp. 152-171, 1981.

Potvin, Genetic algorithms for the travelling salesman problem, *Annals of Operations Research,* vol. 63, pp. 339-370, 1996.

Pramanik and Biswas, Multi-objective Assignment Problem with Generalized Trapezoidal Fuzzy Numbers, *International Journal of Applied Information System,* vol. 2, pp. 13-20, 2012.

Razali and Geraghty, Genetic Algorithm Performance with Different Selection Strategies in Solving TSP, *Proceedings of the The World Congress on Engineering,* London, July 6-8, 2011.

Sahu and Thapadar, Solving the Assignment problem using Genetic Algorithm and Simulated Annealing, *International Journal of Applied Mathematics1,* vol. 36, 2017.

Sehrawat, Ms. and Singh, Mr., Modified Order Crossover (OX) Operator, *International Journal on Computer Science and Engineering,* vol. 3, pp. 2019-2023, 2011.

Sharma, Kumar, Dr. and Tyagi, Dr., A Review of Genetic Algorithm and Mendelian Law, *International Journal of Scientific & Engineering Research,* vol. 7, pp. 488-499, 2016.

Soni and Kumar, Dr., Study of Various Crossover Operators in Genetic Algorithms, International Journal of Computer Science and Information Technologies, *International Journal of Computer Science and Information Technologies,* vol. 5, pp. 7235-7238, 2014.

Taha, *Operation Research: An Introduction*, 8th edition, New Delhi, 2006.

Tailor and Dhodiya, Genetic Algorithm Based Hybrid Approach to Solve Multi-Objective Assignment Problem, *International Journal of Innovative Research in Science, Engineering and Technology,* vol. 5, pp. 524-535, 2016.

Triantaphyllou, E., Shu, B., Sanchez, S. and Ray, T., Multi-Criteria Decision Making: An Operations Research Approach, *John Wiley & Sons*, vol. 15, pp. 175-186, 1998.

Tsai, Wei and Cheng, Multiobjective fuzzy deployment of manpower, *International Journal of the Computer, the Internet and Management,* vol. 7, pp. 1-7, 1999.

Uddin and Shanker, Grouping of parts and machines in presence of alternative process routes by genetic algorithm, *International Journal of Production Economics,* vol. 76, pp. 219-228, 2002.

Umbarkar, A.J. and Sheth, P.D., Crossover Operators in Genetic Algorithms: A Review, *Ictact Journal on Soft Computing,* vol. 6, pp. 1083-1092, 2015.

Yeniay, Penalty Function Methods for Constrained Optimization with Genetic Algorithms, *Mathematical and Computational Applications,* vol. 10, pp. 45-56, 2005.

Biographies

Md. Mahbubur Rahman is an Assistant Professor of Department of Industrial Engineering and Management, Khulna University of Engineering & Technology, Khulna, Bangladesh. He received his B.Sc. Eng. in Industrial & Production Engineering degree from Department of Industrial Engineering and Management, Khulna University of Engineering & Technology. His main research areas are optimization, computer-aided manufacturing, and soft computing. He has published five journals paper.

Md. Kutub Uddin is a Professor of Department of Mechanical Engineering, Khulna University of Engineering & Technology, Khulna, Bangladesh. He received his B.Sc. Eng. in Mechanical Engineering degree from Department of Mechanical Engineering, Bangladesh University of Engineering & Technology, Dhaka, Bangladesh. He received his M. Eng. degree from Asian Institute of Technology (AIT), Bangkok, Thailand. He received his Doctor of Philosophy (Ph. D.) degree from Indian Institute of Technology (IIT), Kanpur, India. His main research areas are warehouse management and optimization.