Volume 3, Number 4, 2014

International Journal of Research in Industrial Engineering

www.nvlscience.com

A Novel Genetic Algorithm for a Flow Shop Scheduling Problem with Fuzzy Processing Time

N. Shahsavari pour[1], M. H. Abolhasani Ashkezari[2,*], H. Sheikhi[2], H. Mohammadi Andargoli[2], H. Abolhasani Ashkezari[3]

[1]Department of Industrial Management, Vali-e-Asr University, Rafsanjan, Iran.
[2]Department of Industrial Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran.
[3]Department of Mechanical Engineering, University of Birjand, Birjand, Iran.

ARTICLE INFO

Article history :
Received:
June 04, 2013
Revised:
June 25, 2014
Accepted:
July 06, 2014
Available online:
January 23, 2016

Keywords:

Enterprise Resource Planning selection, Three-parameter interval grey numbers, AHP, Shannon Entropy, Three-parameter grey interval incidence degree

ABSTRACT

Various procedures, methods, constraints and objectives are studied in a flow shop problem during the past decades. In order to adapt the problem to the reality form, its parameters are considered as a fuzzy model. In this problem, we consider the processing time as the trapezoidal fuzzy numbers. The purpose of this problem is to find an optimum sequence in a way that the makespan or the completing time of jobs to be minimized. In order to solve this problem, in this paper, the Random-Elitist Genetic Algorithm (REGA) is presented in this regard. Observing the performance and the efficiency of this algorithm, we code it by the VBA and compare with the other results. We first test the performance of different crossover operators for our algorithm. Next, using a specific example, we examine the performance of our algorithm. The results indicated that due to very good searching; this algorithm has the good performance in finding the optimal solution and reaching the optimum solution in a very short time.

1. Introduction

The flow shop included n number of jobs must be done in m machines with the same sequences. This problem has been presented for the first time by Johnson [1]. Within the current years, many researches have been done on this problem. Ogbu and Smith [2] and Van Laarhoven and Aarts [3] used Simulated Annealing (SA) to solve this problem. The genetic algorithm is the other one using to solve this problem [4 and 5]. The original principles of the Genetic Algorithm (GA) have been presented by Holland [6] and his colleagues in the University of Michigan.

*Corresponding author
E-mail address: abolhasani.hossein@yahoo.com

Please cite this article in press as: N. Shahsavari pour, M. H. Abolhasani Ashkezari, H. Sheikhi, H. Mohammadi Andargoli, H. A. Abolhasani Ashkezari, A Novel Genetic Algorithm for a Flow Shop Scheduling Problem with Fuzzy Processing Time, Int. J. of Research in Industrial Engineering (2014)

The investigation to develop the mathematical frame of this algorithm and its application was continued simultaneously after the rudimentary report resulted in publication of a book by Holland in 1975 [6]. Whereas many parameters are not exact in reality they must be considered in a fuzzy state, because of the influences of many environmental factors, the hardness and impossibility of determining the exact form. The problem has been considered as a fuzzy state by many articles up to this moment. Mccahon and Stanley [7] and Tsujimura et al. [8] presented a method for flow shop scheduling problem with fuzzy processing time. KhademiZare and Fakhrzad [9] presented a hybrid genetic algorithm to solve flexible flow-shop problems using fuzzy approach, their goal is to minimize the total job tardiness. Sadinezhad and Ghaleh Assadi [10] developed a novel fuzzy CDS algorithm in fuzzy flow shop scheduling by applying the preference ratio concept. Lai and Wu [11] applied the virus-evolutionary genetic algorithms to search for the best schedules.

In this paper, we have presented an algorithm developed from genetic algorithms. The REGA with the addition of the concept of Elitism and with more attention to better answers at each step gives them a greater chance to be selected. And, with the concept of Random, to avoid falling in local optimum. To show the efficiency of this algorithm, we solve the problem of a fuzzy flow shop. Comparisons indicate this algorithm has a great ability in finding the optimal solution in a very short time.

Here is presented a summary of this article: A genetic algorithm is described in the second part of this article. Application of fuzzy concept and implementing it with the flow shop problem, and the method used to compare fuzzy numbers is presented in third section. In the fourth section, we explain the proposed REGA algorithm. Results of the coding algorithm by using VBA, to measure its effectiveness have come in the fifth section. At the end of this article, we have concluded from the previous sections.

2. Ranking of Fuzzy Numbers

In this section we described implementation of fuzzy logic in flow shop problem. For more practical flow shop problem in the real world, we assume the processing time of jobs on the machines as fuzzy number. For this purpose, we have assumed processing time of job *i* on the machine *j* as a trapezoidal fuzzy number (*a*,*b*,*c*,*d*:*w*). Figure 1 shows how to display a trapezoidal fuzzy number:

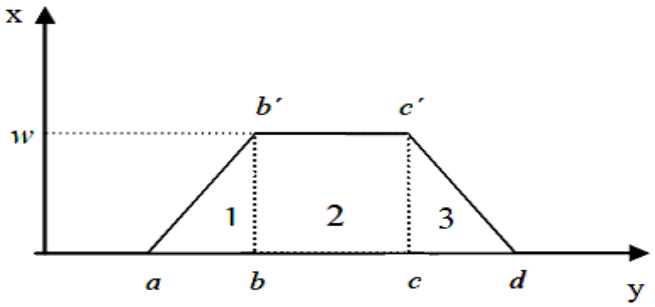

FIGURE 1. Trapezoidal fuzzy number (*a*,*b*,*c*,*d*:*w*)

Please cite this article in press as: N. Shahsavari pour, M. H. Abolhasani Ashkezari, H. Sheikhi, H. Mohammadi Andargoli, H. A. Abolhasani Ashkezari, A Novel Genetic Algorithm for a Flow Shop Scheduling Problem with Fuzzy Processing Time, Int. J. of Research in Industrial Engineering (2014)

We use "The radius of gyration of an area", in order to convert the fuzzy number (*a*,*b*,*c*,*d*:*w*) into crisp number S(A) presented by Deng et al. [12]. In Figure 2, we consider an area A, and the element of area dA of coordinates x and y.

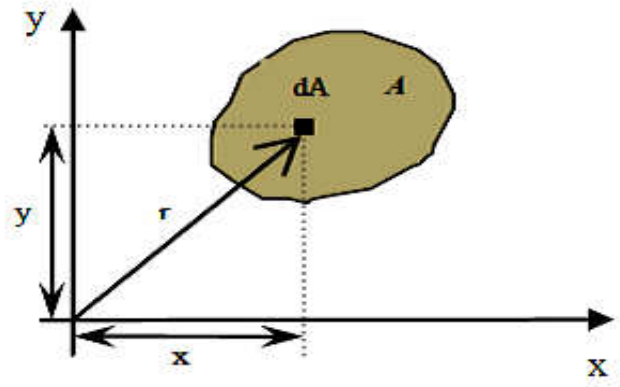

FIGURE 2. Moment of inertia of an area

The moment of inertia of the area A with respect to the *x* axis and *y* axis are defined as formulas (1) and (2):

$$I_x = \int_A y^2 \ dA \tag{1}$$

$$I_y = \int_A x^2 \ dA \tag{2}$$

In formulas (3) and (4), the radius of gyration of an area A with respect to the $x(r_x)$ axis and $y(r_y)$ are shown. The radius of gyration points of the fuzzy number A is denoted as $(r_x(A), r_y(A))$.

$$I_x = r_x^2 A \quad \Rightarrow \quad r_x = \sqrt{\frac{I_x}{A}} \tag{3}$$

$$I_y = r_y^2 A \quad \Rightarrow \quad r_y = \sqrt{\frac{I_y}{A}} \tag{4}$$

In Figure 1, the moment of inertia of generalized trapezoidal fuzzy number can be calculated as formulas (5) and (6):

$$(I_x) = (I_x)_1 + (I_x)_2 + (I_x)_3 \tag{5}$$

$$(I_y) = (I_y)_1 + (I_y)_2 + (I_y)_3 \tag{6}$$

where,

$$(I_x)_1 = \int_{abb'} y^2 dA = \int_0^w y^2 \frac{(b-a)(w-y)}{w} dy = \frac{(b-a)w^3}{12} \tag{7}$$

Please cite this article in press as: N. Shahsavari pour, M. H. Abolhasani Ashkezari, H. Sheikhi, H. Mohammadi Andargoli, H. A. Abolhasani Ashkezari, A Novel Genetic Algorithm for a Flow Shop Scheduling Problem with Fuzzy Processing Time, Int. J. of Research in Industrial Engineering (2014)

$$(I_y)_1 = \int_{abb'} x^2 dA = \frac{(b-a)^3 w}{4} + \frac{(b-a)a^2 w}{2} + \frac{2(b-a)^2 aw}{3} \quad (8)$$

And, we can calculate $(I_x)_2, (I_y)_2, (I_x)_3, (I_y)_3$ same (7) and (8):

$$(I_x)_2 = \frac{(c-b)w^3}{3} \quad (9)$$

$$(I_x)_3 = \frac{(d-c)w^3}{12} \quad (10)$$

$$(I_y)_2 = \frac{(c-b)^3 w}{3} + (c-b)b^2 w + (c-b)^2 bw \quad (11)$$

$$(I_y)_3 = \frac{(d-c)^3 w}{12} + \frac{(d-c)c^2 w}{2} + \frac{(d-c)^2 cw}{3} \quad (12)$$

So, the radius of gyration of an area *A* calculated from (13) and (14):

$$r_x(A) = \sqrt{\frac{(I_x)_1 + (I_x)_2 + (I_x)_3}{(((c-b)+(d-a))w)/2}} \quad (13)$$

$$r_y(A) = \sqrt{\frac{(I_y)_1 + (I_y)_2 + (I_y)_3}{(((c-b)+(d-a))w)/2}} \quad (14)$$

And finally, we use formula (15) for ranking of fuzzy numbers:

$$S(A) = r_x(A) \times r_y(A) \quad (15)$$

3. Genetic Algorithm

Charles Darwin's theory of evolution which was presented in 1859 [13], assigned a special place in a case of optimizing problems. The genetic algorithm can be called simply a browser method based on the observation of offspring characteristics related to the sequence generations and children selection on the basis of the principle of survival of the fittest. The genetic algorithm for the offspring of a generation (the problem solution in a process) borrowed the rules of genetic science applying them in order to produce offspring with better characteristic (the closer solutions to the target of problem). In each generation by means of selection process proportional to value solution and the new offspring (solution) procreation better approximation derived from final solution. This process makes the generation being more to do with the problem condition. During the past decades, the genetic algorithm applied extensively in a wide area of sciences, trading and engineering as some browser and optimizing tools. The main successful reason using these algorithms much better than others is their extensive applicability, readiness for using, the simplicity and their general prognosis [4].

Please cite this article in press as: N. Shahsavari pour, M. H. Abolhasani Ashkezari, H. Sheikhi, H. Mohammadi Andargoli, H. A. Abolhasani Ashkezari, A Novel Genetic Algorithm for a Flow Shop Scheduling Problem with Fuzzy Processing Time, Int. J. of Research in Industrial Engineering (2014)

3.1. Explanation of the REGA

In this section, we explain our REGA. In this algorithm, a genetic algorithm is developed, which it has two special attentions. The first aspect of this algorithm is that there is talk of being elitist. In genetic algorithm, a problem may arise during its implementation and that, a better solution than the previous solution is found. But because the choices of new parents are selected randomly, it is possible that the good answers, not choice, and eliminated. In this algorithm, using the concept of elitism, from removal of the top answers would prevent each generation. In every generation, a percentage of the top answers are entered directly into the roulette wheel, and with this, we have chosen this solution with a higher probability.

The second issue that we apply our own algorithm, to avoid falling in local optimum. For this purpose, in every generation, we create a number of "random permutation" and this answers, are added to the solution in selection process. With this action, the chance to give these answers, to be selected in a roulette wheel and we have also studied a larger solution space, and we reduce the probability of falling in local optimum. We have to decrease the random rate in proportional to the number of produced generations as follow:

$$R_m = R'_m - \alpha \times \frac{t}{T} \tag{16}$$

t is a symbol applied to indicate the number of produced generations until then, T shows the maximum number of generations, R_m indicates the random rate in each generation, R'_m specifies the fixed and constant value used to show the maximum possibility of random rate and α is a parameter which defines the random rate ratio. It makes the random rate to be applied with great potential (most likely to be applied) and the searching spaces to be extended too. Pseudo-code of proposed REGA algorithm is presented in the following:

1. Specify the initial parameters (*M*, *N*, Population number, Generation number, Crossover rate, Mutation rate, Elitist rate, Random rate, α, *K*)
2. Calculate Elitist number (Elitist number = Elitist rate * population number)
3. Determine the fuzzy processing time for each job on each machine
4. Determine the initial population
5. Calculate the fitness for each sequence
6. Roulette wheel
7. Crossover (1-PMX 2-CX 3-OX 4-Uniform)
8. Mutation
9. Calculate the number of random number (Random number = Random rate * Population number)
10. Creating the sequences of random (At this stage in order to creating new sequences, randomly generate some new sequences)
11. Collection of the total population (Total population = Parent population + Crossover population + Mutation population + Random population)
12. Calculate the fitness for total population
13. Select the new population (New population = Elitist population + Total population)

Please cite this article in press as: N. Shahsavari pour, M. H. Abolhasani Ashkezari, H. Sheikhi, H. Mohammadi Andargoli, H. A. Abolhasani Ashkezari, A Novel Genetic Algorithm for a Flow Shop Scheduling Problem with Fuzzy Processing Time, Int. J. of Research in Industrial Engineering (2014)

14. Reduce the random rate (Random rate = Random rate – (α * t/T))
15. Reduce the mutation rate (Mutation rate = Mutation rate – (K * t/T))
16. If End condition is reached, End program,
    Else, GO to5.

3.2. Elitism

The elitism in single-purpose genetic algorithm is to copy a number of best solutions to the next generation with no change, observing in each ones population. Using the elitism leads to no decrease in the fitness of the best chromosomes in the sequence generations. The elitism is effective in convergence to global optimum. The Genetic implement, GENITOR, presented by Whitley [14] is an elitist genetic algorithm in which each offspring enters to parent's population. The produced offspring replaces by the worst members of population. The CHC algorithm [15] is another form of the elitist algorithm which selects N better solutions from the $2N$ member of population composed of parents and offspring. The NSGA-II and SPEA methods have been presented by Deb, et al. [16], and Zitzler and Thiele [17] respectively.

As mentioned before there are various procedures to transmit a number of superior chromosomes to the next generation. A chromosome can be transmitted from each generation to the next ones as a best chromosome in this regard. Practically, there is no time for it to combine with other chromosomes creating some suitable solutions. In another procedure, it is possible to transmit the whole superior solutions to the next generations which lead to decrease in a number of new solutions being produced within the evolutionary process. This operation, practically, slows the access to optimum solutions makes them trap into the local optimum. The proper procedure we have used in this article is a facet between these two strategies leading to P (p=Elitist rate) percent transmission from superior chromosomes to the next generation. P parameter indicates the selection pressure of elitist operator.

3.3. Selection

The chromosomes which are selected by initial population to reproduction called parents. The selection is a process applied to choose each parent of population in order to crossover operation. The purpose of this selection is to choose better parents leads to produce offspring with high propriety. The selection specifies how to select the parents. On the basis of "Darwin evolutionary theory", the best ones must have more chance to reproduction and survival. "The pressure of selection" procedure meaning the attention degree toward the better parent selection defined by Goldberg and Deb in 1991 [18]. The high selection pressure leads to more emphasis on the parent selection in order to choose ones with more worthiness or qualification. About the various selection methods have been discussed in Goldberg and Deb [18], Back [19] books. In our algorithm, we use the Roulette Wheel to select parents.

3.4. Crossover

When two children belong to the members of one generation are chose on the base of their proprieties within the selection process, they will be allowed to reproduction and producing

Please cite this article in press as: N. Shahsavari pour, M. H. Abolhasani Ashkezari, H. Sheikhi, H. Mohammadi Andargoli, H. A. Abolhasani Ashkezari, A Novel Genetic Algorithm for a Flow Shop Scheduling Problem with Fuzzy Processing Time, Int. J. of Research in Industrial Engineering (2014)

new children. The connection between parents and the production of next generation is done by means of crossover operator. We have used four different crossovers in this respect. The PMX, OX, CX methods which has been presented by Goldberg [4] been compared with each other in Oliver et al. [20] book. Another method is Uniform crossover presented in Syswerda article [21].

3.5. Mutation

Mutation, in the nature, is a process in which a part of gene changes randomly. Browsing to find the better solutions among the current ones, the crossover operator has been applied in this regard. But the mutation operator considered as an assistant to help the searching process related to solution space. The population varieties will be maintained by mutation and a new genetic structure produced in population by means of some changes taking place in some of the genes randomly. Maintaining the population varieties, the mutation operator prevents algorithm from falling in the local optimum. According to Goldberg [4] in each generation, we have to decrease the possibility of mutation in proportional to the number of produced generations as formula (17):

$$P_m = P'_m - K_{pm} \times \frac{t}{T} \tag{17}$$

P_m Indicates the mutation rate in each generation P'_m specifies the fixed and constant value used to show the maximum rate of mutation and K_{pm} is a parameter that defines the ratio of mutation rate. It makes the mutation to be applied with great potential (most likely to be applied) and the searching spaces to be extended too. The mutation rate, then, decreases and searching process centralized in some part of the finding results. The convergence speed increases contrary to mutation rate. Michalewics [22] has presented the other mutation methods. For the mutation operation, we have selected the interchanging method in our algorithm. As such that two genes selected randomly substituted for each other, if two changes are going to do in this respect (FIGURE 3):

FIGURE 3. Mutation operation

4. Computational Results

In this paper, in order to perform various experiments, and comparison with results from other papers and to measure the efficiency of the REGA algorithm, we coded it in VBA. We run our algorithm in Intel T6570, 2.1 GHz with 2 GB RAM. In Example 1, we now examine the four crossover operators presented in the above. To solve this problem, we use the methods of PMX, OX, CX and Uniform for a crossover operator. In order to compare the performance of these operators, we use the example mentioned in Kalczynski and Kamburowski [23]. In the example mentioned in their article a flow shop problem has come in crisp state. This example consists of six jobs, which is performed on the three machines.

Please cite this article in press as: N. Shahsavari pour, M. H. Abolhasani Ashkezari, H. Sheikhi, H. Mohammadi Andargoli, H. A. Abolhasani Ashkezari, A Novel Genetic Algorithm for a Flow Shop Scheduling Problem with Fuzzy Processing Time, Int. J. of Research in Industrial Engineering (2014)

Optimal sequence unique for this problem is (3, 5, 4, 6, 2, 1) which has the makespan=425. In Table 1, processing time of jobs on the machines is given.

TABLE 1. Processing time of jobs on machines

Jobs	Machine 1	Machine 2	Machine 3
J1	19	46	65
J2	44	63	12
J3	85	56	98
J4	59	68	25
J5	87	66	53
J6	51	4	63

In this example, in order to be more visible the impact of crossover operator, we assumed Random rate=0 and Elitist rate=0.05. For each crossover operator, we assumed crossover rate[0.05, 1] and for each rate, we repeated the experiment 20 times, and we consider their average as output. The results show that the crossover rate between [0.4, 0.6] creates better result. Also The OX operator creates better answers from other operators. We have used this operator for the next experiments. Results of this example are shown in Figure 4.

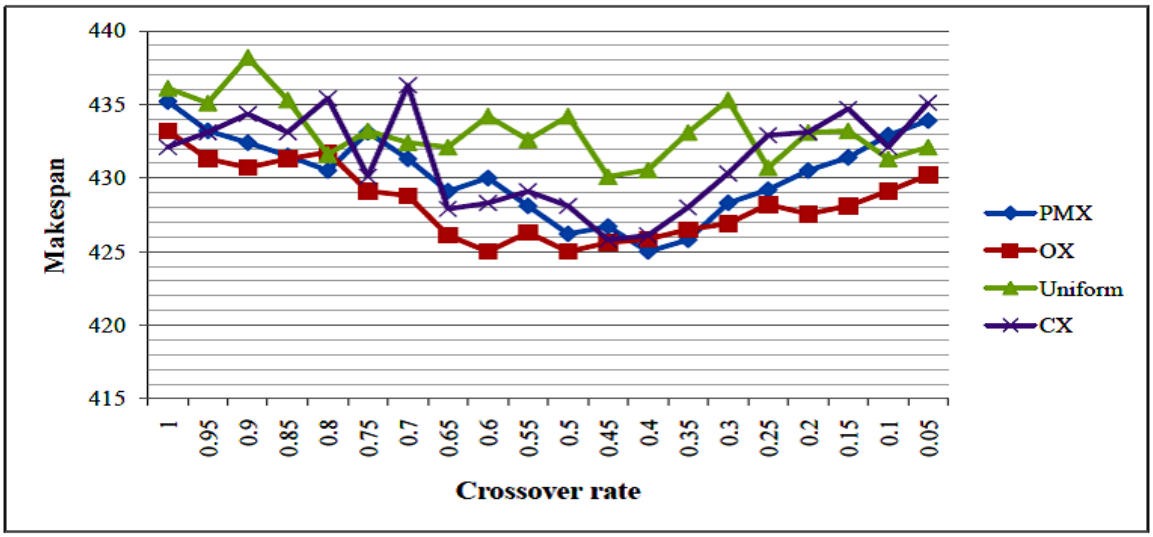

FIGURE 4. Comparison of crossover operators

In order to find the appropriate parameters for running algorithm, we performed various experiments with the different parameters. For this purpose we tested these values: mutation rate=[0.1, 0.9], Elitist rate=[0.05, 0.5], Random rate=[0.1, 0.9], k=[0.1, 0.9], α=[0.1, 0.9] and Generation number=[10, 300]. For each parameter, we repeated the experiments for 10 times, and their average was considered as the output. The results of these experiments are shown in Figure 5.

According to the results of these experiments, the optimum parameters obtained in this way: Population=70, Crossover rate=0.5, Mutation rate=0.5, k=0.5, Elitist rate=0.11, Random rate=0.6, α=0.7, Generation number=200.

In Example 2, to measure the speed of our algorithm, we use the example in the article by Lai and Wu [11]. Processing time of jobs on the machines is shown in Table 2.

Please cite this article in press as: N. Shahsavari pour, M. H. Abolhasani Ashkezari, H. Sheikhi, H. Mohammadi Andargoli, H. A. Abolhasani Ashkezari, A Novel Genetic Algorithm for a Flow Shop Scheduling Problem with Fuzzy Processing Time, Int. J. of Research in Industrial Engineering (2014)

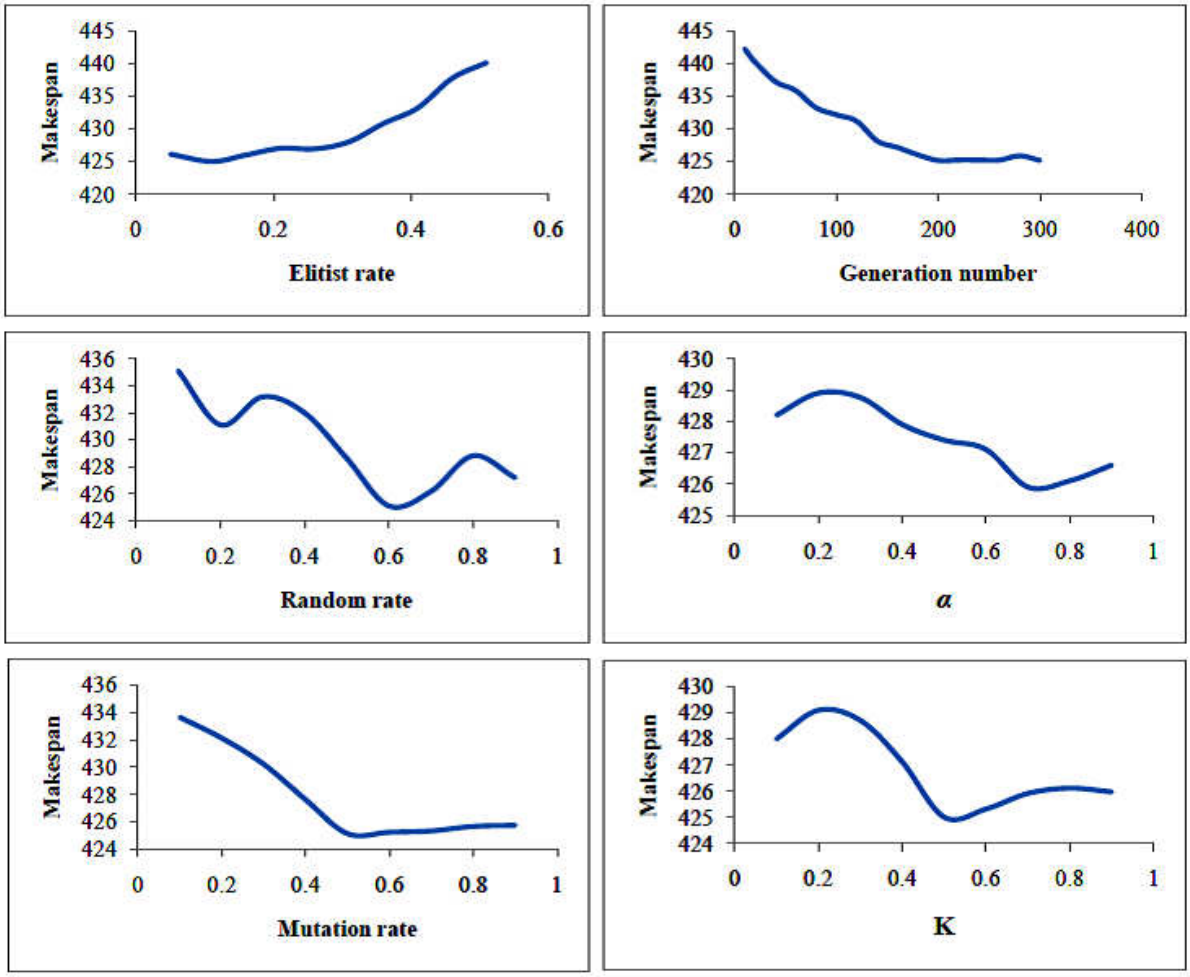

FIGURE5. Results of experiments

To make a proper comparison, we put REGA algorithm parameters as parameters of the algorithm proposed by Lai and Wu and just added the specific parameters of our algorithm. Algorithm results of Lai and Wu are shown in Table 3 and the result of the REGA is shown in Table 4. The results show that the algorithm running time is reduced.

Fuzzy makespan for all the above sequences is (145,186,186,232). Some other of optimal sequence, with the same makespan is shown in Table 5, which was achieved in the implementation of this algorithm.

TABLE 2. Fuzzy processing times

Jobs	Machine 1	Machine 2	Machine 3
J1	(6, 7, 7, 13)	(8, 11, 11, 15)	(5, 9, 9, 18)
J2	**(14, 18, 18, 22)**	**(24, 25, 25, 26)**	**(6, 15, 15, 23)**
J3	(15, 20, 20, 23)	(18, 26, 26, 28)	(8, 14, 14, 20)
J4	**(4, 8, 8, 11)**	**(8, 9, 9, 12)**	**(18, 25, 25, 29)**
J5	(9, 13, 13, 17)	(7, 10, 10, 12)	(10, 13, 13, 19)
J6	**(6, 7, 7, 11)**	**(6, 9, 9, 13)**	**(12, 16, 16, 23)**
J7	(17, 22, 22, 28)	(3, 9, 9, 14)	(26, 29, 29, 30)
J8	**(20, 25, 25, 26)**	**(16, 19, 19, 22)**	**(14, 17, 17, 18)**
J9	(6, 10, 10, 13)	(21, 27, 27, 28)	(5, 8, 8, 9)
J10	**(19, 21, 21, 26)**	**(18, 23, 23, 27)**	**(10, 11, 11, 15)**

Please cite this article in press as: N. Shahsavari pour, M. H. Abolhasani Ashkezari, H. Sheikhi, H. Mohammadi Andargoli, H. A. Abolhasani Ashkezari, A Novel Genetic Algorithm for a Flow Shop Scheduling Problem with Fuzzy Processing Time, Int. J. of Research in Industrial Engineering (2014)

TABLE 3. CPU time for Lai and Wu [11]

Exp.	Pop. size	Cross. rate	Max. gen.	Immigration rate	Opt. schedule	CPU time
1	35	0.55	60	0.35	(4, 1, 6, 9, 7, 2, 5, 3, 8, 10)	90.938
2	40	0.6	70	0.3	(4, 6, 1, 9, 7, 2, 5, 3, 8, 10)	
3	45	0.65	80	0.25	(4, 9, 6, 1, 7, 2, 5, 3, 8, 10)	121.906 154.766
4	50	0.7	90	0.2	(6, 4, 9, 5, 3, 1, 7, 2, 8, 10)	194.250 234.922
5	55	0.75	100	0.15	(1, 4, 6, 9, 7, 2, 5, 3, 8, 10)	
6	60	0.8	110	0.1	(1, 4, 6, 9, 5, 2, 7, 3, 8, 10)	282.531

TABLE 4. CPU time for the REGA

Experiment	Population size	Crossover rate	Maximal generation	Mutation rate	*K*	Random rate	*α*	Elitist rate	Fitness	Optimal schedule	CPU time
1	35	0.55	60	0.5	0.5	0.6	0.7	0.11	76.95	(4, 1, 6, 9, 7, 2, 5, 3, 8, 10)	8.38
2	**40**	**0.6**	**70**	**0.5**	**0.5**	**0.6**	**0.7**	**0.11**	**76.95**	**(4, 6, 1, 9, 7, 2, 5, 3, 8, 10)**	**10.02**
3	45	0.65	80	0.5	0.5	0.6	0.7	0.11	76.95	(4, 9, 6, 1, 7, 2, 5, 3, 8, 10)	13.43
4	**50**	**0.7**	**90**	**0.5**	**0.5**	**0.6**	**0.7**	**0.11**	**76.95**	**(6, 4, 9, 5, 3, 1, 7, 2, 8, 10)**	**15.93**
5	55	0.75	100	0.5	0.5	0.6	0.7	0.11	76.95	(1, 4, 6, 9, 7, 2, 5, 3, 8, 10)	18.49
6	**60**	**0.8**	**110**	**0.5**	**0.5**	**0.6**	**0.7**	**0.11**	**76.95**	**(1, 4, 6, 9, 5, 2, 7, 3, 8, 10)**	**21.88**

TABLE 5. Result of the REGA for other sequences

Experiment	Population size	Crossover rate	Maximal generation	Mutation rate	*K*	Random rate	*α*	Elitist rate	Fitness	Optimal schedule	CPU time
1	70	0.5	200	0.5	0.5	0.6	0.7	0.11	76.95	(1, 6, 4, 9, 5, 3, 7, 2, 8, 10)	35.03
2	**70**	**0.5**	**200**	**0.5**	**0.5**	**0.6**	**0.7**	**0.11**	**76.95**	**(6, 1, 4, 5, 9, 3, 7, 2, 8, 10)**	**35.81**
3	70	0.5	200	0.5	0.5	0.6	0.7	0.11	76.95	(6, 4, 9, 1, 5, 3, 7, 2, 8, 10)	37.44
4	**70**	**0.5**	**200**	**0.5**	**0.5**	**0.6**	**0.7**	**0.11**	**76.95**	**(1, 6, 4, 9, 5, 2, 7, 3, 8, 10)**	**36.50**

Please cite this article in press as: N. Shahsavari pour, M. H. Abolhasani Ashkezari, H. Sheikhi, H. Mohammadi Andargoli, H. A. Abolhasani Ashkezari, A Novel Genetic Algorithm for a Flow Shop Scheduling Problem with Fuzzy Processing Time, Int. J. of Research in Industrial Engineering (2014)

5. Conclusion

In this paper we solveda flow shop scheduling problem. In order to increase the efficiency and more applicable to the reality, we consider processing time of jobs on machines as a trapezoidal fuzzy numbers. We have presented a new genetic algorithm to solve this problem. In the REGA, by using elitism feature, gives a greater chance of selection for better solutions in each generation, and by using random feature, avoids from falling in local optimum. In order to evaluate the efficiency of proposed algorithm, we coded it in VBA. We first examined the effects of different crossover operators, in reaching to optimum solution. For this purpose we used an example that we know the answer. The results show that the OX operator creates a better answer for the REGA. So, we use it to evaluate the performance of our algorithm compared with the solutions Lai and Wu [11]. The results of the implementation of this algorithm show the REGA has great ability in finding the optimum solution in very short time. In this paper, we consider the problem in the single objective that it can be used for multi-objective. Also, this REGA can be used in solving other scheduling problems.

References

[1] Johnson, S.M. (1954). Optimal two and three–stage production schedules with setup times included. *Naval Research logistics Quarterly*, Vol. 1, No. 1, pp.61-68.

[2] Ogbu, F. A. and Smith, D. K. (1991). Simulated annealing for the permutation flow shop problem. *Omega*, Vol. 19, No. 1, pp. 64-67.

[3] Van Laarhoven, P. J. M. and Aarts, E. H. L. (1987). *Simulated annealing: theory and applications*. Dordrecht: D.Reidelpubl.Co.

[4] Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning*. Addison-wesley, Reading, MA.

[5] Reeves, C. (1995). A genetic algorithm for flow shop sequencing. *Computers and Operations Research*, Vol. 22, No. 1,pp. 5-13.

[6] Holland, J.H. (1975). *Adaptation in natural and artificial systems*, The university of Michigan Press, Ann Arbor.

[7] Mccahon, C. S. and Stanley, L. E. (1992). Fuzzy job sequencing for a flow shop. *European Journal of Operational Research*, Vol. 62,No.3,pp. 294-301.

[8] Tsujimura, Y., Park, S.H., Chang, I. S., and Gen, M. (1993). An effective method for solving flow shop scheduling problems with fuzzy processing times. *Computers and Industrial Engineering*, Vol. 25, No., pp.239-242.

[9] KhademiZare, H. and Fakhrzad, M.B. (2011). Solving flexible flow-shop problem with a hybrid genetic algorithm and data mining: A fuzzy approach. *Expert Systems with Applications*, Vol. 38,No. 6, pp.7609-7615.

[10] Sadinezhad, S. and GhalehAssadi, R. (2008). Preference ratio-based maximum operator approximation and its application in fuzzy flow shop scheduling. *Applied Soft Computing*, Vol. 8,No. 1,pp. 759-766.

[11] Lai, P. J. and Wu, H.C. (2011). Evaluate the fuzzy completion times in the fuzzy flow shop scheduling problems using the Virus-Evolutionary genetic algorithms. *Applied Soft Computing*, Vol. 11,No. 8,pp. 4540-4550.

[12] Deng, Y., Zhenfu, Z. and Qi, L. (2006). Ranking fuzzy numbers with an area method using radius of gyration. *Computers and Mathematics with Applications*, Vol. 51,No. 6, pp.1127-1136.

Please cite this article in press as: N. Shahsavari pour, M. H. Abolhasani Ashkezari, H. Sheikhi, H. Mohammadi Andargoli, H. A. Abolhasani Ashkezari, A Novel Genetic Algorithm for a Flow Shop Scheduling Problem with Fuzzy Processing Time, Int. J. of Research in Industrial Engineering (2014)

[13] Darwin, C. (1859). *On the origin of species by means of natural selection or the preservation of favored races in the struggle for life.* Murray, London.

[14] Whitley, D. (1989). The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best.In Proceedings of Third International Conference on Genetic Algorithms, pp. 116-121.

[15] Eshelman, L.J. (1991). *The CHC adaptive search algorithm: How to have safe search when engaging in non-traditional genetic recombination. In: Foundations of Genetic Algorithms*, Morgan Kaufmann Publishers, pp. 265-283.

[16] Deb, K., Agrawal, S., Pratap, P., and Meyarivan, T. (2000). A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In Proceedings of the Parallel Problem Solving from Nature VI Conference, pp. 849-858.

[17] Zitzler, E. and Thiele, L. (1999). Multi objective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. Evolutionary Computation, IEEE Transactions, Vol. 3,No. 4,pp. 257-271.

[18] Goldberg, D. E. and Deb, k. (1991). *A comparative Analysis of selection schemes used in genetic algorithm, in: G.J.E. Rawlins (Ed.), Foundations of genetic algorithms*, Morgan Kaufman, Los Altos, pp. 69-93.

[19] Back, T. (1994). Selective pressure in evolutionary algorithms: a characterization of selection mechanisms. In Proceedings of the First IEEE Conference on Evolutionary Computation, Piscataway, NJ: IEEE press, pp. 57-62.

[20] Oliver, I.M., Smith, D. J. and Holland, J.R.C. (1987). A study of permutation crossover operators on the traveling salesman problem," Genetic Algorithms and their Applications. Proceedings of the Second International Conference on Genetic Algorithms, Cambridge, MA: Lawrence.

[21] Syswerda, G. (1989). Uniform crossover in genetic algorithms. In Proceedings of the Third International Conference on Genetic Algorithms, CA: Morgan Kaufmann,pp. 2-9.

[22] Michalewics, Z. (1992). *Genetic algorithms+ Data structures= Evolution Programs.*Springer-Verlag.

[23] Kalczynski, P. J. and Kamburowski, J. (2008). An improved NEH heuristic to minimize makespan in permutation Flow shops. *Computers & Operations Research*, Vol. 35, pp.3001-3008.

Please cite this article in press as: N. Shahsavari pour, M. H. Abolhasani Ashkezari, H. Sheikhi, H. Mohammadi Andargoli, H. A. Abolhasani Ashkezari, A Novel Genetic Algorithm for a Flow Shop Scheduling Problem with Fuzzy Processing Time, Int. J. of Research in Industrial Engineering (2014)