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,

$$Tot_{c} = \sum_{j=1}^{n} Cn_{b_j \ j} \tag{15}$$

The total normalized time,

$$Tot_{t} = \sum_{j=1}^{n} Tn_{b_j \ j} \tag{16}$$

The total normalized quality,

$$Tot_{q} = \sum_{j=1}^{n} Qn_{b_j \ j} \tag{17}$$

The fitness of chromosome,

$$F_r = 1/\left(W_c * Tot_c + W_t * Tot_t + W_q * Tot_q\right) \tag{18}$$

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

$$pselect_r = \frac{F_r}{\sum_{r=1}^{m} F_r} \tag{19}$$

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.

$$x_{14} = x_{23} = x_{31} = x_{46} = x_{55} = x_{62} = 1 \tag{22}$$

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.

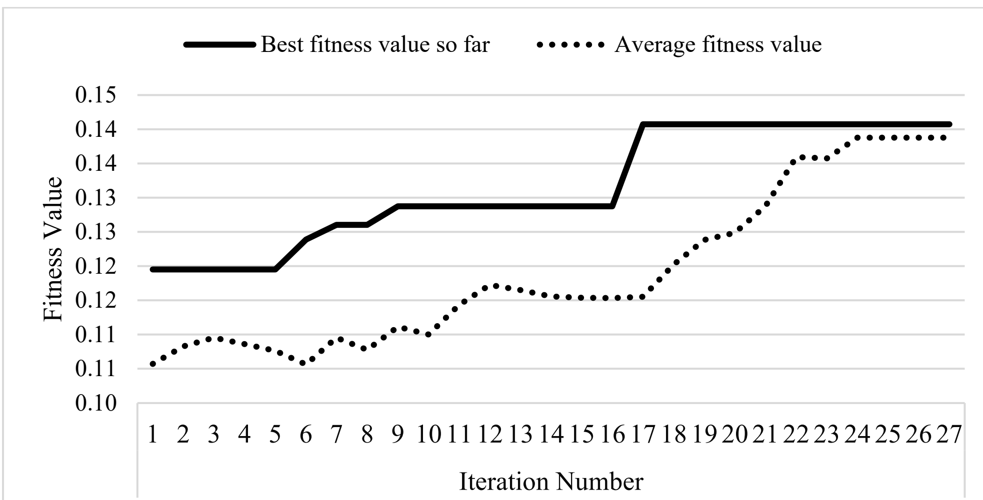

Figure 1. Convergence to the global maximum fitness for population size 24

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.

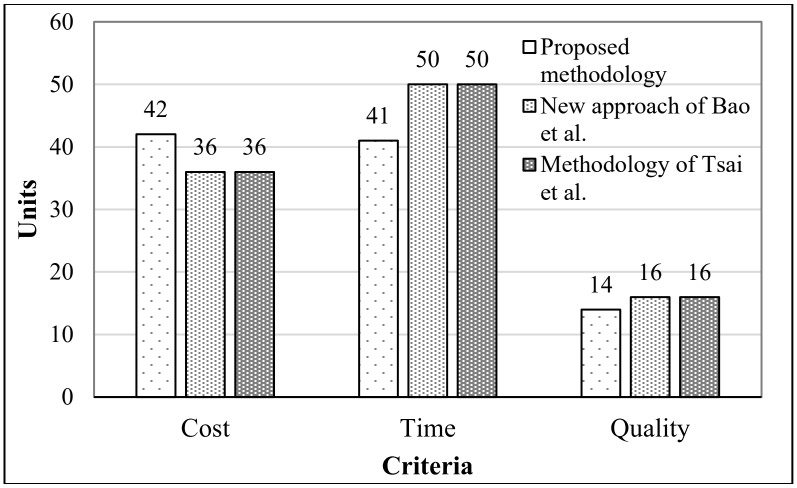

Figure 2. Results in different methodologies

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.

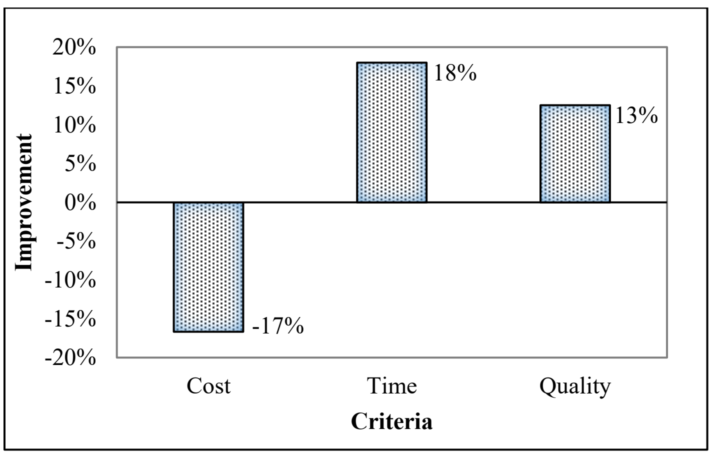

Figure 3. Improvement by proposed methodology relative to Bao et al. and Tsai et al.

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.