Solving Group Scheduling Problem in No-wait Flow Shop with Sequence Dependent SetupTimes

Abolfaz Adressi[*1], Reza Bashirzadeh[1], Vahid Azizi[1], Saeed Tasouji Hassanpour[2]
[1*]Department of Industrial Engineering, K.N.Toosi University of Technology, Iran
[*]Email of Corresponding Author: aadressi@mail.kntu.ac.ir
[1]Department of Industrial Engineering, K.N.Toosi University of Technology, Iran
[1]Department of Industrial Engineering, K.N.Toosi University of Technology, Iran
[2]Faculty of Industrial Engineering, Tarbiat Modares University (TMU), Iran, Tehran
Received: 29 January, 2014; Accepted: 23 March, 2014

Abstract
Different manufacturing enterprises use regularly scheduling algorithms in order to help meeting demands over time and reducing operational costs. Nowadays, for a better useofresources and manufacturingin accordance withcustomer needs and given the level ofcompetitionbetweencompanies, employing asuitablescheduling programhasa double importance. Conventional productionmethods are constantly substituted with new ones for improving the efficiency and effectiveness of the entire production system. In this paper, two Meta-heuristic algorithms, Genetic and simulated annealing, have been used in order to solve the group scheduling problem of jobs in a single stage No-wait flow shop environment in which setup times are sequence dependent,. The purpose of solving the proposed problem is to minimize the maximum time needed to complete the jobs (Makespan). The results show that Genetic algorithm is efficient in problems with small and large dimensions, with respect to time parameter of problem solving.

Keywords
Group Scheduling, No-wait Flow Shop, Sequence Dependent Setup Times, Metaheuristic Algorithms.

1. Introduction

Existingcompetitions among manufacturing companies requires them to be more efficient and flexible in the face of these competitions. Manufacturing enterprises are constantly discovering and using innovative methods in order to overcome the abnormalities created in their manufacturing environments. Moreover, conventional productionmethods are constantly substituted with new methods for improving the efficiency and effectiveness of the entire production system. Different manufacturing enterprises frequently use scheduling algorithms in order to help observing the needs and requirements of the customer over time and reducing operational costs. One of these methods is CellularManufacturing method (CM). In 1970, a method of production called cellular manufacturing was developed.

In cellular manufacturing, pieces are allocated to different groups based on their similarities in shape, material or similarity in processing operations. Machines are also allocated to different cells in order to separate the production line. Then, groups of pieces are allocated to a specific cell for production. Each cell consists of several machines which are capable of doing necessary operations for making groups pieces. This allocation of machines and jobs has many benefits such as significantly decreasing the set up time as well as the supplies of processing works, and simplifying the flow of pieces and tools [1].

Sequencingandscheduling are decision problems which play a fundamental role in manufacturing andservice industries [2]. These two issues have been used for improving the efficiency of production flow since the beginning of the last century. Therefore, the next step for increasing the efficiency of production is finding the best processing sequence of allocated groups to the cell and also allocated jobs to a group in order to minimize or maximize some of the considered criteria. This subject of study is called group scheduling. The purpose of this paper is minimizing the maximum job completion time. The purpose of minimizing and maximizing of completion time of jobs is minimizing the completion time of working groups and also the jobs of the groups on the last machine. In the group scheduling problems, all jobs which belong to a group need similar set up time on machines. Therefore, a major and important set up time is needed for processing each group on each machine. Setup time operations of a group consist of machine preparation, providing the tools needed, settings related to the required Jig and fixtures, material inspection and their cleanings [3], which should be investigated as a separated operation on the machines for some problems instead of considering it as a part of the processing time. Scheduling problems consisting of separable setup times are divided into two major groups: sequence dependent scheduling and sequence independent scheduling. If the setup time of a group for each machine depends on the immediate prior processed group on which the machine was processed, the problem is put in the category of "sequence dependent group scheduling" problems; otherwise, the considered problem is called "sequence independent group scheduling". In 1992, Wortman explained the importance of considering the sequence dependent setup times for effective managementof production capacity [4]. There are many sequence dependent scheduling problems in the real world.

In this paper, group scheduling problem in a single phase No-wait flow shop with consideration of the limitation of sequence dependent setup time for groups has been investigated and for solving this problem meta-heuristic Genetic algorithms and simulated annealing have been used. The rest of the article is as follows:

In the second section, the extant literatures relevant to the subject of the present paper have been reviewed. In the third section, the problem of study and also its assumptions have been defined. The fourth and fifth sections are related to the solution of the proposed problem using meta-heuristic Genetic algorithms and simulated annealing. The sixth section presents the specifications of the proposed problems. In the seventh section, the results obtained by performing meta-heuristic algorithms are presented. The eighth section addresses the validity of the proposed algorithms in this paper and finally, the conclusion and suggestions for future research are presented in the last section.

2. Literature review

Scheduling problems with no waiting occurred in those categories of production environments in which a work should be processed No-wait on a machine or between machines frombeginning to the end. The reason of occurrence of such environments is the kind of technology or lack of storage capacity between machines and workstations. For example, temperature, density or other factors cause each operation to follow its prior operation immediately. To illustrate, in steel production, when molten steel is exposed to some sequential operations like fusion, casting and rolling, such a condition occurs. Also, in food industry in order to ensure having fresh productions, the operation of putting food production in conserve cans should be immediately performed right after cooking. This also happens in pharmaceutical, chemical, petrochemical, and service industries.

First works in No-wait scheduling subject can be related to Artanari's works in 1971 and 1974 [5]. The No-wait flow shop scheduling problems with more than two machines are categorized in Np-Hard problems [6]. A large number of researches have been done for solving the No-wait flow shop scheduling by considering different criteria like the maximum completion time of works and time in total flow which have led to presenting many heuristic and meta-heuristic algorithms. As an example, we can point out the research performed in the No-wait flow shop field with three machines which was began by Pihler in 1960 and continued till 1993 by Gangadharan and lajendran. The main concentration of these kinds of research was on present heuristic algorithms [7]. In 1990, Rajendran and Chaudhuri, by consideration of time criterion in flow, presented two heuristic algorithms for solving the mentioned problem. They used two heuristic algorithms for the development of the primary sequence of jobs and then improved this sequence by works entrance method which was presented by Nawaz-Enscore-Ham (NEH) [8]. In 1996, Hall and Sriskandarajah comprehensively reviewed the No-wait scheduling problems of machines. In this paper, they explained the application of this problem in industries and investigated the computational complexities. They also studied the results and performance of the existing algorithms and presented some suggestions for the next studies [9]. In another research in 1998 which was performed on the No-wait flow shop problems, Aldowaisan and Allahverdi investigated a problem in which when a job started its processing from machine 1 in the production line, the work had to go through the whole production line without any delay so that the works should not wait not only between machines but also on the machines. They called this problem No-wait flow shop problem [10]. In 2006, Gupta and Stafford performed a comprehensive review on the No-wait flow shop scheduling problems over the last 50 years [11]. There is a great body of available research which was performed in No-wait flow shop scheduling problems and still is continued so that as one of the last works is the research done by Pang in 2012 [12]. He studied the No-wait double machine flow shop scheduling considering the group setup times.

Group scheduling contexts arose at the beginning of the 20^{th} century which my be due to the reduction of setup times as pointed out by Mitrofanov in 1966 and Burbidge in 1975 [13]. The first method in optimizing a double machine sequence dependent group scheduling problem was studied by Ham, Hitomi and Yoshida in 1985 [14]. A double machine flow shop scheduling problem was shown by Baker in 1999 and Sekiguchi in 1983 in which each group has a setup time. Schaller, Gupta and Vakharia in 2000 and Reddy and Narendran in 2003 showed heuristic algorithms for solving the sequence dependent flow shop group scheduling problem by consideration of different conditions like unavailability of all jobs at the beginning [15, 16] .

In 2011, Karimi et al. studied a group scheduling problem in flexible flow shop by consideration of sequence dependent preparation times and objective function of minimization of maximum completion jobs time. In order to solve the intended problem, they presented a hybrid ICA algorithm and semi-electromagnetic mechanism. Their obtained results show that the presented algorithm is better than the other existing algorithms in the literature review [17].

3. Definition of the problem of study

In this paper, the group scheduling problem in a single stage No-wait flow shop by consideration of sequence dependent setup times for working groups on machines has been studied, and the existing machines in this step are parallel to each other. The assumptions and the structure of this problem can be stated as follows:

1- the set of $G=\{g_1, g_2, ..., g_N\}$ consists of N groups which each g_i group includes n_i jobs as $\{j_{i1}, j_{i2}, ..., j_{in_i}\}$.

2- All jobs and groups in similar sequence on all machines are processed No-waitly (Sequential scheduling). For example, if a conveyor Belt is used No-waitly for transferring jobs between the machines, therefore all jobs should be transferred and processed in the same sequence between all the machines and stations.

3- All works in each group are available at the beginning of the scheduling program.

4- The number of machines in the studied station is shown by m.

5- The existing machines in the station are similar and the same.

6- The setup and preparation time of each group for each machine depends on the immediate prior processed group on which the machine has been processed (sequence dependent setup time), shown by S_{plc} which indicates that setup time of machines for processing of each group depends on the immediate prior processed group. The setup time of each group in each step may be different, but this time is similar for all the machines belonging to the same step.

7- In order to establish the No-wait condition, the jobs should be processed with no delay on machines.

4. Solving the studied problem using Genetic algorithm

In order to solve the proposed problem in this paper, meta-heuristic Genetic and simulated annealing algorithms have been used. The Genetic algorithm is a search technique and optimization based on the Genetic principals and natural selection. Genetic algorithm allows that a population consisting of a large number of people based on some specific rules maximize competency (which means that it minimizes the cost function). This method was introduced by John Holland in the 1960s and 1970s and finally by David Goldberg, who could solve the difficult problem of controlling of gas transfer pipe lines. Holland was the first one who tried, by using his theory called Schema Theorem, to extend a theoretical base for Genetic algorithm. Dejong research (1975) showed the Genetic algorithm benefits for optimization which was considered as the first integrated attempt for finding optimized Genetic algorithm parameter. It may be said that Goldberg significantly contributed else to Genetic algorithm. After that, many of the evolutionary scheduling was tested by different successes.

In the following, different parts of this algorithm like chromosome structure and also the manner of calculation of the fitness function and Genetic algorithm operators used for creating the first population in this paper have been explained.

4.1 ChromosomeStructure

In Genetic algorithms each chromosome is an indicator of one point in the search space and a possible solution for the interested problem. The chromosomes themselves (solutions) consist of some constant genes (variable). Considering the solution space of the problem of study and that this problem is a combination of two assumptions of group scheduling and No-wait in scheduling problems, different kinds of coding can be suggested for the problem. However, Salmasi et al. investigated two coding approaches for representation of the initial solution. In the first approach, a sequence of groups and jobs in the first step is an indicator of an initial solution, while in the second approach, for indicating the initial solution, the groups allocated to each machine, group sequence on each machine and also the sequence of jobs in groups in the first step should be specified. The results of this research show that the application of the first approach gives better solutions [18]. Therefore, this approach has also been used in this paper for solution coding and making a suitable chromosome. An example of the structure of a chromosome for presentation of an initial solution in the problem of study has been shown in figure 1.

2	1	3	11	8	6	10	9	4	5	7

Fig. 1. Representation of the suggested chromosome paper

The suggested chromosome in the above figure is related to three working groups. The number of the working groups is equal to 1, 2, and 3. Group 1 consists of jobs 4, 5, and 6, group 2 consists of jobs 7, 8, and 9, and group 3 consists of jobs 10 and 11. The left side of the above chromosome includes defining of the working groups sequence. In this example, working groups sequences are respectively equal to working groups 2, 1, and 3. The right side includes defining the existing jobs sequence in working groups.

4.2 Crossover operator

Intersection operator is making one or some children by the selected parents in the mating process. The most common form of intersection consists of two parents that produce two children. The purpose is producing new child expectiong that the good specialties of two parents gather in their children and produce a better solution. In this paper, single point intersection has been used for combining the chromosomes. In the employed single point intersection, first, an accidental point in the first or second part of the chromosome has been selected, the sequence of jobs (or groups) in the left side of the point in parent for its corresponding child has been protected and the sequence of jobs (or groups) in the right side of the point based on the other parent are put in order. This scheme makes a good assurance for appropriate search of solution space of the problem. Since this method produces infeasible chromosomes for the problem, it is necessary to do the following scheme for producing feasible chromosomes:

- Copy the genes of parent 1 from the beginning to the breakpoint in child 2.
- Delete the corresponding genes of duplicated genes from parent 2.
- Copy the genes of parent 2 from the beginning to the breakpoint in child 1.
- Delete the corresponding genes of the duplicated genes from parent 1.
- Put the remaining genes of parent 1 to empty genes of child 1.
- Put the remaining genes of parent 2 to empty genes of child 2.

The experiences show that the intersection operator rate should be about 0.65 to 0.85 [19].

4.3 Mutation operator

Some factors in nature, like ultraviolet radiation, cause unpredictable changes in chromosomes. Since the Genetic algorithm follows evolution principles, in these algorithms, the mutation operator is also applied with low possibility. Mutation causes searching in the un-manipulated spaces of the problem. It can be concluded that the most important task of mutation is avoidance from converging to local optimization. The applied mutation operator in this paper is an exchange mutation in which two genes are accidentally selected alongside the chromosome and their values are exchanged with each other. In this paper, the mutation operator is applied like the intersection operator for each two parts of the chromosome. This approach creates a good assurance for an appropriate search of solution space of the problem. This is a very small rate which is usually considered between 0.01 to 0.05 for binary gens and 0.05 to 0.2 for numerical gens [19].

4.4 Genetic algorithm parameter adjustment

In this paper, the typical problems are divided into two small and large categories and they have been tested. In order to adjust the parameters related to the Genetic algorithm used in this paper, Taguchi method has been applied and the optimization parameters for small and large problems are shown in tables (1) and (2):

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

5. Solving the problem using simulated annealing algorithm

Simulated annealing algorithm is a strong solution technique which produces very good solutions for one and multi-objective optimization problems. This algorithm, by constructing and evaluating the sequential solutions, moves step by step toward the optimum solution. For movement, a new neighborhood is accidentally created and evaluated. In this method, the points near the given point in search space are studied. In case the new point is a better point, it is selected as the new point in the search space and if it is worse, based on a probability function it will be selected.

5.1 Simulated annealing algorithm parameter adjustment

Considering that in this paper, the simulated annealing algorithm based on population has been used. As such, this algorithm consists of 4 initial population variable parameters, number of neighborhood, the rate of temperature decrease and maximum iteration. So, the adjustment of parameters related to small and large problems proposed in this paper are as follows:

Table. 3. Parameter values for small scale simulated annealing

Maximum Iteration	Decreasing Rate	Neighborhood Size	Initial Population
100	0.95	30	40

Table. 3. Parameter values for large scale simulated annealing

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

6. Numerical experiments

Table. 5. Generated problems (small problems)

Problems	Number of groups×Jobs in each group	Number of machines in stage	Problems	Number of groups×Jobs in each group	Number of machines in stage
1	4×3	2	**7**	7×6	5
2	4×3	3	**8**	7×6	6
3	4×3	4	**9**	7×6	7
4	4×4	2	**10**	7×7	5
5	4×4	3	**11**	7×7	6
6	4×4	4	**12**	7×7	7

Table. 6. Generated problems (large problems)

Problems	Number of groups×Jobs in each group	Number of machines in stage	Problems	Number of groups×Jobs in each group	Number of machines in stage
13	10×5	8	**18**	10×10	12
14	10×5	10	**19**	15×10	8
15	10×5	12	**20**	15×10	10
16	10×10	8	**21**	15×10	12
17	10×10	10			

It should be noted that processing time of jobs in this paper for small and large problems and sequence time setup times for them , uniformly, are as (5-75) and (5-25), respectively.

7. Computational results

After adjusting the parameter and obtaining the optimized parameters using Taguchi method, each proposed problems are run 5 times by Matlab software and then we concentrate on the average of the obtained solutions. The obtained results can be seen in tables (7) and (8):

According to the obtained results by running metahuristic Genetic algorithms and simulated annealing and comparing them, it can be concluded that the Genetic algorithm for problems with small dimensions is effective and efficient and for problems with large dimensions also presents acceptable solutions in acceptable times.

Table. 7. Computational Results (Genetic Algorithm)

Number of Problems	Answers Obtained from Genetic Algorithm					Number of Problems	Answers Obtained from Genetic Algorithm				
	Ave C_{max}	Stdev	Worst Ans	Best Ans	Ave Time		Ave C_{max}	Stdev	Worst Ans	Best Ans	Ave Time
1	56	0	56	56	11.322	12	97.2	0.4472	98	97	106.99
2	38	0	38	38	11.052	13	88	0	88	88	255.045
3	28.4	0.5477	29	28	10.752	14	71	0.7071	72	70	260.152
4	86	0	86	86	16.087	15	59.8	0.8367	61	59	246.408
5	58	0	58	58	16.56	16	198.2	0.4472	199	198	1177.73
6	43.2	0.4472	44	43	16.152	17	159.2	0.4472	160	159	1162.37
7	109	0	109	109	80.012	18	133.2	0.4472	134	133	1312.12
8	91	0	91	91	76.576	19	291.4	0.5477	292	291	2222.30
9	78	0	78	78	78.430	20	233.2	0.4472	234	233	2179.02
10	135	0	135	135	103.63	21	195.25	0.5	196	195	2282.95
11	113	0	113	113	104.41						

Table. 8. Computational Results (Simulated Annealing Algorithm)

Number of Problems	Answers Obtained from Simulated Annealing					Number of Problems	Answers Obtained from Simulated Annealing				
	Ave C_{max}	Stdev	Worst Ans	Best Ans	Ave Time		Ave C_{max}	Stdev	Worst Ans	Best Ans	Ave Time
1	56	0	56	56	140.95	12	91.6	0.8944	93	91	1770.328
2	38	0	38	38	139.25	13	87	0	87	87	1917.256
3	27.3	0.6745	29	27	156.44	14	68.8	0.8366	70	68	2018.78
4	86	0	86	86	224.42	15	59.2	1.3038	61	58	1993.85
5	58	0	58	58	223.56	16	194.4	0.5477	195	194	8573.875
6	42	0	42	42	227.74	17	155.2	1.3038	157	154	9205.97
7	107.3	0.674	109	107	1.68.16	18	131	1	132	130	9572.466
8	91	0	91	91	1045.2	19	288.2	2.8635	292	285	15644.99
9	75.7	1.252	78	75	1141.1	20	224.6	1.8165	227	223	16386.23
10	133.3	0.483	134	133	1439.6	21	190.8	0.8366	192	190	17715.69
11	105	0	105	105	1503.5						

In this paper, in order to evaluate the efficiency of each proposed algorithm, Relative Percentage Deviation (RPD) from best solution according to reference [20] has been used. This criterion is calculated by the following formula:

$$RPD = \frac{|\,Method_{sol} - Best_{sol}\,|}{Best_{sol}} \times 100 \tag{1}$$

where $Method_{sol}$ is equal to the obtained solution for the problem in each proposed algorithms which is the average of 5 time running of algorithms for each problems and $Best_{sol}$ is equal to the best solution obtained for the problem produced by proposed algorithms. Table (9) shows the calculated RPD for the problems of interest.

Table. 9. RPD Values for Sample Problems

Number of problems	RPD_{GA}	RPD_{SA}	Number of problems	RPD_{GA}	RPD_{SA}
1	0	0	12	0.0681	0.0066
2	0	0	13	0.0115	0
3	0.0519	0.0111	14	0.0441	0.0118
4	0	0	15	0.0310	0.0207
5	0	0	16	0.0216	0.0021
6	0.0286	0	17	0.0338	0.0078
7	0.0187	0.0028	18	0.0246	0.0077
8	0	0	19	0.0225	0.0112
9	0.04	0.0093	20	0.0457	0.0072
10	0.0150	0.00226	21	0.0276	0.0042
11	0.0762	0			

In the above table, RPD_{GA} and RPD_{SA} indicate RPD of Genetic and simulated annealing algorithms respectively.

8.Validation of the proposed algorithms

In this paper, in order to ensure the correctness of the obtained solutions from the proposed algorithms, two methods have been used. One of these methods of validation of the proposed algorithms is that we consider a small problem consisting of two working groups and two jobs in each group, with the number of working groups being equal to 1 and 2 and the number of jobs in the groups being equal to 3, 4, 5 and 6. Then all the permutations related to the working groups which here will be 2! And also all the permutations related to works in both working groups which here will be 4! Have been considered. Figure 2 shows an example of permutations related to groups and jobs:

2	1	4	6	3	5

Fig. 2. Example of Sequence Related to Problem Validation

It should be noted that the processing time of jobs in this example for small and large problems and sequence dependent setup times for them uniformly are (5-75) and (5-25) respectively.

For each sequence related to this typical example, which consists of 48 sequences for this problem, the lower bound has been calculated and the results of these obtained lower bounds are

compared with solutions obtained from the proposed Genetic algorithm for this problem. In this problem, the smallest value of the obtained lower bound is equal to 100 that was exactly consistent with the solutions obtained by the proposed Genetic algorithm for this problem.
Another method which was used in this paper for validation of the proposed algorithm is that the proposed problem is solved by refrigeration simulation algorithm and then we compare the obtained solutions of each algorithm. For this purpose, considering the abnormality of the distribution of the obtained results produced by the proposed algorithms, in order to define the equality of the average of the obtained solutions by the two proposed algorithms in the safety level of 95% using Minitab 16 software, we used nonparametric Kruskal Wallis. The obtained solutions produced by this test (equality of the average of the obtained solutions produced by the proposed algorithms) are against one test (inequality of the average of the obtained solutions produced by the proposed algorithms). The produced solutions by Kruskal Wallis test can be seen in figure (3):

Kruskal-Wallis Test: Response versus Factor

```
Kruskal-Wallis Test on Response

Factor     N  Median  Ave Rank      Z
1         21   91.00      22.0   0.25
2         21   91.00      21.0  -0.25
Overall   42              21.5

H = 0.06  DF = 1  P = 0.801
H = 0.06  DF = 1  P = 0.801  (adjusted for ties)
```

Fig. 3. Obtained results from Kruskal Wallis test

9. Conclusion and suggestions for next studies

In this paper, meta-heuristic Genetic algorithms and simulated annealing for solving the group scheduling in No-wait flow shop by consideration of sequence dependent set up times for working groups have been proposed. For performing the operations related to the proposed algorithms, Matlab 2010 software has been used. In order to evaluate the efficiency of the proposed algorithms, Relative Percentage Deviation from the best solution criterion (RPD) has been used. Also for comparing the two algorithms according to the equality of the average of the obtained solutions by each method, the nonparametric Kruskal Wallis test has been used. The obtained solutions show that the proposed Genetic algorithm in solving the proposed problem in this paper is efficient, especially in problems with large dimensions.
The solution of the proposed problem using different meta-heuristic Genetic algorithms and comparing the obtained solutions by them with the solution presented in this paper and also solving the proposed problem as multi step are suggested as future studies.

10. References

[1] Logendran, R. 2002. "Class Notes for Design and Scheduling of Cellular ManufacturingSystems "(1E564), Oregon State University.

[2] Pinedo, M. 2002. Scheduling Theory Algorithms and Systems, Prentice Hall.

[3] Allahverdi, A., Gupta, J.N.D., Aldowaisian, T. 1999. "A Review of SchedulingResearch Involving Setup Considerations ".Omega, International Journal ofManagement Science, 27, 219-239.

[4] Wortman, D.B. 1992. "Managing Capacity: Getting the most from your Firm's Assets". md Eng, 24, 47-49.

[5] Jinxing, X., Wenxun, X., Zhixin L., Jiefang, d. 2004. "Minimum Deviation Algorithm forTwo-Stage No-Wait Flowshops with Parallel Machines". An International JournalComputers and Mathematics with applications, 47, 1857-1863.

[6] Papadimitriou, C.H., Kanellakis, P. C. 1980. "Flowshop scheduling with limited temporarystorage". Journal of Associate Computer Machinery, 31, 343-357.

[7] Hejazi S.R and SAGHAFIAN S. 2004. "Flowshop-scheduling problems with makespan criterion: a review". International Journal of Production Researc,43, 2895-2929.

[8] Rajendran C. and Chaudhuri, D. 1990. "Heuristic algorithms for continuousflow-shop problem". Naval Research Logistic Quarterly, 37, 695-705.

[9] Hall, N.G., Sriskandarajah, C. 1996. "A survey of machine scheduling problems with blocking and no-wait inprocess", Operations Research, 44, 510-525.

[10] Aldowaisan, T., and Allahverdi, A. 1998. "Total flowtime in no-wait flowshops with separated setup times". Computers and Operations Research, 25(9), 757-765.

[11] Gupta, J. N., and Stafford Jr, E. F. 2006. "Flowshop scheduling research after five decades". European Journal of Operational Research, 169(3), 699-711.

[12] Pang, K. W. 2013. "A genetic algorithm based heuristic for two machine no-wait flowshop scheduling problems with class setup times that minimizes maximum lateness". International Journal of Production Economics, 141(1), 127-136.

[13] Kusiak, A., and Chow, W. S. 1987. "Efficient solving of the group technology problem". Journal of manufacturing systems, 6(2), 117-124.

[14] Ham, I., Hitomi, K., Yoshida, T. 1985. Group Technology, Kluwer-Nijhoff Publishing,Boston, MA.

[15] Schaller, J.E., Gupta, J.N.D., and Vakharia, A.J. 2000. "Scheduling a Flowline Manufacturing Cell with Sequence Dependent Family Setup Times", European Journal ofOperational Research, 125, 324-339.

[16] Reddy, V., Narendran, T.T. 2003. "Heuristics for Scheduling Sequence Dependent Set-upJobs in Flow Line Cells". International Journal of Production Research, 41(1), 193-206.

[17] Karimi N., Zandieh M., and Najafi A.A. 2011. "Group scheduling in flexible flow shops: a hybridized approach of imperialist competitive algorithm and electromagnetic-like mechanism". International Journal of Production Research, 49, 4965-4977.

[18] Rasti, S., Salmasi, N. 2011. " A SIMULATED ANNEALING ALGORITHM FOR SEQUENCE-DEPENDENT PROBLEMS INVOLVED IN FLEXIBLE FLOW SHOP GROUP SCHEDULING". Journal of Industrial Engineering & Management, 27(1), 75-91.

[19] Rabiee, M. 2011. "Solving two stage No-wait flexible flow shop". Thesis of Master Degree, Faculty of Industrial Engineering, K.N.Toosi University of Technology.

[20]Zandieh, M., and Karimi, N. 2011. "An adaptive multi-population genetic algorithm to solve the multi-objective group scheduling problem in hybrid flexible flowshop with sequence-dependent setup times". Journal of Intelligent Manufacturing, 22(6), 979-989.