International Journal of Engineering

Journal Homepage: www.ije.ir

Critical Path Method for Lot Streaming Problem in Flexible Job Shop Environment

B. Yousefi Yegane*[a], I. Nakhai Kamalabadi[a], N. Khanlarzade[b]

[a] *Faculty of Engineering, University of Kurdistan, Sanandaj, Iran*
[b] *Faculty of Industrial Engineering, Tarbiat Modares University, Tehran, Iran*

PAPER INFO

Paper history:
Received 19 August 2016
Received in revised form 25 January 2017
Accepted 25 January 2017

Keywords:
Memetic Algorithm
Flexible Job Shop Scheduling Problem
Lot Streaming
Critical Path Method

ABSTRACT

This paper addressed a Flexible Job shop Scheduling Problem (FJSP) with the objective of minimization of maximum completion time (C_{max}) which job splitting or lot streaming is allowed. Lot streaming is an important technique that has been used widely to reduce completion time of a production system. Due to the complexity of the problem; exact optimization techniques such as branch and bound algorithm will lose their efficiency at medium and large dimensions. Thus, metaheuristic techniques are good options in order to obtain high quality solutions within a reasonable computational time. In this study, firstly, the considered problems are solved in both permitted and not permitted of lot streaming by means of memetic algorithm and then the obtained solutions will be improved by using the critical path method heuristic. The numerical results indicate the high efficiency of the memetic algorithm in comparison with the previous methods; in addition, adding preemption and using allowable idle machines have led to a considerable improvement in the objective function.

***doi**: 10.5829/idosi.ije.2017.30.02b.13*

1. INTRODUCTION

The appropriate performance in on time delivery of product to the customer is a very important factor in modern competitive environments. Customers' tastes are always changing therefore under such conditions, utilization of flexible production systems and taking appropriate methods for planning and scheduling of operations is inevitable.

Flexible Job shop Scheduling Problem (FJSP) deals with scheduling for execution of a group of operations each of which includes one or more operations while any operation may be done over a group of machineries. Planning of operations is proposed to optimize one or more criteria as target function.

Flexible Job shop Scheduling Problem (FJSP) includes two major sub-problems as follows:

1. Allocation sub-problem that deals with dedicating any operation to a machine among possible machineries;
2. Scheduling sub-problem in which sequence of operation over machineries is determined.

In classic job shop scheduling problem and flexible job shop scheduling, it is assumed that an operation can be started when the former operation has been fully completed. This assumption is significant when any operation only consists of one sublot. Now, suppose that each operation includes a number of similar sublots (take, for example, producing 20 similar work pieces which comprise of 3 operations in manufacturing of any sublot). This mode is characterized as Lot-Streaming. Given the latter assumption, it is observed that execution of an operation will not necessarily require completion of the prior operation therefore the next operation can start; as a result, these operations will have overlapping time with each other. Overlapping of operation is a new assumption to which few papers have been referred and it plays essential role in reduction of completion time. In some of industries like petro-chemistry and glass industries, this assumption is one of the hypotheses which should be certainly taken into consideration. The assumption of overlapping time of operations means that starting of next operation is not necessarily subjected to completion of previous one, but

*Corresponding Author's Email: babak.yaganeh@uok.ac.ir (B. Yousefi Yegane)

after spending certain period of time as well as starting the former operation, the next operation can start. This assumption will cause more complexity in the Flexible Job shop Scheduling Problem [1].

2. LITERATURE REVIEW

2. 1. Literature Review of Flexible Job Shop Scheduling Problem

To solve FJSP problem, two approaches have been proposed: hierarchical approach and integrated approach. In hierarchical approach, the problem is converted into two sub-problems: Allocation problem and scheduling problem [2]. This separation will reduce complexity; nevertheless, the obtained results are also far from the optimal solutions. In integrated approach, allocation and scheduling are done simultaneously. Compared to hierarchical method, in this approach better solution will also result and the necessary time for acquiring the solutions is longer. In the following, initially we review on some of the conducted studies with hierarchical approach and then, we will explore the conducted studies with an integrated approach.

Brandimarte [3] was the first to apply hierarchical approach to solve FJSP problem; He utilized *dispatching rules* to solve allocation sub-problem and then solved scheduling problem by means of *Tabu Search* method. Brucker and Schile [4] have proposed a polynomial algorithm to solve FJSP problem with two operations. By assuming of overlapping operations, Fattahi et al. [2] solved the problem with simulated annealing algorithm. Saidi and Fattahi [5] presented a mathematical model for this problem in which the assumption of setup times dependency on the sequence of operation has been considered; they also employed tabu-search method to solve their mathematical model. Kacem et al. [6] proposed a genetic algorithm and developed an approach by localization in order to achieve initial assignment. Xia & Wu [7] employed Particle Swarm Optimization (*PSO*) to solve allocation problem and used simulated annealing algorithm and local search method for scheduling problem. Zandieh et al. [8] have adapted genetic algorithm to solve both allocation and scheduling problems. In their suggested algorithm, they have proposed several methods to create initial population and many strategies to reproduce a new generation at next phases.

Taking integrated approach, Hurink et al. [9] suggested Tabu Search algorithm to solve *FJSP* problem. Dauzere-Peres and Pauli [10] developed Disjunctive Graph Model for job shop scheduling problem and considered this fact that any operation should be completed by one machine in FJSP problem. A new neighborhood structure has been extracted with extension of the aforesaid disjunctive graph, and Tabu search was used to solve the problem. Mastrololli and Gambardella [11] developed the tabu-search method of Dauzere-Peres and Pauli [10] and proposed two neighborhood functions.

So far the preemption assumption has been explored by many researchers as well; however, FJSP problem has not been implied among few conducted studies, Runge and Sourd [12] have taken the assumption of preemption in doing operation with sharpness- tardiness minimization in scheduling problem. Hendel et al. [13] have studied on scheduling problem by supposing of existing interval as well as upon updated delivery with sharpness- tardiness minimization as a criterion. In this study, sharpness costs have been dependent on the starting time of operation. Hou et al. [14] studied scheduling for performing *n* independent operation on *m* parallel machines by aiming at minimization of time for completion and by assuming the possible preemption in operation. Farughi et al. [15] have proposed a memetic algorithm to solve FJSP problem and then optimized the given answers with introducing a critical path based-heuristic technique.

Thammano and Phu-ang [16] proposed a hybrid bee colony algorithm for *FJSP* to minimize C_{max}. Hu et al. [17] solved flexible job shop scheduling problem by means of a hybrid harmony search (HHS) algorithm based on integrated approach.

2. 2. Literature Review of Lot Streaming Problem

Lot streaming or lot splitting is a concept in which a large production lot is split into smaller sublots so that its operations at successive stations can be overlapped and its progress accelerated.

According to this concept, the sublots are transported from one station to the next for processing without waiting for the entire production lot to be processed at the earlier station before being moved to the next stage. In other words, an operation and its preceding operation of the same product can be processed simultaneously, which creates an overlap between systems and can reduce the makespan and work in process. The literature of lot streaming problems can easily be categorized into two main streams, one stream dealing with the determination of the optimal sublot sizes for a single job (also called sublot sizing) and the other addressing the sublot sizing and job sequencing decisions simultaneously for the multiple job Case, given various job and shop characteristics. Cetinkaya and Duman [18] addressed the lot streaming problem of multiple jobs in a two-machine mixed shop to minimize C_{max}.

Liu [19] developed an effective heuristic method for discrete lot streaming with variable sub-lots. Sancar Edis and Ornek [20] proposed a heuristic method for stochastic flow. Pan et al. [21] proposed a discrete bee colony algorithm for the lot streaming problem in a flow shop environment to minimize total weighted earliness and tardiness penalties. Han et al. [22] have investigated on multi objective lot streaming flow shop scheduling problem and solve it with an improved NSGA-II algorithm. Han et al. [23] have considered a multi-objective lot streaming flow shop scheduling problem with interval processing time and have proposed a novel evolutionary algorithm to solve the problem. Lot streaming problem in a flow shop environment is studied by Fattahi et al. [24] and two hybrid metaheuristics based on variable neighborhood search are developed to solve the problem in order to reach C_{max}.

Our study improves the resulting solutions by hybridization of *MA* and *CPM* methods which has been proposed by Farughi et al. [15] in two modes; first we assume that operations can be overlapped but the lot streaming is not permitted and then lot streaming of jobs is allowed. In order to make sure the effectiveness of the proposed algorithms in this paper, 25 problems, which designed by Fattahi et al. [2], are employed and the given results are compared by Fattahi et al in terms of value of objective function and CPU time. Two types of overlapping in operation have been illustrated in Figure 1.

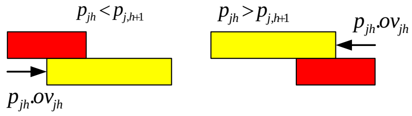

Figure 1. Types of overlapping operations

3. MATHEMATICAL FORMULATION OF FLEXIBLE JOB SHOP SCHEDULING PROBLEM

In this section, firstly some notations and definitions used throughout the paper are presented and following that the mixed integer linear programming formulation of FJSP will be introduced.

Sets and indices

m	*Set of machines*
n	*Set of jobs*
i	*Indices for machines*
j	*Indices for jobs*
h	*Sequence of operation for each job*

Parameters

ov_{jh}	*Overlapping coefficient between 0 and 1; 1 means no overlapping is allowed and 0 means complete overlapping permitted*
o_{jh}	*Operation h of job j*
ps_{jh}	*the processing time of operation of operation* o_{jh}
t_{jh}	*start time of the processing of operation* o_{jh}
k_i	*the number of operations assigned to machine i*
Tm_{ik}	*the start of working time for machine i in priority k*
P_{ijh}	*the processing time of operation h of job j on machine i*
L	*a large number*
P_{ijh}	*the processing time of operation h of job j on machine i*

Decision Variables

x_{ijkh}	*1 if machine i can perform* o_{jh} *in priority k and it will be 0 otherwise*
y_{ijh}	*1 if machine i is selected for operation* o_{jh} *and 0 otherwise*
a_{ijh}	*1 if operation* o_{jh} *is assigned to machine i and 0 otherwise*
r_{jh}	*1 if processing time of a predecessor operation is smaller than its successor and 0 otherwise*

A mixed integer linear programming (*MILP*) for flexible job shop scheduling with overlapping operations is developed as follows:

$$Min\ C_{max}$$

$$s.t.$$

$$C_{max} \geq t_{jh_j} + ps_{jh_j} \qquad \forall j \qquad (1)$$

$$\textstyle\sum_i y_{ijh}.P_{ijh} = ps_{jh} \qquad \forall j \qquad (2)$$

$$t_{jh} + ps_{jh}.ov_{jh} \leq t_{j,h+1} + (1 - r_{jh})L \qquad \forall j, h \qquad (3)$$

$$t_{jh} + ps_{jh}.ov_{jh} \leq t_{j,h+1} + ps_{j,h+1} - ps_{jh}.ov_{jh} + r_{jh}L \quad \forall j, h \qquad (4)$$

$$Tm_{ik} + ps_{jh}.x_{ijkh} \leq Tm_{i,k+1} \qquad \forall i, j, h, k \qquad (5)$$

$$Tm_{ik} \leq t_{jh} + (1 - x_{ijkh}).L \qquad \forall i, j, h, k \qquad (6)$$

$$Tm_{ik} + (1 - x_{ijkh}).L \geq t_{jh} \qquad \forall i, j, h, k \qquad (7)$$

$$y_{ijh} \leq a_{ijh} \qquad \forall i, j, h \qquad (8)$$

$$\textstyle\sum_j \sum_h x_{ijkh} = 1 \qquad \forall i, k \qquad (9)$$

$$\textstyle\sum_i y_{ijh} = 1 \qquad \forall j, h \qquad (10)$$

$$\textstyle\sum_k x_{ijkh} = y_{ijh} \qquad \forall i, j, h \qquad (11)$$

$$t_{jh} \geq 0 \qquad \forall j, h \qquad (12)$$

$$x_{ijkh} \in \{0,1\} \qquad \forall i, j, h, k \qquad (13)$$

$$y_{jkh} \in \{0,1\} \qquad \forall i, j, h \qquad (14)$$

The first constraint calculates the completion time and the second determines the processing time of operation o_{jh} on the selected machine. Constraints (3) and (4) enforce each job to follow a specified operation sequence and to consider the overlapping constraints. Constraint (5) assures that each machine processes only one operation at a time. Constraints (6) and (7) guarantee that each operation o_{jh} can be started after its assigned machine is idle and the previous operation o_{jh} 1 is completed. The flexibility of the problem is shown in constraint (8) which determines the alternative machines for each operation. Constraint (9) assigns the operations to a machine and sequence operations on all machines. As we know, in scheduling problems each operation can be performed only on one machine and at one priority at the same time; these conditions are shown in constraints (10) and (11). Here, x_{ijkh} leads to the assignment of each operation to a machine and sequencing the operations assigned on all machines. It should be noticed that the current study assumes that all machines are accessible from the beginning and preemption is possible throughout the operation at any time and for the time of necessity and with the existing idle machine or machines they may be utilized for this purpose. In the following, the process of solution of this problem will be described by means of memetic algorithm.

4. THE PROPOSED MEMETIC ALGORITHM

Memetic algorithms are a general framework which aims to provide the search with a desirable trade-off between intensification and diversification through the combined use of a crossover operator (to generate new promising solutions) and a local optimization procedure (to locally improve the generated solutions) [25].

In this section, structure of the proposed memetic algorithm which is intended to solve FJSP problem with assuming overlapping time of operations will be illustrated. The method of solving a problem in the proposed memetic algorithm is to take an integrated approach.

4. 1. Genetic Algorithm (GA) After definition of appropriate structure of gene and generation of population and computation of systemic fitness function, a group of best genes will be selected and then, next generation will be reproduced by means of crossover and mutation operators. In memetic algorithm, local search is also done to improve quality of the solutions before reproducing next generation.

The first generation can be produced by means two techniques:

1. *Creating random population;*

2. *Application of certain rules for reproduction;*

Leipins & Hiliard [26] express that although taking certain rules for reproducing initial population increase convergence speed, early convergence does not necessarily denote producing high quality solutions. For this reason, this type of generation of primary chromosomes does not result in many advantages. Thus, in this paper the initial population will be randomly generated. The used genetic structure has been shown in Figure 2. Each solution for the studied problem (each chromosome) is obtained by juxtaposing several genes. With respect to any solution, completion time of operations is calculated as fitness function. In order to select appropriate chromosomes, *Elite* method, which proposed by Yeh [26], has been used.

machine	*job*	*operation*	*priority*

Figure 2. The structure of the suggested gene executing memetic algorithm

In this method, first chromosomes are arranged in descending order and then the best chromosomes are selected for the next steps.

Crossover plays an important role in exchange data between chromosomes. The used crossover operator in this study is Precedence Operation Crossover (*POX*). In this operator, two offspring are generated by means of 2 chromosomes as parents. The logic of this operation is that one gene is randomly selected in first parent (which shows execution of a special operation by special machine). Afterwards, all correspondent genes to the selected gene transfer to first offspring while the rest genes on this chromosome are completed from genes in second parent chromosome. This process is done in a gene in second parent to generate next offspring. In Figure 3, the execution steps in *POX* operator are shown. In this figure, firstly gene no. 7, which has been defined and selected based on genetic structure, indicates second operation of task 2 by machine no. 2 in second precedence. Then, genes nos. 2 and 8, which display performing of operation of task 2 are selected as well and all of them are transferred to first offspring. In this offspring chromosome other genes are transcribed by second parent. These steps are done regarding the second offspring chromosome, but via genes in second and first parent chromosomes respectively. The noteworthy point regarding the given operator is that the reproduced new generations are feasible transcriptions from the parents. A feasible chromosome in this trend is a chromosome in which order of doing operation tasks is based on prerequisite operation relationship (i.e. first operation 1, then operation 2 and so forth from one job). The role of mutation operator is to prevent genes from embedding on optimal local points.

Similarly, application of several mutation operators may cause finding some solutions, which may not be often reproduced by crossover operator. In this study, in order to obtain a new child through a mutation operator, two genes are randomly selected from the parent's chromosome. In these genes, operation processor machines related to the specified job will be replaced with substitute machines which have the ability to process that specific operation; the new values of the transformed genes are copied in the corresponding places of child's chromosomes.

P1	1111	3211	2121	1412	5421	4131	2222	5232	2313	3322	4332	3433
P2	2411	1111	5421	3211	3222	4121	4132	5232	2313	3433	3324	5333
Ch1	2411	1111	5421	3211	2122	4131	2223	5232	2314	3432	3323	4332
Ch2	2411	3211	1111	5421	4121	4132	2222	5232	2313	3432	3323	5333

Figure 3. Creation of new generation by means of *POX* operator

The remaining empty cells of the child's chromosome is filled with corresponding genes on the parent's chromosome. It should be noted that in this case, as with the *POX* operator, only the number related to the order of processing of operations in each gene should be reviewed according to the values of previous genes and, if necessary, it should be changed. Figure 4 illustrates the mutation operator on the parent's chromosome (P) for producing child (CH). As observed, genes 1 and 7 have been randomly selected from chromosome P. Gene 1 (1111) belongs to processing of operation 1 from job 1. As observed in Table 1, in addition to machine 1, machines 2 and 3 are also able to perform operation 1 of job 1; accordingly, for example, machine 1 is replaced with machine 2 and the value of the gene changes from 1111 to 2111 and it is copied into the corresponding cell (first cell) of chromosome CH. Gene 7 of chromosome P (2222) is related to implementation of operation 2 of job 2; according to Table 1, in addition to machine 2, machines 3 and 4 are able to perform this operation. Therefore, the value of the gene has changed from 2222 to 3222 and has copied in the corresponding place in chromosome CH (seventh cell). Afterwards, to fill the remaining cells of chromosome CH, the corresponding genes on chromosome P are transferred into chromosome CH in their original order. For instance, gene 3211 is copied to the second cell of chromosome CH. Gene 2121 is copied in the third cell of chromosome CH, but as mentioned before, since machine 2 has performed its first activity in gene 1 of chromosome CH (2111), its second activity in gene 3 of this chromosome is written as 2122 instead of 2121. This process follows the same order until the other cells of chromosome CH are also completed.

4. 2. Local Search To execute local search, it necessitates firstly defining an appropriate neighborhood structure. Overall, neighborhood of a solution or chromosome is a solution that results from replacement for two or more different genes. In this paper, location of two genes is replaced with each other in order to use local search method. In other words, local search has been used by means of pair substitution. It is worthy of notice that it is possible during execution of local search for the studied chromosome to be converted into impossible chromosome. Therefore, possibility of chromosome will be always controlled after the end of local search.

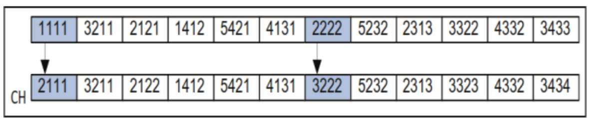

Figure 4. Mutation operator

5. LOT STREAMING

The first approach for *FJSP* problem was based on this assumption that an operation should be done by the same machine after allocation of any operation to one machine while such an assumption may lead to idleness of some machines within time intervals with respect to nature of the problem; alternately. In addition, if one can use these machines to release some part of operation for the next step, it is expected for us to witness an improvement in period of completion in these operations. Accordingly and in order to use the machines more than ever, any operation will be divided into two separate parts and then the current problem will be solved by taking this attitude. It is a matter of fact that this method will increase the number of constraints and variables and also complexity of the problem. In order to divide any operation into two parts, 5 different conditions of dividing operations, including 90-10, 80-20, 70-30, 60-40 and at last 50-50 are tested out and the best situation is selected for minimization of completion time. After dividing according to above percentages, the new problems are solved by means of described memetic algorithm in the previous sections. Figure 5 shows that in ***MFJS6***, to obtain the best value of completion time, lots must divide into two equivalent sublots.

6. COMPUTATIONAL RESULTS

In order to reduce the maximum completion time, Fattahi et al. [2] have added assumption for overlapping time of operations to their model and remarkably reduced maximum completion time. In this paper, the presented mathematical model in section 3 is solved under following considerations and the given results have been compared with the results derived from the survey done by Fattahi et al. [2]. All the results from this model indicate the improvement compared to the previous study.

1- Solving the problem by memetic algorithm with the assumption of overlap of operations;

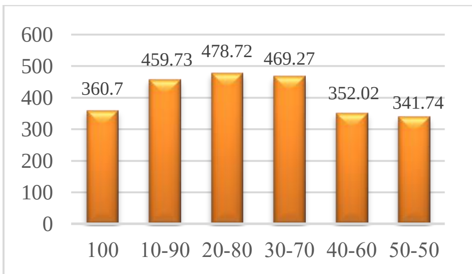

Figure 5. completion time for ***MFJS6***

TABLE 1. Computational results

Problem	SA[a-Ft]		MA[b]-with overlapping				CPM method				Percent of improvement
	OV		$W.LS$		LS		OV^F		$LS.Ov$		ef
	C_{max}	CPU time(s)	C_{max}	CPU time(s)	C_{max}	CPU time(s)	C_{max}	CPU time(s)	C_{max}	CPU time(s)	
SFJS1	66	2	66	3	64	1	66	3	66	2	0.00
SFJS2	107	6	107	3	100	1	94.2	4	85	2	20.56
SFJS3	221	6	221	3	209.8	2	219.7	4	175	3	20.81
SFJS4	355	6	355	5	321.6	3	346.3	5	303.6	5	14.48
SFJS5	119	10	119	7	109.2	4	108.2	6	92.1	5	22.61
SFJS6	256	10	256	7	256	6	236.36	8	236.36	8	0.00
SFJS7	233.5	12	233	8	207.18	7	218	8	187	11	19.91
SFJS8	193	20	193	14	217.8	11	193	13	154	12	20.21
SFJS9	171.7	20	171.7	14	196	13	177.7	15	131.4	14	23.47
SFJS10	419.5	26	423.85	15	417	13	419.5	15	376	18	10.37
MFJS1	441	20	466	18	441	16	440	19	421	21	4.54
MFJS2	315.5	20	316	20	301.3	20	315.5	21	301.3	22	4.12
MFJS3	419.5	25	419.5	22	357.56	22	419.5	22	419.5	24	0.00
MFJS4	376.8	30	374.6	26	340.5	23	343.3	28	310.6	28	17.57
MFJS5	313.3	35	313.3	27	309.8	26	286.5	30	243.1	29	22.41
MFJS6	**360.7**	**58**	**346.8**	**39**	**341.74**	**34**	**336.5**	**37**	**328.43**	**39**	**8.95**
MFJS7	402.2	118	402.2	56	387.4	48	331.2	62	324	67	19.44
MFJS8	430.3	190	430	73	419	61	414.4	80	396.4	90	7.88
MFJS9	511.7	210	511.7	81	499	73	491	93	478	102	6.59
MFJS10	504.5	284	517.3	108	504.5	96	470.6	110	453.4	113	10.13
LFJS1	589.9	384	542	134	532.8	118	500.2	141	465	149	21.17
LFJS2	862.3	640	791	217	745.1	189	730.1	215	702.7	221	18.51
LFJS3	824.6	840	763	263	720	237	702.6	275	679	297	17.66
LFJS4	864.4	920	797	389	808	352	756.1	406	732	440	15.32
LFJS5	1285	1860	1148	647	1015.45	591	1086	683	1047	712	18.52

2- Solving the problem with memetic algorithm by lot streaming assumption;

3- Improving the results derived from memetic algorithm by critical path method of Farughi et al. [15];

In order to test the algorithm and making sure of its efficiency, 25 examples used by Fattahi et al. [2] were solved and then compared with their reported results. To be sure of convergence in memetic algorithm, any problem was executed 10 times and the best results are given in Table 1. The aforesaid examples are called and divided into three classes of small, medium, and large-sized with quantities of 10, 10, and 5 respectively. For example, *SFJS1:2.2.2* is one of small- sized examples including 2 tasks and any task consists of 2 operations and 2 machines. Similarly, *MFJS4:5.3.6* for example is an instance in medium-sized class which includes 5 tasks, and any task consists of 3 operations with 7 machines. Overlapping rate has been assumed similar to overlapping rate in examples for Fattahi et al as 0.1 and a computer with Celeron processor (2GHz) and a memory with 512 MB capacities were utilized to solve 25 above examples. The obtained results may indicate that if lot streaming is not considered in jobs, memetic algorithm can find optimal solution for small and medium size problems by means of Fattahi et al. [2] method. Nevertheless, in large scale examples, this algorithm can find near-optimal solutions (and better solutions compared to the results of Fattahi et al method) with more appropriate speed; and by lot streaming consideration the solutions indicate a noticeable improvement. Likewise, the results derived from critical path method also confirm the improvement of solutions. To calculate the ef column of Table 1, the following equation is used:

$$ef = 100 \times \frac{|Pr.Ov - W.Ov|}{W.Ov}$$

By meticulous observation of the Table 1, it is seen that memetic algorithm has not created any improvement (optimization) for small- sized problems and some items from medium-sized examples without lot streaming and this is due to optimal solutions in these problems. The noticeable point in this table is the improvement of optimal solutions by means of creation of lot streaming approach while we know that to other results, the optimal solution is proposed to best performance in terms of minimum and or maximum value of target function or functions. This case is justifiable due to the changing nature of the problem in terms variables and constraints and increasing number of jobs. Expressing better, the suggested method has problem and it should be converted into a new problem. Likewise, critical path method heuristic approach has also improved the results and this can be interpreted due to changing in total structure of problem, and also because of adding possibility to employ parallel machines. A comparison of the makespan with lot streaming and CPM method with the previous algorithms is shown in Figure 6, this figure shows that proposed method of this paper can reduce the makespan measure significantly.

In the following, Gantt chart *MFJS6* problem which is shown in Table 2 is given in order to be sure of output of proposed methods and it is compared with Gantt chart for this example that proposed by Fattahi et al. [2]. In Figure 7, Fattahi et al. [2] Gantt chart is shown for this example. Figures 8-10 indicate respectively the given Gantt chart from memetic algorithm under the mode without preemption; the mode of creating preemption and optimization of memetic algorithm

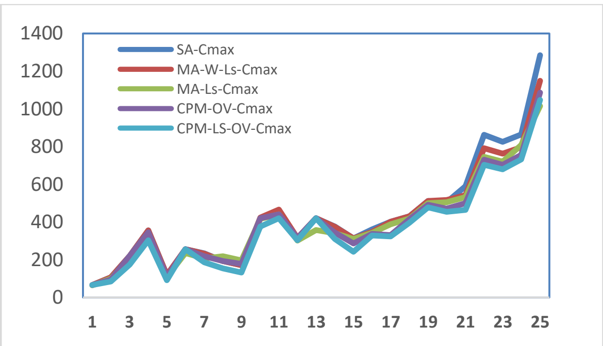

Figure 6. Difference between objective function of proposed method and SA of Fattahi et al.

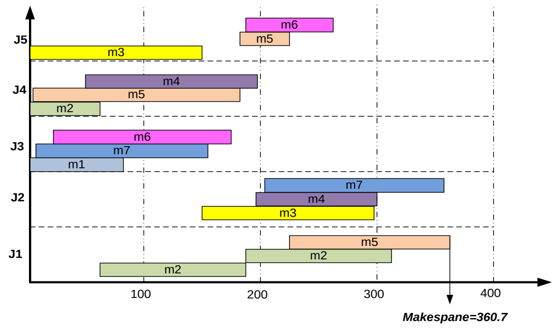

Figure 7. Gantt chart presented for MFJS6

results by critical path heuristic technique, and improvement of solution obtained by solving problem given condition of existing preemption by critical path method. It should be noted that in Figure 10, both successive jobs indicate two separate elements of the same job; for example, job 1 has been displayed within two separate operations under the titles of J_1 and J_2.

TABLE 2. Data for *MFJS6*

	Machine	Process Time	Machine	Process Time	Machine	Process Time
	M1	147	M4	140	M4	150
J_1	M2	123	M2	130	M5	141
	M3	145	M1	123	M7	200
	M1	214	M3	87	M5	178
J_2	M3	150	M2	66	M6	95
			M4	90	M7	150
	M1	87	M7	145	M4	190
J_3	M2	62	M3	180	M5	60
			M4	105	M6	153
J_4	M1	87	M3	250	M4	145
	M2	65	M5	173	M6	136
	M2	123	M3	86	M5	110
J_5	M3	145	M4	65	M6	85
	M1	128	M5	47		

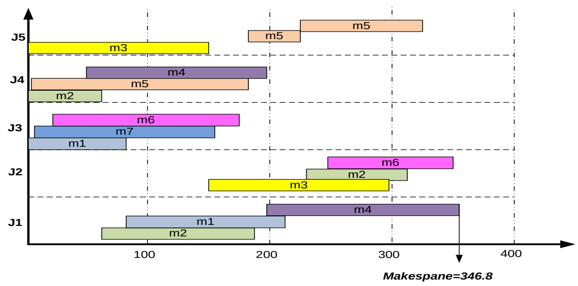

Figure 8. Gantt chart derived from MA without lot streaming

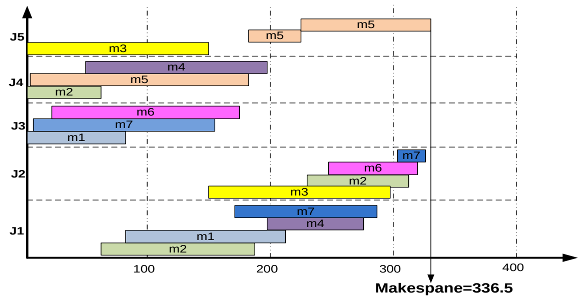

Figure 9. The improved Gantt chart of ***MFJS6*** without lot streaming by critical path method

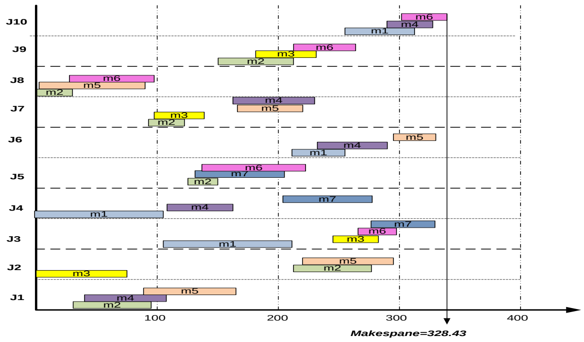

Figure 10. The improved Gantt chart of ***MFJS6*** with lot streaming and critical path technique

7. CONCLUSION

A new methodology was proposed to solve Flexible Job shop Scheduling Problem (FJSP) with assumed overlapping time of operations and by using memetic algorithm with two different approaches. In the first scenario, by assuming preemption of operations was not permissible, the problem was analyzed and then in the second scenario preemption of operations is considered. The obtained results from memetic algorithm for both scenarios were compared with the results which have been presented in Fattahi et al. [2]. This comparison indicates that if preemption is not allowed, memetic algorithm with small-size dimensions has achieved optimal solutions and concerning to medium and large scale problems, the results closing to optimal levels and the results better than those obtained by Fattahi et al. [2] were acquired. In the case of permissible preemption, the acquired results also indicate significant improvement in the generated solutions. Similarly, by adding idle and possible machines to any operation, heuristic method based on finding critical path and its optimization was also proposed in this paper in which the results derived from this method show improvement in objective function of the problem. It should be noticed that application of critical path method is based on the assumption that the necessary machines can be used all the times regardless of the necessary set-up time of machines.

8. ACKNOWLEDGEMENTS

The authors would like to thank the anonymous referees for their comments and suggestions that helped improve a previous version of this paper.
The authors are also grateful to Mr. Homayoun Shaabani for his assistance with the English language editing of the manuscript.

9. REFERENCES

1. Alvarez-Valdés, R., Fuertes, A., Tamarit, J.M., Gimenez, G. and Ramos, R., "A heuristic to schedule flexible job-shop in a glass factory", ***European Journal of Operational Research***, Vol. 165, No. 2, (2005), 525-534.
2. Fattahi, P., Jolai, F. and Arkat, J., "Flexible job shop scheduling with overlapping in operations", ***Applied Mathematical Modelling***, Vol. 33, No. 7, (2009), 3076-3087.
3. Brandimarte, P., "Routing and scheduling in a flexible job shop by tabu search", ***Annals of Operations Research***, Vol. 41, No. 3, (1993), 157-183.
4. Brucker, P. and Schlie, R., "Job-shop scheduling with multi-purpose machines", ***Computing***, Vol. 45, No. 4, (1990), 369-375.
5. Saidi-Mehrabad, M. and Fattahi, P., "Flexible job shop scheduling with tabu search algorithms", ***The International Journal of Advanced Manufacturing Technology***, Vol. 32, No. 5-6, (2007), 563-570.
6. Kacem, I., Hammadi, S. and Borne, P., "Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems", ***IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)***, Vol. 32, No. 1, (2002), 1-13.
7. Xia, W. and Wu, Z., "An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems", ***Computers & Industrial Engineering***, Vol. 48, No. 2, (2005), 409-425.
8. Zandieh, M., Mahdavi, I. and Bagheri, A., "Solving the flexible job-shop scheduling problem by a genetic algorithm", (2008).
9. Hurink, J., Jurisch, B. and Thole, M., "Tabu search for the job-shop scheduling problem with multi-purpose machines", ***Operations-Research-Spektrum***, Vol. 15, No. 4, (1994), 205-215.
10. Dauzere-Peres, S. and Paulli, J., "An integrated approach for modeling and solving the general multiprocessor job-shop scheduling problem using tabu search", ***Annals of Operations Research***, Vol. 70, (1997), 281-306.
11. Gambardella, L. and Mastrolilli, M., "Effective neighborhood functions for the flexible job shop problem", ***Journal of Scheduling***, Vol. 3, No. 3, (1996), 3-20.
12. Runge, N. and Sourd, F., "A new model for the preemptive earliness–tardiness scheduling problem", ***Computers & Operations Research***, Vol. 36, No. 7, (2009), 2242-2249.
13. Hendel, Y., Runge, N. and Sourd, F., "The one-machine just-in-time scheduling problem with preemption", ***Discrete Optimization***, Vol. 6, No. 1, (2009), 10-22.

14. Huo, Y., Leung, J.Y.-T. and Wang, X., "A fast preemptive scheduling algorithm with release times and inclusive processing set restrictions", ***Discrete Optimization***, Vol. 6, No. 3, (2009), 292-298.
15. Farughi, H., Yousefi Yegane, B. and Fathian, M., "A new critical path method and a memetic algorithm for flexible job shop scheduling with overlapping operations", ***Simulation***, Vol. 89, No. 3, (2013), 264-277.
16. Thammano, A. and Phu-ang, A., "A hybrid artificial bee colony algorithm with local search for flexible job-shop scheduling problem", ***Procedia Computer Science***, Vol. 20, (2013), 96-101.
17. Hu, Y., Yin, M. and Li, X., "A novel objective function for job-shop scheduling problem with fuzzy processing time and fuzzy due date using differential evolution algorithm", ***The International Journal of Advanced Manufacturing Technology***, Vol. 56, No. 9-12, (2011), 1125-1133.
18. Çetinkaya, F.C. and Duman, M., "Lot streaming in a two-machine mixed shop", ***The International Journal of Advanced Manufacturing Technology***, Vol. 49, No. 9-12, (2010), 1161-1173.
19. Liu, S., "A heuristic method for discrete lot streaming with variable sublots in a flow shop", ***The International Journal of Advanced Manufacturing Technology***, Vol. 22, No. 9-10, (2003), 662-668.
20. Edis, R.S. and Ornek, M.A., "A tabu search-based heuristic for single-product lot streaming problems in flow shops", ***The International Journal of Advanced Manufacturing Technology***, Vol. 43, No. 11-12, (2009), 1202-1210.
21. Pan, Q.-K., Tasgetiren, M.F., Suganthan, P.N. and Chua, T.J., "A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem", ***Information Sciences***, Vol. 181, No. 12, (2011), 2455-2468.
22. Han, Y.-Y., Gong, D.-w., Sun, X.-Y. and Pan, Q.-K., "An improved NSGA-II algorithm for multi-objective lot-streaming flow shop scheduling problem", ***International Journal of Production Research***, Vol. 52, No. 8, (2014), 2211-2231.
23. Han, Y., Gong, D., Jin, Y. and Pan, Q.-k., "Evolutionary multi-objective blocking lot-streaming flow shop scheduling with interval processing time", ***Applied Soft Computing***, Vol. 42, (2016), 229-245.
24. Fattahi, P., Azizi, V. and Jabbari, M., "Lot streaming in no-wait multi product flowshop considering sequence dependent setup times and position based learning factors", ***International Journal of Engineering Transactions A: Basics***, Vol. 28, (2015), 1031-1039.
25. Lai, X. and Hao, J.-K., "A tabu search based memetic algorithm for the max-mean dispersion problem", ***Computers & Operations Research***, Vol. 72, (2016), 118-127.
26. Yeh, W.-C., "A memetic algorithm for the n/2/flowshop/αf+βcmax scheduling problem", ***The International Journal of Advanced Manufacturing Technology***, Vol. 20, No. 6, (2002), 464-473.

Critical Path Method for Lot Streaming Problem in Flexible Job Shop Environment

B. Yousefi Yegane[a], I. Nakhai Kamalabadi[a], N. Khanlarzade[b]

[a] *Faculty of Engineering, University of Kurdistan, Sanandaj, Iran*
[b] *Faculty of Industrial Engineering, Tarbiat Modares University, Tehran, Iran*

P A P E R   I N F O

Paper history:
Received 19 August 2016
Received in revised form 25 January 2017
Accepted 25 January 2017

Keywords:
Memetic Algorithm
Flexible Job Shop Scheduling Problem
Lot Streaming
Critical Path Method

چکیده

در این تحقیق مساله زمان‌بندی کارِ کارگاهی منعطف با هدف کمینه‌سازی بیشینه زمان تکمیل مورد مطالعه قرار گرفته است. فرض اصلی در این تحقیق این است که هر کار شامل تعدادی انباشته است و لذا میتوان بجای انجام دادن یکباره هر کار توسط یک ماشین، کار مورد نظر را به دو یا چند کار مجزا تفکیک نمود و هر کار جدید را توسط ماشینهای مختلف پردازش نمود؛ به عبارت بهتر هر انباشته به زیر انباشته‌هایی تقسیم میشود که منجر به افزایش میزان بهره‌وری ماشین‌ها و کاهش زمان بیکاری آنها نیز خواهد شد. با توجه به پیچیدگی بالای مساله مورد بررسی، استفاده از رویکردهای دقیق نظیر الگوریتم شاخه و کران و نظایر آن، کارآیی خود را در حل مسایل با اندازه متوسط و بزرگ از دست می‌دهند، لذا به منظور دستیابی به جوابهای با کیفیت بالا در زمان مناسب استفاده از رویکردهای فراابتکاری انتخابی مناسب خواهد بود. در این تحقیق، ابتدا مساله مطرح شده، در دوحالت انجام فعالیت‌ها با در نظر گرفتن تفکیک هر کار به دو کار مستقل و بار دیگر بدون استفاده از این رویکرد، با استفاده از الگوریتم ممتیک حل شده و سپس جواب‌های بدست آمده با استفاده از یک رویکرد ابتکاری مبتنی بر مسیربحرانی بهبود داده می‌شوند. نتایج عددی بدست آمده در مقایسه با سایر روشهای بکار گرفته شده برای حل مساله، نشان دهنده کارآیی بالای الگوریتم ممتیک مورد استفاده است ضمن اینکه مجاز دانستن تفکیک کار به دو کار جدید و استفاده از ماشین‌های بیکار مجاز، بطور قابل ملاحظه‌ای سبب بهبود تابع هدف مساله شده است.

doi*: 10.5829/idosi.ije.2017.30.02b.13*