International Journal of Engineering and Technology, *1 (2) (2012) 44-57*

www.sciencepubco.com/index.php/IJET

An ant colony system for solving fuzzy flow shop scheduling problem

M.H. Abolhasani Ashkezari* [a], N. Shahsavari Pour [b] and H. Mohammadi Andargoli [c]

[*a] Department of Industrial Engineering, Science and Research Branch, Islamic Azad University, Kerman, Iran
e-mail: Abolhasani.hossein@yahoo.com
[b] Department of Industrial Engineering, Science and Research Branch, Islamic Azad University, Kerman, Iran
e-mail: shahsavari_n@alum.sharif.edu
[c] Department of Industrial Engineering, Science and Research Branch, Islamic Azad University, Kerman, Iran
e-mail: hamed.mohammadi327@gmail.com

Abstract

During the past years, the flow shop has been regarded by many researchers and some extensive investigations have been done on this respect. Flow Shop includes n works performed on m machines in a same sequence. It is very difficult in the real world to determine the exact process time of an operation on a machine. Therefore, we consider in this article the process time as trapezoidal fuzzy numbers. Our purpose is that we obtain a sequence of works using such fuzzy numbers in order to minimize maximum fuzzy time of completion entire jobs or fuzzy makespan. We offered an optimization algorithm of Ant Colony System (ACS) to solve this problem. Finally, we present computational results for explanation and comparison with other articles in future.

Keywords: *Ant Colony Optimization, Flow shop problem, Makespan, Scheduling, Ranking of fuzzy number.*

1 Introduction

First, Flow Shop was studied by Johnson in 1954 [1]. It is a problem that we perform n work carried out on m machine in a similar sequence. The purpose is to find a sequence for works to minimize makespan. Many researchers have been executed on Flow Shop in the previous years, for example, the works of Widmer and Hertz [2], and Gupta and Stafford [3] confirm this interest. Flow Shop is in the collection of combinatorial optimization problems, involving in NP-hard

problems. Many heuristic methods have been yet presented to solve this problem; see [4-9]. This problem could be solved using exact calculation methods, but it will need many calculations and time; see [10-12].

Hence to solve this problem, metahuristic methods such as Simulated Annealing Algorithm (SA) [13-15], Genetic Algorithm (GA) [16,17], Tabu search Algorithm [18], and other metahuristic method have been used to a great extent and acceptable results have been gained. The method of Ant Colony optimization (ACO) is one of the metahuristic algorithms used in many cases of solving flow shop problems. In the next chapter, we will explain this algorithm in details. Ying and Liao [19], In their paper, the problem of flow shop has been studied with Ant Colony System (ACS) method and this problem has been studied in multipurpose state by Yagmahan and Yenisey [20]. Also Khalouli et al. [21] has solved hybrid flow shop by the use of ACO algorithm. ACO is a metaheuristic method using for solving many scheduling problems [22]. ACO was first presented by Dorigo [23], called Ant System (AS). The base of this algorithm was inspired by real ants using pheromone to communicate each other. Ants leave pheromone during movement from the food resource to their nest. Other ants receive the pheromone as a signal to follow, the continuous process results in finding new paths or even a better one toward the nest. Ants moving in a shorter route come to the destination sooner, causing a high traffic in the shorter route. The pheromone rate increases in shorter paths. Hence, the shorter paths that contain more pheromone concentration are chosen with higher probability by an ant. The mechanism will continue to obtain the shortest paths. ACS was developed by Dorigo and Gambardella [24] for improvement of AS performance. Different procedures have been done to improve the ACO algorithm performance in several articles; see [25-29].

Due to uncertainty in the real world with its applied problems, they should be considered in fuzzy state. In various papers, fuzzy states of flow shop problem has been studied and researched by various parameters and objectives, and many methods have been presented to solve this; see [30-38]. Also fuzzy problems were solved with ACO method by Li and Rong [39] and Cheng et al. [40]. This article is the first searching study related to the application of ACS in fuzzy flow shop. Since the environmental problems which make us unable to determine the correct time of job processing time on machines in the reality, the job processing time on machines has been considered as a fuzzy form in this article. Using the ACS, we would minimize the makespan or the completion time of jobs, in order to obtain the optimum sequence of jobs.

Here is the summary of different sections of this paper: The ACO algorithm history has been paid in the next section. More explanations about the ACS optimizing process and fuzzy theory and more information about how to do it in our problems, presented in the third and forth sections respectively. In order to observe theory performance, calculated results due to applying theory in visual basic application (VBA), are clarified in the fifth part and the last part demonstrates the analysis and inference of this article.

2 Ant Colony system

After reviewing ACO history in this part ACS algorithm was explained, which uses a cycle to approach an optimized solution. After the initial input parameters, a number of ants select randomly by state transition rule in order to move and creating some paths by their own. Each ant verifies path pheromone in his path. In order to prevent an improper increasing, algorithm use local update rule to decrease pheromone rate and in order to respect to best path after finishing entire tours by global update rule algorithm authorize optimum ant to save more pheromone on its path. This algorithm is repeated to obtain stopping rule. We explain this procedure in this section in details.

2.1 State transition rule in ACS

The ant *k* is now in node *i* choosing the next node *j* using a rule called "state transition rule". Suppose that q_0 $(0 \le q_0 \le 1)$ is a parameter that determines the importance between of exploration and exploitation. In this case, first a quantity of *q* is chosen using uniform function distributed in [0, 1]. Now if $q \le q_0$, the best job will be chosen based on the following formula:

$$j = \begin{cases} \arg\max_{u \in S_K(i)} \left\{ [\tau(i,u)]^{\alpha} [\eta(i,u)]^{\beta} \right\} & \text{if} \quad q \le q_0 \\ J & \text{otherwise} \end{cases} \tag{1}$$

Where, *J* is a random variable. Otherwise if $q > q_0$, the best job will be chosen based on random-proportional rule, shown with the following equation:

$$p_k(i,j) = \begin{cases} \dfrac{[\tau(i,j)]^{\alpha} [\eta(i,j)]^{\beta}}{\sum_{u \in S_k(i)} [\tau(i,u)]^{\alpha} [\eta(i,u)]^{\beta}} & \text{if} \quad j \in S_k(i) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Where $\tau(i,u)$ is the density of pheromone between two nodes *i* and *u*, and $\eta(i,u) = \dfrac{1}{\delta(i,u)}$ the inverse of distance $(\delta(i,u))$ between two nodes *i* and *u*, and $S_k(i)$ is the set of nodes that could be chosen by ant k. α is a parameter that we could control the influence of pheromone $(\alpha > 0)$ and β is a parameter that determines the importance of distance $(\beta > 0)$.

2.2 Local updating rule in ACS

An ant reduces the rate of pheromone of observed nodes during a tour by using local updating rule. It aims at prevention of hyper-increase of pheromone of observed nodes. It causes to enable ant to find other nodes and new ways, and thus, avoid any engagement in a local optimum. The equation of local updating rule is as follows:

$$\tau(i,j) = (1-\rho l)\tau(i,j) + \rho l \times \tau_0 \tag{3}$$

Where, τ_0 is initial rate of pheromone and $\rho l(0 \le \rho l \le 1)$ is pheromone evaporating parameter.

2.3 Global updating rule in ACS

Global updating rule is used after all ants finish their tour. In order to pay more attention to the best tour found, only the ant finding the shortest way is allowed to save more pheromone, this mechanism shown as follows:

$$\tau(i,j) = (1-\rho g)\tau(i,j) + \rho g \times \Delta\tau(i,j) \tag{4}$$

Where

$$\Delta\tau(i,j) = \begin{cases} \left(l_{gb}\right)^{-1} & if \quad (i,j) \in global\ best\ tour \\ 0 \end{cases} \tag{5}$$

Where $\rho g(0 \le \rho g \le 1)$ is pheromone evaporating parameter, and l_{gb} is the length of shortest tour found.

3 Implementation of ACS in fuzzy flow shop

After describe ACS algorithm, in this section we described implementation of this algorithm in fuzzy flow shop problem.

3.1 Theory of fuzzy logic in flow shop

In flow shop, like the other real problems, because products dynamic surrounding collate, exact determination of goals, constraints and model parameters are really hard. As figured, in this article process times are shown on by trapezoidal fuzzy number *(a,b,c,d:w)* (see fig. 1).

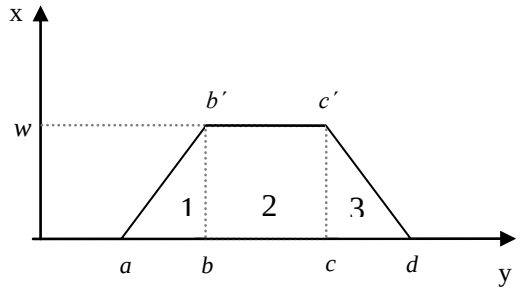

Fig. 1.Trapezoidal fuzzy number *(a,b,c,d:w)*.

For convert fuzzy number *(a,b,c,d:w)* to crisp number *S(A)*, we use "The radius of gyration of an area" presented by Deng et al [41]. Consider an area *A* and the element of area *dA* of coordinates *x* and *y* (see fig. 2).

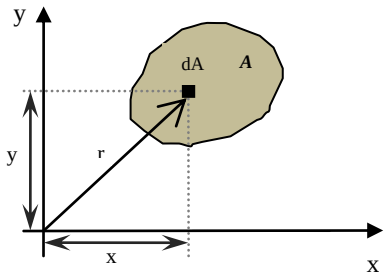

Fig. 2.The moment of inertia of an area.

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

$$I_x = \int_A y^2 \ dA, \tag{6}$$

$$I_y = \int_A x^2 \ dA. \tag{7}$$

The radius of gyration of an area *A* with respect to the *x* (r_x) axis and *y* (r_y) are as follow:

$$I_x = r_x^2 A \quad \Rightarrow \quad r_x = \sqrt{\frac{I_x}{A}} \tag{8}$$

$$I_y = r_y^2 A \quad \Rightarrow \quad r_y = \sqrt{\frac{I_y}{A}} \tag{9}$$

In this paper, when a generalized fuzzy number *A* is given, the radius of gyration points of the generalized fuzzy number *A* is denoted as $(r_x(A), r_y(A))$ that we can calculate from above equations (8, 9). For example, in Fig. 1, the moment of inertia of generalized trapezoidal fuzzy number can be calculated as follow:

$$(I_x) = (I_x)_1 + (I_x)_2 + (I_x)_3 \ , \tag{10}$$

$$(I_y) = (I_y)_1 + (I_y)_2 + (I_y)_3 \tag{11}$$

Where,

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

$$(I_y)_1 = \int_{abb'} x^2 dA = \frac{(b-a)^3 w}{4} + \frac{(b-a)a^2 w}{2} + \frac{2(b-a)^2 aw}{3} \tag{13}$$

And, we can calculate $(I_x)_2, (I_y)_2, (I_x)_3, (I_y)_3$, same (12) and (13):

$$(I_x)_2 = \frac{(c-b)w^3}{3}, \tag{14}$$

$$(I_x)_3 = \frac{(d-c)w^3}{12}, \tag{15}$$

$$(I_y)_2 = \frac{(c-b)^3 w}{3} + (c-b)b^2 w + (c-b)^2 bw, \tag{16}$$

$$(I_y)_3 = \frac{(d-c)^3 w}{12} + \frac{(d-c)c^2 w}{2} + \frac{(d-c)^2 cw}{3}, \tag{17}$$

So, the radius of gyration of an area *A* calculated as follow:

$$r_x(A) = \sqrt{\frac{(I_x)_1 + (I_x)_2 + (I_x)_3}{(((c-b)+(d-a)).w)/2}}, \tag{18}$$

$$r_y(A) = \sqrt{\frac{(I_y)_1 + (I_y)_2 + (I_y)_3}{(((c-b)+(d-a)).w)/2}} \tag{19}$$

And, finally, Rank fuzzy numbers can be obtained as follow:

$$S(A) = r_x(A) \times r_y(A). \tag{20}$$

3.2 Calculate distance between two jobs

To calculate the distance between these two jobs we used SPIRIT method presented by Widmer and Hertz in 1989 [2]. On the basis of this method, the d_{ij} distance between $job(i)(i = 1,..,n \cup N)$ and $job(u)(u = 1,..,n)$ is shown through the following equation:

$$d_{ij} = t_{i1} + \sum_{k=2}^{m} (m-k) * \left| t_{ik} - t_{jk-1} \right| + t_{jm} \tag{21}$$

Where t_{ij} is the duration of performing *job(i)(i=1,…,n)* on *machine(j)(j=1,…,m)*, And heuristic information is given as follows:

$$\eta(i,u) = \frac{1}{d_{iu}} \qquad (i \neq u) \tag{22}$$

4 Experimental results

In this part of paper, in order to explain our algorithm and to observe the function of this algorithm in comparison to other papers, we represent our implementations of this algorithm. Many papers have considered the flow shop problem with the aim of minimizing makespan, while the presented paper has studied this problem in fuzzy mood. The differences exist within the fuzzy or crisp solutions, and solutions driven by theory or reality can be observed, compared and analyzed .For flow shop, Taillard [42] has studied typical problems with various numbers of machines and different jobs. To find the appropriate answers, first we solved the problem with different parameters. To discover best parameter for our algorithm, we solved our problem with $\alpha = (0.1,10)$, $\beta = (0.1,20)$, $q_0 = (0.5,0.99)$, $ant\ number = (10,120)$, $\rho l = (0.01,0.99)$, $\rho g = (0.01,0.99)$. For every parameter, We repeated the experiment for 10 times. Finally, we use ANOVA for analyze different parameter (for example, see Fig. 3, and 4 for α and β). Through repeated experiments we found the quantities of $\alpha = 0.5$, $\beta = 9$, $q_0 = 0.8$, $ant\ number = 30$, $\rho l = 0.1$, $\rho g = 0.15$, and fixed these parameters for our next experiments on these quantities. The algorithm terminates when the total number of iterations reaches 5000.

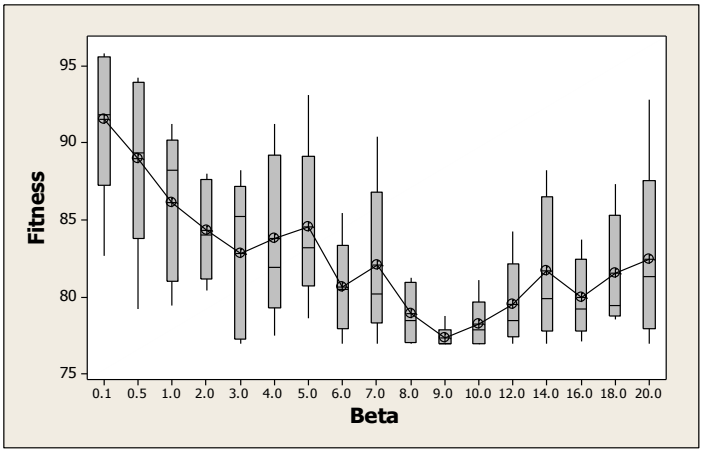

Fig. 3. Analyze of β.

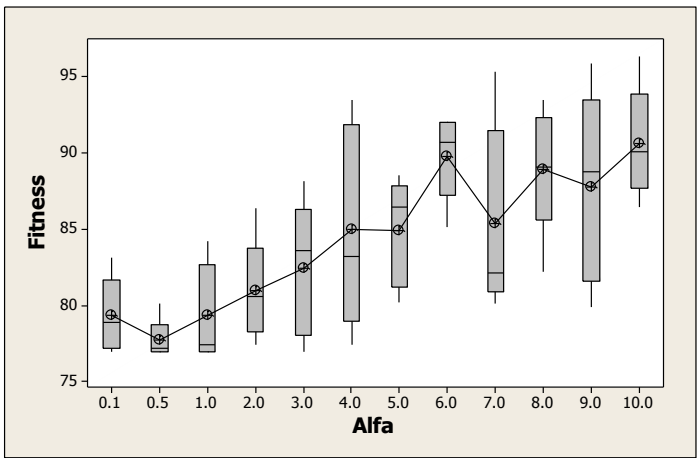

Fig. 4. Analyze of α .

In this paper, to examine our method we presented two procedures. First we compared our results with Lai and Wu [38], and then we presented our implementation for different example for future research. We coded this algorithm in VBA and implemented in Intel T6570, 2.1 GHz with 2 GB RAM.

Example 4.1 According to Lai and Wu [38], in table 1, we considered fuzzy processing times of 3 machines and 10 jobs, and run our ACS algorithm. The results of our implementation are shown in table 2. We accepted the result of Lai and Wu, and we found some other optimum permutations with the same fitness calculating from (20).

Table 1: Fuzzy processing times

	Fuzzy processing times in machine 1	Fuzzy processing times in machine 2	Fuzzy processing times in 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)

Table 2: The best permutations with ACS

Best permutations presented by Lai and Wu (2011)	ACS permutations	Fuzzy Makespan	Fitness
(4,1,6,9,7,2,5,3,8,10)	(4,1,6,9,7,2,5,3,8,10)	(145, 186, 186, 232)	76.95
(4,6,1,9,7,2,5,3,8,10)	(4,6,1,9,7,2,5,3,8,10)	(145, 186, 186, 232)	76.95
(4,9,6,1,7,2,5,3,8,10)	(4,9,6,1,7,2,5,3,8,10)	(145, 186, 186, 232)	76.95
(6,4,9,5,3,1,7,2,8,10)	(6,4,9,5,3,1,7,2,8,10)	(145, 186, 186, 232)	76.95
(1,4,6,9,7,2,5,3,8,10)	(1,4,6,9,7,2,5,3,8,10)	(145, 186, 186, 232)	76.95
(1,4,6,9,5,2,7,3,8,10)	(1,4,6,9,5,2,7,3,8,10)	(145, 186, 186, 232)	76.95
Some other optimum permutations	(6,4,9,1,5,2,7,3,8,10)	(145, 186, 186, 232)	76.95
	(6,4,9,1,5,3,7,2,8,10)	(145, 186, 186, 232)	76.95
	(6,4,9,5,2,7,1,3,8,10)	(145, 186, 186, 232)	76.95
	(6,4,9,5,1,2,7,3,8,10)	(145, 186, 186, 232)	76.95
	(6,4,9,1,5,3,7,2,8,10)	(145, 186, 186, 232)	76.95
	(6,4,9,5,2,1,7,3,8,10)	(145, 186, 186, 232)	76.95
	(1,6,4,9,5,3,7,2,8,10)	(145, 186, 186, 232)	76.95
	(1,6,4,5,9,2,7,3,8,10)	(145, 186, 186, 232)	76.95
	(1,6,4,9,5,2,7,3,8,10)	(145, 186, 186, 232)	76.95
	(1,4,6,9,5,3,7,2,8,10)	(145, 186, 186, 232)	76.95
	(6,1,4,5,9,3,7,2,8,10)	(145, 186, 186, 232)	76.95
	(6,1,4,5,9,2,7,3,8,10)	(145, 186, 186, 232)	76.95
	(6,4,1,5,9,3,7,2,8,10)	(145, 186, 186, 232)	76.95
	(6,4,1,9,5,3,7,2,8,10)	(145, 186, 186, 232)	76.95
	(6,4,1,9,5,2,7,3,8,10)	(145, 186, 186, 232)	76.95
	(1,4,9,6,5,3,7,2,8,10)	(145, 186, 186, 232)	76.95
	(1,4,9,5,6,2,7,3,8,10)	(145, 186, 186, 232)	76.95
	(1,4,9,6,5,2,7,3,8,10)	(145, 186, 186, 232)	76.95
	(4,1,5,9,6,2,7,3,8,10)	(144, 187, 187, 232)	76.96
	(4,1,5,6,9,2,7,3,8,10)	(144, 187, 187, 232)	76.96
	(4,6,1,5,9,2,7,3,8,10)	(144, 187, 187, 232)	76.96

Example 4.2 In this example, to obtain a base for future research, the fitness of some problems are calculated. We consider 5, 10, and 20 machines and 5 to 200 jobs. For any pair of machines and jobs, randomly generated an integer process time t_{ij} $(i=1,\ldots,n$ and $j=1,\ldots,m)$ from the uniform distribution [1, 20]. To assure the calculations and obtained quantity for every example, we repeated the experiment 10 times and considered their fitness as the output numbers, which the result are shown in table 3. In fig. 5, the relationship between jobs and machines number and their fitness are shown.

Table 3: Computational experiments and results (m=5)

m×n	Fitness 1	Fitness 2	Fitness 3	Fitness 4	Fitness 5	Fitness 6	Fitness 7	Fitness 8	Fitness 9	Fitness 10	Average
5×5	47.76	43.01	45.35	50.01	51.99	48.40	50.81	45.95	49.28	53.52	**48.61**
5×10	83.04	80.95	81.27	83.02	84.56	81.24	88.68	80.83	85.92	82.24	**83.18**
5×20	120.01	126.78	120.31	126.55	126.20	128.28	119.81	122.44	133.66	126.32	**125.04**
5×30	170.28	172.45	184.91	180.61	184.81	167.20	168.01	180.38	173.60	180.05	**176.23**
5×40	224.45	223.36	240.26	229.84	229.96	215.73	221.27	226.77	231.91	222.85	**226.64**
5×50	276.84	291.14	281.55	282.41	279.95	281.30	276.47	284.61	288.44	282.90	**282.56**
5×60	327.83	329.56	329.10	320.19	321.41	332.65	333.31	319.82	328.73	320.08	**326.27**
5×70	380.54	381.96	378.01	379.91	390.01	387.61	377.67	371.92	388.30	389.91	**382.58**
5×80	427.14	430.35	437.61	428.80	433.61	429.05	417.90	429.71	423.63	419.82	**427.76**
5×90	485.83	481.92	491.87	493.47	478.37	488.93	485.12	488.83	483.27	479.00	**485.66**
5×100	537.61	539.37	555.87	536.72	544.19	539.02	547.85	551.62	532.64	542.74	**542.76**
5×120	646.07	651.92	645.05	643.87	657.47	642.91	659.02	663.95	641.95	643.33	**649.55**
5×140	750.13	738.00	733.39	743.60	751.45	748.85	741.90	749.04	752.18	739.25	**744.78**
5×160	861.58	877.18	871.16	869.94	879.28	871.20	870.05	879.63	883.46	869.93	**873.34**
5×180	961.47	977.61	966.10	978.85	969.01	957.01	978.72	965.92	970.29	979.01	**970.40**
5×200	1053.68	1048.00	1055.91	1059.38	1060.72	1045.30	1048.95	1059.08	1051.19	1042.70	**1052.49**

Table 4: Computational experiments and results (m=10)

m×n	Fitness 1	Fitness 2	Fitness 3	Fitness 4	Fitness 5	Fitness 6	Fitness 7	Fitness 8	Fitness 9	Fitness 10	Average
10×5	71.03	71.03	75.70	74.70	82.04	81.35	79.21	79.67	79.67	75.11	**76.95**
10×10	101.24	101.41	106.24	102.33	107.29	100.08	104.38	99.46	101.12	105.76	**102.93**
10×20	149.52	143.92	151.00	149.96	155.52	152.09	161.33	153.25	155.56	148.94	**152.11**
10×30	204.23	212.03	209.30	211.23	214.18	201.40	209.55	208.65	207.29	213.88	**209.17**
10×40	263.74	266.54	259.00	268.75	260.03	269.79	268.13	258.44	261.80	266.69	**264.29**
10×50	320.16	318.73	322.50	321.94	317.44	329.91	322.23	327.71	320.90	318.48	**322.00**
10×60	379.13	373.48	370.58	360.92	376.62	367.81	372.97	365.62	362.22	368.01	**369.74**
10×70	433.40	430.94	427.27	420.83	439.80	433.38	421.29	425.59	437.00	431.03	**430.05**
10×80	488.70	486.93	480.76	484.33	478.90	480.87	471.22	488.29	487.43	481.91	**482.93**
10×90	544.89	549.42	546.36	540.09	551.18	543.21	557.95	550.80	540.90	549.11	**547.39**
10×100	595.02	589.61	599.98	587.71	590.61	570.84	590.34	585.10	601.79	598.09	**590.91**
10×120	700.39	702.87	690.03	711.21	706.70	698.07	689.12	699.91	698.27	705.56	**700.21**
10×140	807.77	810.98	813.30	808.76	801.94	804.77	817.19	810.86	798.90	809.61	**808.41**
10×160	917.04	923.30	910.62	909.91	927.96	920.61	919.01	924.90	929.39	931.36	**921.41**
10×180	1017.20	1009.91	1017.54	1020.93	1010.90	1028.27	1002.76	998.52	1012.80	1019.59	**1013.84**
10×200	1112.57	1129.73	1121.21	1109.92	1113.50	1101.69	1119.37	1115.20	1128.41	1130.79	**1118.24**

Table 5: Computational experiments and results (m=20)

m×n	Fitness 1	Fitness 2	Fitness 3	Fitness 4	Fitness 5	Fitness 6	Fitness 7	Fitness 8	Fitness 9	Fitness 10	Average
20×5	129.76	131.24	139.82	127.53	129.00	138.01	141.82	139.03	129.11	135.56	**134.09**
20×10	159.81	165.52	165.83	167.01	157.10	167.29	166.22	161.19	159.02	155.47	**162.45**
20×20	220.15	221.19	229.19	219.01	216.90	229.00	231.67	218.98	222.71	229.25	**223.81**
20×30	282.18	276.92	286.02	290.10	279.01	292.37	285.74	281.90	287.26	271.82	**283.33**
20×40	336.02	340.00	331.74	328.67	344.27	335.62	339.02	324.10	338.41	343.99	**336.18**
20×50	394.42	400.71	399.14	415.34	411.13	403.17	397.42	387.00	409.12	408.86	**402.63**
20×60	456.51	461.02	466.22	479.20	459.82	461.73	450.82	467.68	471.00	456.77	**463.08**
20×70	508.30	498.11	517.72	511.74	519.95	519.64	504.87	513.01	521.92	500.09	**511.54**
20×80	561.71	574.01	559.27	580.09	578.00	558.11	569.71	569.48	561.99	571.73	**568.41**
20×90	625.01	627.47	631.01	633.31	618.92	621.00	616.19	619.69	623.89	620.00	**623.65**
20×100	693.41	700.01	709.90	699.39	710.87	707.11	690.00	688.12	694.31	703.97	**699.71**
20×120	785.17	789.00	769.24	776.10	772.82	794.23	781.81	787.07	779.93	781.01	**781.64**
20×140	903.53	900.37	901.92	898.30	889.45	909.34	897.77	899.40	905.12	910.56	**901.58**
20×160	1010.53	1017.73	1023.10	1018.25	1029.71	1006.85	1025.25	1009.79	1012.81	1022.82	**1017.68**
20×180	1111.88	1119.92	1109.01	1112.43	1107.91	1110.10	1123.81	1118.72	1129.41	1120.63	**1116.38**
20×200	1229.72	1221.93	1220.03	1239.71	1230.10	1243.92	1217.28	1238.82	1248.35	1218.39	**1230.83**

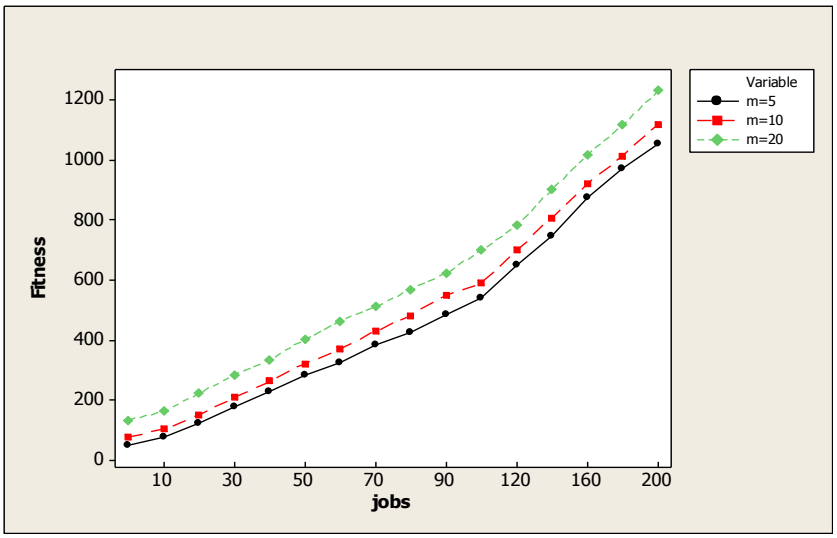

Fig. 5.The relationship between jobs and machines number and their fitness.

5 Conclusion

In order to make the problem of flow shop scheduling more applicable, we considered it in fuzzy state. For this purpose we considered the process times of jobs on machines in the form of fuzzy numbers. In this problem, our goal was to find an optimum sequence that we could be able to minimize the time of completing the jobs (makespan). To realize this goal and to solve this problem, we presented an algorithm of Ant Colony System (ACS). Because this paper is first research in implementation of ACS in fuzzy flow shop, in order to observe different between fuzzy and crisp states and assay our algorithm, we had coded our problem in VBA which could be referred to previous section, and compare our results with another article. In the end, we present some examples for evaluation of our algorithm.

In future research we can apply this presented ACS algorithm in job shop, open shop, and other scheduling problem. We, also, studied the problem as a single-objective form which is able to be studied with other objectives or as the multi-objective ones. On the other side we used ACS algorithm to solve this problem. Other ACO methods or other metahuristic method could also be used.

References

[1] S.M. Johnson, Optimal two and three–stage production schedules with setup times included, *Naval Research logistics Quarterly* 1 (1) (1954) 61-68.
[2] M. Widmer, A. Hertz, A new heuristic method for the flow shop sequencing problem, *European Journal of Operational Research* 42 (2) (1989) 186-193.
[3] J.N.D. Gupta, E.F. Stafford, Flow shop scheduling research after five decade, *European Journal of Operational Research* 169 (3) (2006) 699-711.
[4] D.S. Palmer, Sequencing jobs through a multi stage process in the minimum total time : A quick method of obtaining a near optimum, *Operation Research Quarterly* 16 (1) (1965) 101-107.
[5] M.L. Smith, R.A. Dudek, A general algorithm for solution of the n-job, m-machine sequencing problem of the flow shop, *Operations Research* 15 (1967) 71-82.
[6] J.N.D. Gupta, An improved combinatorial algorithm for the flow shop scheduling problem, *Operations Research* 19 (7) (1971) 1753-1758.
[7] D.G. Dannenbring, An evaluation of flow shop sequencing heuristics, *Management Science* 23 (11) (1977) 1174-1182.
[8] M.E. Nawaz, E. Enscore, I. Ham, A heuristic algorithm for the m-machine, n-job flow shop sequencing problem, *Omega* 11 (1) (1983) 91-95.
[9] T.S. Hundal, J. Rajgopal, An extension of palmer's heuristic for the flow-shop scheduling problem, *International Journal of production Research* 26 (1988) 1119-1124.
[10] M. Pinedo, Scheduling: theory, algorithms, and systems, Prentice *Hall, Englewood Cliffs, NJ*, 1995.
[11] E. Ignall, L.E. Schrage, Application of the branch and bound technique to some flow shop scheduling problems, *Operations Research* 13 (3) (1965) 400-412.
[12] G.B. McMahon, P.G. Burton, Flow shop scheduling with the branch and bound method, *Operations Research* 15 (1967) 473-481.
[13] P.J.M. Van Laarhoven, E.H.L. Aarts, Simulated annealing: theory and applications, *Dordrecht: D.Reidel publ.Co*, 1987.
[14] I.H. Osman, C.N. potts, Simulated annealing for permutation Flow-shop scheduling, *Omega* 17 (6) (1989) 551-557.
[15] F.A. Ogbu, D.K. Smith, Simulated annealing for the permutation flow shop problem, *Omega* 19 (1) (1991) 64-67.
[16] D.E. Goldberg, Genetic algorithms in search, optimization and machine learning, *Reading, MA: Addison- Wesley*, 1989.
[17] C. Reeves, A genetic algorithm for flow shop sequencing, *Computers and Operations Research* 22 (1) (1995) 5-13.
[18] E. Nowicki, C. Smutnicki, A fast tabu search algorithm for the permutation flow shop problem, *European Journal of Operational Research* 91 (1) (1996) 160-175.

[19] K.C. Ying, C.J. Liao, An ant colony system for permutation flow shop sequencing, *Computers and Operations Research* 31 (5) (2004) 791-801.
[20] B. Yagmahan, M.M. Yenisey, Ant colony optimization for multi-objective flow shop scheduling problem, *Computers and Industrial Engineering* 54 (2008) 411-420.
[21] S. Khalouli, F. Ghedjati, A. Hamzaoui, A meta-heuristic approach to solve a JIT scheduling problem in hybrid flowshop, *Engineering Applications of Artificial Intelligence* 23 (5) (2010) 765-771.
[22] T. Stutzle, M. Dorigo, The ant colony optimization metaheuristic: Algorithms, applications, and advances, *Handbook of metaheuristics, Norwell,MA: kluwer Academic publishers*, 2003, pp.251-285.
[23] M. Dorigo, Optimization, learning and natural algorithm, *Ph.D. thesis, DEI, Politecnico Di Milano*, Italy, 1992.
[24] M. Dorigo, L.M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, *IEEE transactions on Evolutionary computation* 1 (1) (1997) 53-66.
[25] T. Stutzle, H. Hoos, Improvements on the Ant-system: Introducing max–min ant system, *Computer and Information Science* (1996) 1-23.
[26] T. Stutzle, H. Hoos, The max–min ant system and local Search for the traveling salesman problem, *Proceedings of 1997 IEEE International Conference on Evolutionary Computation*, IEEE press, 1997, pp.309-314.
[27] B. Bullnheimer, R.F. Hartl, C. Strauss, A new rank based version of the Ant system: A computational study, *Central European Journal for Operations Research and Economics* 7 (1) (1999) 25-38.
[28] O. Cordon, I. ferandez de Viana, F. Herrera, L. Moreno, A new aco model integrating evolutionary computation concepts: the best–worst Ant system, *universito libre de Bruxelles, Belgium*, 2000, pp. 22-29.
[29] L.M. Gambardella, M. Dorigo, Ant–Q: A reinforcement learning approach to the traveling salesman problem, *Proceedings of the twelfth international conference on machine learning*, morgan kaufman, 1995, pp.252-260.
[30] P.M. Stanfield, R.E. King, J.A. Joines, Scheduling arrivals to a production system in a fuzzy environment, *European Journal of Operational Research* 93 (1) (1996) 75-87.
[31] T.P. Hong, T.T. Wang, Fuzzy flexible flow shops at two machine centers for continuous fuzzy domains, *Information Sciences* 129 (1-4) (2000) 227-237.
[32] J. Balasubramanian, I.E. Grossmann, Scheduling multi-stage flow shops with parallel units: an alternative approach to optimization under uncertainty, *Computer Aided Chemical Engineering* 15 (2003) 154-159.
[33] Y. Tsujimura, S.H. Park, I.S. Chang, M. Gen, An effective method for solving flow shop scheduling problems with fuzzy processing times, *Computers and Industrial Engineering* 25 (1-4) (2003) 239-242.
[34] C.S. Mccahon, L.E. Stanley, Fuzzy job sequencing for a flow shop, *European Journal of Operational Research* 62 (3) (2003) 294-301.

[35] B. Javadi, M. Saidi-Mehrabad, A. Haji, I. Mahdavi, No-wait flow shop scheduling using fuzzy multi-objective linear programming, *Journal of the Franklin Institute* 345 (2008) 452-467.
[36] S. Sadi nezhad, R. Ghaleh Assadi, Preference ratio-based maximum operator approximation and its application in fuzzy flowshop scheduling, *Applied Soft Computing* 8 (1) (2008) 759-766.
[37] H. Khademi Zare, M.B. Fakhrzad, Solving flexible flow-shop problem with a hybrid genetic algorithm and data mining: A fuzzy approach, *Expert Systems with Applications* 38 (6) (2011) 7609-7615.
[38] P.J. Lai, H.C. Wu, Evaluate the fuzzy completion times in the fuzzy flow shop scheduling problems using the Virus-Evolutionary genetic algorithms, *Applied Soft Computing* 11 (8) (2011) 4540-4550.
[39] S.G. Li, Y.L. Rong, The reliable design of one-piece flow production system using fuzzy ant colony optimization, *Computers and Operations Research* 36 (5) (2009) 1656-1663.
[40] B. Cheng, K. Li, B. Chen, Scheduling a single batch-processing machine with non-identical job sizes in fuzzy environment using an improved ant colony optimization, *Journal of Manufacturing Systems* 29 (1) (2010) 29-34.
[41] Y. Deng, Z. Zhenfu, L. Qi, Ranking fuzzy numbers with an area method using radius of gyration, *Computers and Mathematics with Applications* 51 (6-7) (2006) 1127-1136.
[42] E. Taillard, Benchmarks for basic scheduling problems, *European Journal of Operational Research* 64 (2) (1993) 278-285.