

Solving group scheduling problem in no-wait flexible flowshop with random machine breakdown

A. Adressi*, S. Tasouji Hassanpour and V. Azizi

Department of Industrial Engineering, K. N. Toosi University of Technology, Iran, Tehran

CHRONICLE

Article history:
Received September 16, 2014
Received in revised format:
June 12, 2015
Accepted July 6, 2015
Available online
July 8 2015

Keywords:
Group Scheduling
No-wait Flowshop
Sequence-dependent Setup Times
Machine Breakdown

ABSTRACT

In this paper, group scheduling problem in no-wait flexible flowshop is considered by considering two stages with group sequence-dependent setup times and random breakdown of the machines. Genetic algorithm and simulated annealing based heuristics have been proposed to solve the problem. The primary objective of scheduling is to minimize the maximum completion time of the jobs for two classes of small and large scale problems. Computational results show that both GA and SA algorithms perform properly, but SA appeared to provide better results for both small and large scale problems.

© 2016 Growing Science Ltd. All rights reserved.

1. Introduction

Common manufacturing approaches are being replaced incessantly by new methods in order to improve the effectiveness and proficiency of the whole manufacturing system. Various manufacturing corporations take advantage of scheduling algorithms to deal with customer's demands and to reduce their own operational costs. In group scheduling problems, all jobs in the same group require identical setup times on the machines. Therefore, by grouping jobs into some groups, we can avoid many wasting times in the schedule. In a no-wait flowshop scheduling problem, it is assumed that n jobs are processed through m machines in a flowshop environment. When the process of a specific job begins on the first machine, it should constantly be processed without waiting in the line of any machine until its processing is completed on the last machine. Scheduling problems such as separable setup times are categorized into sequence-independent and sequence-dependent scheduling problems. We consider a problem as “group scheduling with sequence-dependent setup times” provided that setup time of a group on a machine relies on the preceding one on that machine, otherwise, we will consider it as “group scheduling with sequence-independent setup times”. In this research, group scheduling problem

* Corresponding author.
E-mail address: aaddressi@mail.kntu.ac.ir (A. Adressi)

in no-wait flexible flowshop considering two stages with group sequence-dependent setup times and also random breakdown of the machines is investigated. Some effective metaheuristic methods have been developed to solve the problem due to objective function of minimizing the makespan of the jobs.

2. Literature Review

In recent years, considerable amount of studies have been devoted on no-wait flowshop scheduling problem. This kind of manufacturing approach can be used in many manufacturing systems and firms such as steel industries where steel should be processed among the machines continuously without losing its temperature, food industries where the food must be packed as conserves immediately after being prepared, plastic modeling where the plastic must be shaped to the desired form before it losses temperature, etc. Initial researches about no-wait flowshop was presented by Artanary (1971, 1974). Rajendran (1994) presented a heuristic method for a no-wait flowshop scheduling problem in order to minimize the maximum completion time of the jobs. Hall and Sriskandarajah (1996) surveyed machine scheduling problems with blocking and no-wait in process, extensively. Aldowaisan and Allahverdi (2003) developed genetic and simulated annealing algorithms based heuristics by considering makespan of jobs as the objective function for no-wait flowshop problems. Comprehensive survey on different studies was performed during the past 50 years about this category of machine scheduling problem was presented by Gupta and Stafford (2006). Gupta et al. (1997) studied two models of no-wait two-stage processing where the first model was about flowshop environment with two machines, the second one was about assembly line and the objective function for both problems was minimization of the maximum completion time of the jobs. They assumed that setup times and transportation times are separated from processing times. They showed that two-stage flowshop problem can be converted into travelling salesman problem and it can be solved in polynomial time. Junlin et al. (2004) investigated two-stage flowshop problem by assuming that, at least, one of the stages includes some identical parallel machines and setup and transportation times are separated from processing times of the jobs on machines. Attar et al. (2011) surveyed flexible flowshop scheduling problem under the assumptions such as sequence-dependent setup times, waiting times and ready times for jobs. In flexible flowshop, at least, one stage includes some identical machines and a particular job can be assigned to one of them. They developed a new Imperialist Competitive Algorithm (ICA) to solve the problem and to validate their algorithm and they compared it to a simulated annealing algorithm. Results showed that ICA performed better than simulated annealing. Shafaei et al. (2011) studied no-wait two-stage flexible flowshop problem with the objective of minimizing the maximum completion time of the jobs. They developed a method named Adaptive Neuro Fuzzy Inference System (ANFIS) to predict the solution time of this category of problems and compared it with 6 heuristic methods in order to evaluate the effectiveness of the method. Liu et al. (2003) proposed a heuristic algorithm for solving two-stage no-wait hybrid flow shop scheduling by considering single machine in either stage. Wang et al. (2005) solved no-wait flexible flow shop scheduling with no-idle machines. Schaller (2001) proposed a new lower bound for the flow shop group scheduling problem. Samarghandi and Elmekawy (2012) proposed two metaheuristic algorithms based on genetic and particle swarm optimization algorithms for solving no-wait flow shop problem with separable setup times and makespan criterion.

In addition to extensive research studies on the no-wait flowshop scheduling problem, setup times on the machines have attracted many researches to focus on this issue. Mostly, the setup time is presumed to be sequence-dependent. This concept aligns with the family grouping situation where similar products with similar setup times on machines are grouped together in order to reduce the setup times required for the whole system. A review on scheduling problems due to setup times can be found in the research presented by Allahverdi et al. (1991). Lin and Liao (2003) developed a recursive based heuristic to solve the two-stage hybrid flowshop problem considering sequence-dependent setup times and due dates with the objective function of minimizing weighted maximal tardiness. Moradinasab et al. (2012) presented their research about no-wait two-stage flexible flowshop problem by assuming sequence-dependent setup times with the objective function of minimizing completion time of the jobs and developed a genetic algorithm and Imperialist Competitive algorithms to solve the problem. Ashhari

(2012) studied no-wait flexible flowshop group scheduling problem with sequence-dependent setup times for the first time. Salmasi et al. (2011) studied total flow time minimization in a flow shop group scheduling problem by considering sequence dependent setup times. Ruiz et al. (2005) used metaheuristic algorithms for solving the flow shop scheduling problem with sequence setup times. Mirabi et al. (2013) considered a two stage hybrid flow shop scheduling problem by considering machine breakdown condition. Logendran et al. (2006) studied two machine group scheduling problems in discrete parts manufacturing with sequence dependent setup times. Jolai et al. (2012) presented a novel metaheuristic algorithm for solving no-wait flexible flow shop with sequence dependent setup times. Gholami et al. (2009) solved hybrid flow shop with sequence dependent setup times with condition of random machine breakdown. Ahmadizar et al. (2009) studied application of chance-constrained programming for stochastic group shop scheduling problems. There has been no research accomplished about no-wait flexible flowshop group scheduling problems with sequence-dependent setup times considering random machine breakdowns. The remaining sections of this paper are organized as follows:

The problem description and assumption is presented in section 3. In sections 4 and 5, we introduce the genetic algorithm and simulated annealing based heuristic respectively. Computational results are presented in section 6 and finally we draw some concluding remarks and possible future directions in section 7.

3. Problem description and assumptions

In this section, assumptions of the proposed problem are described in order to explain the structure of the proposed problem.

Set $G = \{g_1, g_2, \dots, g_N\}$ includes N groups where each group, g_i , consists of n_i jobs in the form $\{j_{i1}, j_{i2}, \dots, j_{in_i}\}$. Each job in each group is processed respectively in station C_1 and C_2 . All jobs and groups are processed in the same sequence on the machines between C_1 and C_2 . Each stage is able to perform a special operation and consists of several identical machines that located in the related station. All machines of a stage have the same speed and characteristics. m_1 and m_2 respectively represent the number of machines in stage 1 and 2, respectively. p_{1j} and p_{2j} demonstrates processing times of stage 1 and 2, respectively. Setup time for a specific group depends on the group which exactly precedes it on that machine. Every operation is processed only on one of the machines in the station. Each machine can handle one operation at any time. In order to establish no-wait condition, completion time of jobs in stage 1 should be equal to starting time of them in stage 2. Each machine can involve random breakdown which depends on the machines life. All machines cannot involve breakdown simultaneously at any time and at least one machine is available in both stages at any time for processing the jobs. The problem deals with two issues: machine assignment and scheduling of jobs. Interruption during the processing is not allowed. No preemption is allowed in the problem, meaning once a job starts to be processed on a machine, it cannot be interrupted before completion. All jobs are available at time zero. All jobs have equal importance or equal weights and finally no cancellation is permitted.

In this paper, bathtub curve as the machine's life curve has been utilized in order to consider the probability of machine breakdowns. This curve includes three different time periods: The first period has a decreasing failure rate, known as early failures. Generally, initial failures and breakdowns in the machines takes place in the first period. Regarding the type of the machines being used, different probability functions can be employed. In this paper, we have not considered this part because in reality breakdowns in this period are not common and occur seldom in this time period, so the failure probability is equal to zero for the first period. The second period consists of a constant failure rate, known as random failures. Useful life period is the other name mentioned for this time period. Various probability function can be employed in this time zone. In this article we have used the negative exponential probability distribution as follows:

$$f(t) = \lambda e^{-\lambda t}, \quad (1)$$

where λ is the failure ratio for machines and t is machine's life according to year. The third period includes an increasing failure rate, known as wear-out failures. This period usually happens after the useful time period of the machine and it is called exhaustion or oldness period. The Weibull probability distribution is employed for the mentioned part as follows:

$$f(t) = \alpha \beta t^{(\beta-1)} e^{(-\alpha*(t^\beta))}, \quad (2)$$

where α is the shape parameter and β is the scale parameter of the distribution.

d_k is the repairing time that required for a machine if a breakdown happens. In coding procedure we have considered $d_k = 16$ minutes as the repairing time.

4. Proposed GA approach to Solving Proposed Problem

Genetic algorithms (GA) were first used by Holland (1975) and generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection and crossover. Genetic algorithm is selected in this article due to its simple data structure and easy implementation for flowshop scheduling problems.

4.1 Chromosome representation

In genetic algorithm, each chromosome represents a point in the search space and also a possible solution for the problem. The representation used in this article follows the procedure proposed by Salmasi et al. (2010). It consists of two parts. The first part in the left side of the chromosome determines sequence of the groups and the one in the right side exhibits the job sequence.

2	1	3	11	8	6	10	9	4	5	7
---	---	---	----	---	---	----	---	---	---	---

Fig. 1. Representation of chromosome

The chromosome in Fig. 1 includes 3 groups indicated by 1, 2 and 3 where group 1 consists of jobs 4,5 and 6, group 2 consists of jobs 7,8 and 9 and finally group 3 is composed of jobs 10 and 11. Group 2 is the first group in the group sequence followed by groups 1 and 3.

4.2 Fitness function

The objective function value of all chromosomes are calculated and ordered in descending way. Fitness function chromosome i is calculated by Eq. (3) where FF_i , OF_w and OF_i represents fitness function for i th chromosome, the worst objective function available and objective function of current chromosome respectively. The equation is added by 1 in order to make it possible for the worst chromosome to be selected for the next population.

$$FF_i = OF_w - OF_i + 1. \quad (3)$$

4.3 Population

The evolution process of GA usually starts from a randomly generated population. In each generation, the fitness of every individual chromosome in the population is assessed, the more fitness function of an individual is, the more chance of being selected will be. New generation is stochastically selected from the current population, and each individual's genome is modified by means of mutation and crossover operator to form a new population. The new population is then used in the next iteration of the algorithm. The initial population for the proposed GA algorithm is generated randomly.

4.4 Mutation

In this paper, in order to perform a mutation operator, two genes are selected randomly and then their positions are replaced by each other. The mutation operator is performed on both parts of the chromosome.

3	1	2	4	6	11	7	14	8	5	13	9	10	12
3	4	2	1	6	11	7	14	8	5	13	9	10	12

Fig. 2. Mutation operator performed on first part of the chromosome

3	1	2	4	6	11	7	14	8	5	13	9	10	12
3	1	2	4	6	11	7	9	8	5	13	14	10	12

Fig. 3. Mutation operator performed on second part of the chromosome

4.5 Crossover operator

There are different types of crossover operators used in GA such as one-point, two-point, uniform and arithmetic ones. In this paper one-point crossover has been utilized.

4.6 Termination conditions

Different terminations conditions can be applied to a GA algorithm. For the proposed GA algorithm, generating a specified number of generations causes the algorithm to reach to its end.

5. Proposed SA approach to Solving Proposed Problem

SA as a generic probabilistic metaheuristic is revealed from the work of Kirkpatrick and Vecchi (1983). It is inspired from process of melting and refreezing materials. It does not always present an optimum solution, but it searches for a fair solution which can sometimes be optimum too. The search in SA is started with a randomized state. In a polling loop, the moves that decrease the energy will always be accepted, while bad moves will only accepted according to a probability distribution with probability of $e^{-\frac{df}{KT_k}}$ that dependent on the temperature of the system. In this probability distribution, T_k , K and df represent temperature degree, Boltzmann constant and the amount of degradation (the difference in the objective value between the current solution and the generated neighboring solution) respectively. If this probability is more than a uniform random number between 0 and 1 then the bad solution will be accepted.

5.1 Representation and neighborhood

Representation of solutions is the same as the one used in the GA algorithm. To create a new neighborhood, two genes are selected and exchanged with each other.

5.2 Movement generator

This generator's duty is creating next moves. It determines the situation of the next move by calculating the cost of current point and the next point.

5.3 Cooling schedule

Cooling schedule has a great impact on the success of the SA optimization algorithm. The parameters, which exist in a cooling schedule are the initial temperature, the equilibrium state, a cooling function, and the final temperature. Determination of the initial temperature is very important in accepting or rejecting the solutions. The higher the temperature, the more significant the probability of accepting a

worst move. On the other hand, low degrees of temperature reduces the acceptance chance of a bad solution and increases the chance of remaining in a local optima. In this article, we define the initial temperature as the worst objective function value in the initial population. There are different methods to decrease the temperature degree such as arithmetical, linear, geometric, logarithmic, very slow decrease and non-monotonic. In this paper we will use geometric method as follows:

$$t_i = \alpha t_{i-1} \quad (4)$$

Experience shows that the value of α should be more than 0.9 in order to reach to the best solution in a proper time.

5.4 Equilibrium state

To reach an equilibrium state at each temperature, a number of sufficient moves must be applied. This algorithm requires to be speculated in a specified temperature degree after some iterations to make a decision of continuing the annealing process in that degree or terminating the process and stepping to the other degree. In most SA methods a number of specified replacements are taken place in a temperature degree named epoch or period for assessing the equilibrium conditions. Number of these replacement is shown by N . We have employed following constraint to speculate equilibrium condition:

$$\left| \bar{f}_e - \bar{f}_e' \right| / \bar{f}_e' \leq \varepsilon, \quad (5)$$

where \bar{f}_e , \bar{f}_e' and ε stand for objective function average in last epoch for all of the accepted replacements, average of all amounts of \bar{f}_e and error respectively.

5.5 Termination condition

We have considered two termination conditions. First one is to reach to final temperature degree. The second is achieving to all of the generated neighborhoods or all of the accepted replacements during algorithm running time.

6. Computational Results

After denoting the sample problems, we ran the metaheuristic algorithms 10 times for each of the problems regarding their category and related optimum parameters. The results for computational exist in the tables. The Proposed GA and SA algorithms are coded in MATLAB 2013a on a computer with 4GB RAM, Intel Core2 Duo P7550 CPU, 2.4 GHz processor. In this paper, the statistical analysis performed by Minitab 16 software. Table 1 and Table 2 present the parameter values for small scale problems and large scale problems in genetic algorithm, respectively.

Table 3 and Table 4 exhibit the parameter values for small scale and large scale problems in simulated annealing. Table 5 show the generated problems for this proposed problem which are divided into two categories, problems with less than 10 jobs are in the small scale branch which are listed from problem number 1 to 24, and problems with more than 10 jobs are assigned to large scale branch which are listed from problem number 25 to 33 in the table. Table 6 shows the results including objective function average value in each iteration of the software and average computational time for the generated problems.

Table 1

Parameter values for small scale genetic algorithm

Maximum Iteration	Mutation Probability	Crossover Probability	Initial Population
150	0.15	0.8	50

Table 2

Parameter values for large scale genetic algorithm

Maximum Iteration	Mutation Probability	Crossover Probability	Initial Population
250	0.1	0.8	70

Table 3

Parameter values for small scale simulated annealing

Maximum Iteration	Decreasing Rate	Neighborhood Size	Initial Population
100	0.99	30	40

Table 4

Parameter values for large scale simulated annealing

Maximum Iteration	Decreasing Rate	Neighborhood Size	Initial Population
100	0.9999	30	50

Table 5

Generated problems

	Number of groups×Jobs in each group	[Machine's life in the first stage]× [Machine's life in the second stage]	Number of machines in each stage
1	4×3	[2,4]×[3,5,7]	M1=2, M2=3
2	4×3	[2,4,6]×[3,5,7]	M1=3, M2=3
3	4×3	[2,4,6,8]×[3,5,7]	M1=4, M2=3
4	4×4	[2,4]×[3,5,7]	M1=2, M2=3
5	4×4	[2,4,6]×[3,5,7]	M1=3, M2=3
6	4×4	[2,4,6,8]×[3,5,7]	M1=4, M2=3
7	5×3	[2,4]×[3,5,7]	M1=2, M2=3
8	5×3	[2,4,6]×[3,5,7]	M1=3, M2=3
9	5×3	[2,4,6,8]×[3,5,7]	M1=4, M2=3
10	5×4	[2,4]×[3,5,7]	M1=2, M2=3
11	5×4	[2,4,6]×[3,5,7]	M1=3, M2=3
12	5×4	[2,4,6,8]×[3,5,7]	M1=4, M2=3
13	7×6	[2,4,6,8,10]×[3,5,7,9,11,3]	M1=5, M2=6
14	7×6	[2,4,6,8,10,12]×[3,5,7,9,11,3]	M1=6, M2=6
15	7×6	[2,4,6,8,10,12,2]×[3,5,7,9,11,3]	M1=7, M2=6
16	7×7	[2,4,6,8,10]×[3,5,7,9,11,3]	M1=5, M2=6
17	7×7	[2,4,6,8,10,12]×[3,5,7,9,11,3]	M1=6, M2=6
18	7×7	[2,4,6,8,10,12,2]×[3,5,7,9,11,3]	M1=7, M2=6
19	8×6	[2,4,6,8,10]×[3,5,7,9,11,3]	M1=5, M2=6
20	8×6	[2,4,6,8,10,12]×[3,5,7,9,11,3]	M1=6, M2=6
21	8×6	[2,4,6,8,10,12,2]×[3,5,7,9,11,3]	M1=7, M2=6
22	8×7	[2,4,6,8,10]×[3,5,7,9,11,3]	M1=5, M2=6
23	8×7	[2,4,6,8,10,12]×[3,5,7,9,11,3]	M1=6, M2=6
24	8×7	[2,4,6,8,10,12,2]×[3,5,7,9,11,3]	M1=7, M2=6
25	11×5	[2,4,6,8,10,12,2,4]×[3,5,7,9,11,3,5,7,9,11]	M1=8, M2=10
26	11×5	[2,4,6,8,10,12,2,4,6,8]×[3,5,7,9,11,3,5,7,9,11]	M1=10, M2=10
27	11×5	[2,4,6,8,10,12,2,4,6,8,10,12]×[3,5,7,9,11,3,5,7,9,11]	M1=12, M2=10
28	11×10	[2,4,6,8,10,12,2,4]×[3,5,7,9,11,3,5,7,9,11]	M1=8, M2=10
29	11×10	[2,4,6,8,10,12,2,4,6,8]×[3,5,7,9,11,3,5,7,9,11]	M1=10, M2=10
30	11×10	[2,4,6,8,10,12,2,4,6,8,10,12]×[3,5,7,9,11,3,5,7,9,11]	M1=12, M2=10
31	12×10	[2,4,6,8,10,12,2,4]×[3,5,7,9,11,3,5,7,9,11]	M1=8, M2=10
32	12×10	[2,4,6,8,10,12,2,4,6,8]×[3,5,7,9,11,3,5,7,9,11]	M1=10, M2=10
33	12×10	[2,4,6,8,10,12,2,4,6,8,10,12]×[3,5,7,9,11,3,5,7,9,11]	M1=12, M2=10

In order to evaluate the effectiveness of the proposed algorithms, we take advantage of Relative Percentage Deviation from the best solution (RPD). RPD is calculated by the following formula:

$$RPD = \frac{|Method_{sol} - Best_{sol}|}{Best_{sol}} \times 100 \quad (6)$$

where $Method_{sol}$ and $Best_{sol}$ equal to the solution for the given problem and the best solution obtained, respectively. Table. 7 shows the RPD values for the investigated problems. In this paper, to determine significant differences existed between proposed genetic and simulated annealing algorithms, non-parametric tests was used (because P-Value < 0.005 according to Fig. 4) and the results of the non-parametric tests can be seen in Fig. 5.

Table 6
Computational Results

	GA Answers		SA Answers	
	Ave. C_{max}	Ave. Time	Ave. C_{max}	Ave. Time
1	80	20.074	76.9	316.25
2	73.3	19.431	70.8	312.64
3	69.6	19.206	67.8	311.32
4	122.6	30.375	111.2	583.93
5	108.9	29.916	104	479.53
6	102.2	29.477	98.8	498.79
7	108	28.900	96.4	501.6
8	97.1	28.201	89.8	442.8
9	90.7	27.563	87.2	446.8
10	186.5	43.113	162	670.8
11	158.2	42.640	145	706.6
12	145.6	42.055	135.2	722
13	205.8	165.49	197.4	2497.8
14	187.5	149.27	183	2273.6
15	175.7	145.41	171.8	2324.2
16	256.2	190.64	249.4	3063.4
17	234	197.18	228	3117.8
18	217.7	208.05	212.8	3107.6
19	248.3	187.36	244.4	2933.6
20	227.7	193.18	222	3003.6
21	212.2	186.19	206.4	3529
22	309.9	242.04	303.2	4907.8
23	283.4	250.45	276	3878.6
24	264	261.79	258	3805.4
25	185.6	543.66	180.6	4826.6
26	165.4	564.6	162.2	4598.8
27	152.6	535.4	149.2	4581.4
28	376.6	2076.6	371	17351
29	336.2	2010.8	328.8	17357.2
30	308.8	204.2	303.6	17441
31	419.6	2629	413.6	20713.6
32	376.2	2517	368	20506.33
33	343.8	2505.6	339.6	21042

Table 7
RPD values

Problem number	GA RPD	SA RPD	Problem number	GA RPD	SA RPD	Problem number	GA RPD	SA RPD
1	0.040312	0	12	0.076923	0	23	0.026812	0
2	0.035311	0	13	0.042553	0	24	0.023256	0
3	0.026549	0	14	0.02459	0	25	0.027685	0
4	0.102518	0	15	0.022701	0	26	0.019729	0
5	0.047115	0	16	0.027265	0	27	0.022788	0
6	0.034413	0	17	0.026316	0	28	0.015094	0
7	0.120332	0	18	0.023026	0	29	0.022506	0
8	0.081292	0	19	0.015957	0	30	0.017128	0
9	0.040138	0	20	0.025676	0	31	0.014507	0
10	0.151235	0	21	0.028101	0	32	0.022283	0
11	0.091034	0	22	0.022098	0	33	0.012367	0

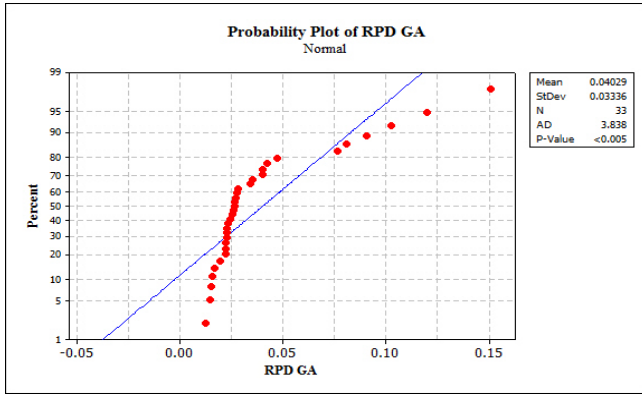


Fig. 4. Normality test for proposed GA RPD (small and large problems)

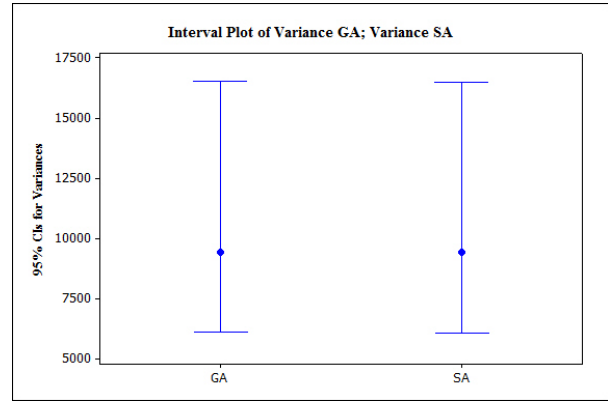


Fig. 5. The results from the test of equal variances for obtained makespan from suggested algorithms (small and large problems)

Table 8

Results from the Mann-Whitney Test for obtained makespan from suggested algorithms (small and large problems)

Median for GA Makespan	Median for SA Makespan	Point Estimate	95% Confidence Interval	P-value
187.50	183.00	5.90	(-40.29; 55.44)	0.71

According to the obtained results from Mann-Whitney test (Fig.6), it can be seen that at confidence level of 95%, we cannot reject hypothesis of the same of independent samples distribution (small and large problems). According to the obtained results from Kruskal-Wallis test (Fig.7), it can be seen that at confidence level of 95%, we cannot reject hypothesis of the equal of obtained makespan from two samples (small and large problems).

Table 9

Results of the Kruskal-Wallis Test for obtained makespan from suggested algorithms (small and large problems)

Median for Factor 1	Median for Factor 2	Average Rank for Factor 1	Average Rank for Factor 2	P-value
187.50	183.00	34.4	32.6	0.705

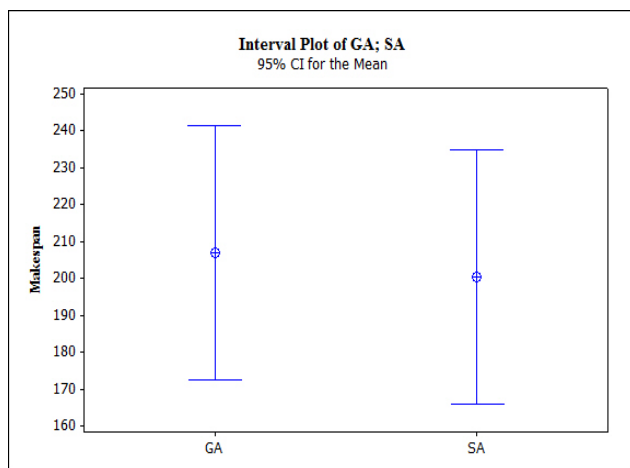


Fig. 6. Interval plot for obtained makespan from suggested algorithms (small and large problems)

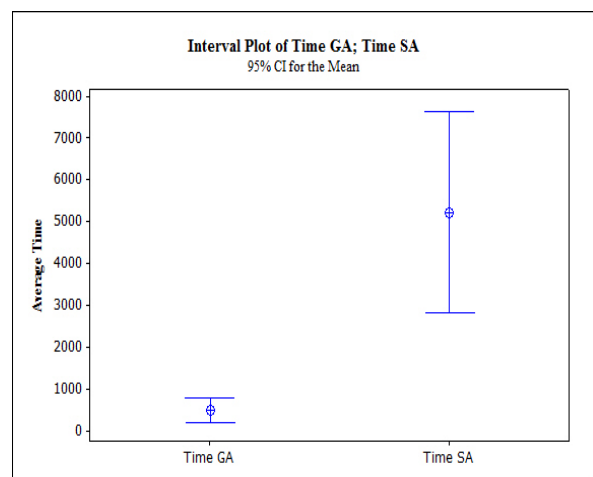


Fig. 7. Interval plot for obtained time from suggested algorithms (small and large problems)

According to the values obtained from Non- Parametric tests (Fig. 6 and Fig. 7) and obtained interval plot (Fig. 8), it can be concluded there was no significant differences between the obtained values (especially for small problems) from the proposed algorithms. According to the obtained values, performance of the proposed algorithms for solving proposed problem is proved.

Considering that the proposed issue in this article has been studied for the first time, comparing the results with the ones existing in other papers is not possible at the moment. Therefore, in order to validate the results, we consider a small problem including two working groups and two jobs in each group. Working groups are identified as group 1 and group 2 where in order to simplify them, we use numbers 1 and 2 for representing the groups. Also, numbers 4, 5, 6 and 7 stand for the jobs sequentially. Now, we take into account all of the possible permutations of the working groups and also all of the possible permutation of the jobs which are equal to $2!$ And $4!$ consecutively. The following figure displays an instance of above-mentioned permutation related to groups and jobs.

2	1	4	6	3	5
---	---	---	---	---	---

Fig. 8. An instance of the sequence for the proposed example

For each sequence of the example, which includes $4! \times 2! = 48$ sequences, we calculate the lower bound and compare it with the results obtained by proposed meta-heuristic algorithms. In this problem the smallest lower bound's value is equal to 100 which is exactly the same value obtained by meta-heuristic for this problem.

7. Conclusion

In this article, we have investigated $FF_2(m_1, m_2)/nwt, fmls, S_{plc} /Cmax$ problem considering random breakdown of the machines. Genetic algorithm and simulated annealing were implemented to solve the problem. Two scales were discussed including small scale problems and large scale ones. Computational results show that simulated annealing is better than genetic algorithm in the point of average of obtained makespan and on the other hand, genetic algorithm excels simulated annealing in the point of computational time especially in large scale problems.

For future work these method can be compared to some other layouts such as cellular manufacturing or process layout in order to define the efficiency of the model. In this article, all of the variables have deterministic values. In the real world many of parameters possess nondeterministic nature, so the fuzzy approach can be applied to the problem. Also, other metaheuristic methods such as particle swarm optimization or ant colony optimization can be developed for the proposed model and the results can be compared. Considering multi stages for the proposed problem and incorporating new constraints such as different methods for repairing machines can be another future works for the readers.

Acknowledgement

The authors would like to thank the anonymous referees for constructive comments on earlier version of this paper.

References

- Aldowaisan, T., Allahverdi, Ali. (2003). New heuristics for no-wait flowshops to minimize makespan. *Computers & Operations Research*, 30, 1219–1231.
- Ahmadizar, F., Ghazanfari, M., & Fatemi Ghomi, S.M.T. (2009). Application of chance-constrained programming for stochastic group shop scheduling problem. *International Journal of Advanced Manufacturing Technology*, 42, 321–334.
- Attar, S.F., Mohammadi, M., & Tavakkoli-Moghaddam, R. (2011). A novel imperialist competitive algorithm for solving flexible flow shop scheduling problem to minimizing maximum completion time. *International Journal of Computer Applications*, 28(10), 27-32.
- Ashhari, A. (2012). Group scheduling in flexible flowshop, master degree thesis, K. N Toosi University of Technology. (In Farsi).
- Gholami, M., Zandieh, M., & Alem-Tabriz, A. (2009). Scheduling hybrid flow shop with sequence-dependent setup times and machines with random breakdowns. *International Journal of Advanced Manufacturing Technology*, 42, 189–201.
- Gupta, J. N., & Stafford, E. F. (2006). Flowshop scheduling research after five decades. *European Journal of Operational Research*, 169(3), 699-711.
- Gupta, J.N.D. Strusevich, V.A., & Zwaneveld, C.M. (1997). Two-stage no-wait scheduling models with setup and removal times separated. *Computer Operation Research*, 24, 1025–1031.
- Jinxing, X., Xijun, W. (2005). Complexity and Algorithms for Two-Stage Flexible Flow shop Scheduling with Availability Constraints. *Computer and Mathematics with Application*, 50, 1629-1638.
- Salmasi, N. (2005). *Multi-stage group scheduling problems with sequence dependent setups*. Ph.D. thesis. Department of Industrial and Manufacturing Engineering, Oregon State University.
- Hall, N.G., Sriskandarajah, C. (1996). A survey of machine scheduling problems with blocking and no-wait inprocess, *Operations Research*, 44, 510-525.
- Jolai, F., Rabiee, M., & Asefi, H. (2012). A novel hybrid meta-heuristic algorithm for a no-wait flexible flow shop scheduling problem with sequence dependent setup times. *International Journal of Production Research*, 50(24), 7447-7466.
- Junlin, C. Weiwu, Y., & Huihe, S. (2004). Scheduling a two-stage no-wait hybrid flowshop with separated setup and removal times. *American Control Conference Boston*. Massachusetts.
- Lin, H. T., Liao, C. J. (2003). A case study in a two-stage hybrid flow shop with setup time and dedicated machines. *International Journal of Production Economics*, 86, 133–143.
- Liu, Z., Xie, J., Li, J., & Dong, J. (2003). A heuristic for two-stage no-wait hybrid flowshop scheduling with a single machine in either stage. *Tsinghua Science and Technology*, 8(1), 43-48.
- Logendran, R, Salmasi, N., & Sriskandarajah, C. (2006). Two-machine group scheduling problems in discrete parts manufacturing with sequence dependent setups. *Journal of Computers and Operations Research*, 33, 158-180.
- Mirabi, M., Fatemi Ghomi, S.M.T., & Jolai, F. (2013). A two-stage hybrid flowshop scheduling problem in machine breakdown condition. *Journal of Intelligence Manufacturing*, 24, 193–199.
- Moradinasab, N., Shafaei, R., Rabiee, M., & Mazinani, M. (2012). Minimization of maximum tardiness in a no-wait two stage flexible flow shop. *International Journal of Artificial Intelligence*, 8, 516-529.
- Wang, Z., Xing, W., & Bai, F. (2005). No-wait flexible flowshop scheduling with no-idle machines. *Operations Research Letters*, 33, 609 – 614.
- Ruiz, R., Maroto, C., & Alcaraz, J. (2005). Solving the flowshop scheduling problem with sequence dependent setup times using advanced metaheuristics. *European Journal of Operational Research*, 165, 34–54.
- Schaller, J. (2001). A new lower bound for the flow shop group scheduling problem. *Computers and Industrial Engineering*, 41, 151-161.

- Samarghandi, H., ElMekkawy T.Y. (2012). A genetic algorithm and particle swarm optimization for no-wait flow shop problem with separable setup times and makespan criterion. *International Journal of Advanced Manufacturing Technology*, 61, 1101–1114.
- Salmasi, D., Logendran, R., & Skandaria, M.R. (2011). Total flow time minimization in a flowshop sequence-dependent group scheduling problem. *Scientia Iranica* ,18, 759-764.
- Shafaei, R., Rabiee, M., & Mirzaeyan, M. (2011). An adaptive neuro fuzzy inference system for makespan estimation in multiprocessor no-wait two stage flow shop. *International Journal of Computer Integrated Manufacturing*, 24, 888-899.