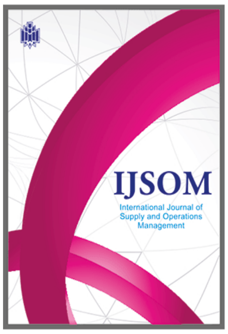
International Journal of Supply and Operations Management

IJSOM

November 2015, Volume 2, Issue 3, pp. 871-887
ISSN-Print: 2383-1359
ISSN-Online: 2383-2525
www.ijsom.com



A Flexible Job Shop Scheduling Problem with Controllable Processing Times to Optimize Total Cost of Delay and Processing

Hadi Mokhtari[*a] and Mehrdad Dadgar[b]

[a] Department of Industrial Engineering, Faculty of Engineering, University of Kashan, Kashan, Iran
[b] Department of Industrial Engineering, Faculty of Engineering, Tarbiat Modares University, Tehran, Iran

Abstract
In this paper, the flexible job shop scheduling problem with machine flexibility and controllable process times is studied. The main idea is that the processing times of operations may be controlled by consumptions of additional resources. The purpose of this study is to find the best trade-off between processing cost and delay cost in order to minimize the total costs. The proposed model, flexible job shop scheduling with controllable processing times (FJCPT), is formulated as an integer non-linear programming (INLP) model and then it is converted into an integer linear programming (ILP) model. Due to NP-hardness of FJCPT, conventional analytic optimization methods are not efficient. Hence, in order to solve the problem, a Scatter Search (SS), as an efficient metaheuristic method, is developed. To show the effectiveness of the proposed method, numerical experiments are conducted. The efficiency of the proposed algorithm is compared with that of a genetic algorithm (GA) available in the literature for solving FJSP problem. The results showed that the proposed SS provide better solutions than the existing GA.

Keywords: Flexible Job Shop Scheduling; Controllable Processing Time; Scatter Search; Disjunctive Graph.

[*] Corresponding author email address: mokhtari_ie@kashanu.ac.ir

1. Introduction

Today, due to diversity of customer demands, reduction of product life cycle, rapid development of processes and technologies, and subsequently rapid variation on the competitive markets, scheduling is of increasing importance. These types of commercial and economic pressures of market requires a system that despite maximizing the productivity of the resources is capable to meet high levels of the customer's consent to the products Therefore, such systems require correct, efficient and feasible schedules (Vokurka and Leary-Kelly, 2000). The flexible job shop scheduling problem is very important in two areas, production management and combinatorial optimization. However, due to the computational complexity, achieving the optimum solution to this problem is very difficult with conventional optimization methods. The combination of multiple optimization criteria will also increase the complexity of problem and create new problems (Xia and Wu, 2005). Garey et al., (1976) showed that the job shop scheduling problem is NP-hard. Since flexible Job shop scheduling problem is a generalization of classical job shop scheduling problem, this problem is at least as hard as classical job shop scheduling. Therefore, it belongs to NP-hard class (Tay and Ho, 2008). In 1990, Bruker and Chile (1990) provided a polynomial algorithm for solving the flexible job shop with two jobs. In 1993, Brandimarte (1997) was the first one who solved it using decomposition of the flexible job shop problem into two sub-problems: (1) routing, and (2) scheduling. The author presented an integrated approach based on a neighborhood structure where no difference between reallocation and re-sequencing of an operation exists.

After developing the meta-heuristic methods, a number of studies used these methods to solve their problems. In recent years, several meta-heuristic methods have been used for solving the flexible job shop problem, such as genetic algorithms (Zhang, et al, 2011), tabu search (Li, et al., 2010), variable neighborhood search (Bagheri and Zandieh, 2011), particle swarm optimization (Feng, et al., 2008), and bee algorithm (Wang, et al, 2011). This study is an attempt to solve the problem of flexible job shop with controllable processing times using a scatter search algorithm. For most deterministic scheduling problems, processing times are considered as constant parameters. However, in the real world, processing time can be changed by allocating resources such as money, time, energy, fuel, catalysts, contract, human resources, etc. In such systems job scheduling and resource allocation should be considered simultaneously such that system operates in its most efficient condition (Karabati, et al., 1995; Herroelen 1998). The scheduling problems with variable processing times are usually called as controllable processing times in literature. Most scheduling problems with controllable processing times are examined in continuous mode or single machine problems. For example, in the scheduling of the hot-rolled steel, pre-heat process depends on the amount of gas injected into the furnace. So, preheating time can be reduced by increasing fuel flow (Janiak, 1989). A number of studies have investigated the scheduling of the CNC machine (Akturk and Ilhan, 2011). Many studies have considered processing times as controllable in the single machine environment (Wang, 2006; Kayvanfar, et al., 2011). Moreover, we can find other researches which considered the case of controllable processing times (Mokhtari, et al., 2011a, Mokhtari et al., 2011b; Mokhtari, 2015). As it can be seen, few studies considered FJSS problem with machine flexibility and controllable processing times. Interesting readers can refer to more recently researches like (Giglio, 2015; Jiang, et al. 2015; Koca et al., 2015; Luo, 2015; Shioura et al., 2015).

2. Statement of the problem

Customer demand may be determined based on the delivery dates along with great penalties. Organizations are willing not to deviate from the delivery dates. One of these ways is the use of part time workers that organizations will be able to perform any job in a more intensive time by spending more costs and get the date of completion of the job closer to their due date. Such problems can be found in abundance in organizations such as production of websites, software and other service companies, consulting firms for designing building plans. Therefore, a balance must be established between the cost of delays and cost compression of operations. In these issues, it is decided which activity should be planned at normal time or in crash time.

2.1. Assumptions

Flexible job shop scheduling problem with flexible routing can be formulated as follows: a set of n independent job $J = \{1,2,\dots,n\}$ on m, $m \in \{M_1, M_2, \dots, M_m\}$ machines will be processed. The set of machines in the workshop is shown with m_i and the set of times of entering jobs in the workshop is shown with r_j which is assumed that $r_j = 0$. On the other hand, given the flexibility and capability of machines, each operation can be processed by a set of machines that is shown with M_{ij} . O_{ij} is the job operation on machine i for $i = 1,2,\dots,m$ and $j = 1,2,\dots,n$ working that interrupting the operation and cancelling them is not permitted. Alternatively, any operation can be processed in two compressed and normal mode.

2.2. Mathematical Model

Following the literature on scheduling, the processing times are usually fixed and definite, while the assumption in the real world is not very practical. With respect to the applicability of this issue, the activity time can be reduced by allocating additional resources. Therefore, it is required to use it in the manufacturing and service sectors and a mathematical model helps managers to make the right and optimal decision. Before formulating the problem, all of the parameters and variables used throughout the model are introduced.

I	Set of jobs
J	Set of operations
M_{ij}	Set of machines that are capable to process operation o_{ij}
C_{ijk}	Completion time of operation o_{ij} on machine k
C_i	Completion time of job i
L_i	Lateness of job i
d_i	Due date of job i

t_{ijk}^{N} Normal processing time of operation o_{ij} on machine k

t_{ijk}^{C} Compress processing time of operation o_{ij} on machine k

$cost_{ijk}^{N}$ Normal processing cost of operation o_{ij} on machine k

$cost_{ijk}^{C}$ Compress processing cost of operation o_{ij} on machine k

$cost_{i}$ Lateness cost of job i

A_{ijk} A binary variable that indicates weather machine k is capable to process operation o_{ij}

t_{ij} Start time of operation o_{ij}

Decision variables:

X_{ijk}^{C} A binary variable that indicates whether operation o_{ij} is processed at compress state on machine k

X_{ijk}^{N} A binary variable that indicates whether operation o_{ij} is processed at normal state on machine k

v_{ijk} A binary variable that indicates whether operation o_{ij} is processed on machine k

Z_{ijhgk} A binary variable that indicates order of operations o_{ij} and o_{hg} on machine k

p_{ij} Processing time of operation o_{ij} on machine k

s_{ijk} Starting time of operation o_{ij} on machine k

$$Min\ Z = \sum_{i=1}^{n}(L_i) * cost_i + \sum_i \sum_j \sum_k X_{ijk}^N * cost_{ijk}^N + X_{ijk}^C * cost_{ijk}^C \tag{1}$$

$$C_i \geq t_{ij} + p_{ij} \qquad i \in I \tag{2}$$

$$L_i \geq C_i - d_i \qquad i \in I \tag{3}$$

$$p_{ijk} = \sum_{k=1}^{m}(t_{ijk}^N * X_{ijk}^N + t_{ijk}^C * X_{ijk}^C) \quad i \in I \quad j \in J_i \tag{4}$$

$$X_{ijk}^N + X_{ijk}^C = v_{ijk} \qquad i \in I \quad j \in J_i \quad k \in K \tag{5}$$

$$t_{ij} + p_{ij} \leq t_{ij+1} \qquad i \in I \quad j \in J_i \tag{6}$$

$$\sum_{k=1}^{m} v_{ijk} = 1 \qquad i \in I \quad j \in J_i \tag{7}$$

$$Z_{ijhgk} + Z_{hgijk} = v_{ijk} * v_{hgk} \quad i \in I \quad j \in J_i \quad i \neq h \qquad h \in I \quad g \in J_i \quad k \in K \tag{8}$$

$$s_{ijk} + p_{ij} \geq s_{hgk} - (1 - Z_{ijhgk}) * M \quad i \in I \quad j \in J_i \qquad i \neq h \quad h \in I \quad g \in J_i \quad k \in K \tag{9}$$

$$s_{hgk} + p_{hg} \geq s_{ijk} - (1 - Z_{hgijk}) * M \quad i \in I \quad j \in J_i \qquad i \neq h \quad h \in I \quad g \in J_i \quad k \in K \tag{10}$$

$$s_{ijk} = t_{ij} \qquad i \in I \quad j \in J_i \tag{11}$$

$$v_{ijk} \leq A_{ijk} \qquad i \in I \quad j \in J_i \quad k \in K \tag{12}$$

$$L_i, s_{ijk}, c_{ijk}, C_i \geq 0 \qquad i \in I \quad j \in J_i \quad k \in K \tag{13}$$

$$Z_{ijhgk}, v_{ijk}, f_{ijhgk}, X_{ijk}^{C}, X_{ijk}^{N} \epsilon \{0,1\} \quad i \in I \quad j \in J_i \qquad i \neq h \quad h \in I \quad g \in J_i \quad k \in K \tag{14}$$

Equation (1) shows that the objective function is composed of total cost of delays and of processing. Equation (2) calculates the completion time of each job. Constraint number (3) specifies the amount of delay for each job. Constraint (4) states the processing time of o_{ij} on Kth machine. Constraint (5) shows that that if an operation is performed on a machine, it must be processed in a state of compressed or normal. Constraint (6) guarantees the sequence of different operations of a job. Constraint (7) implies that each operation is processed by a machine. Constraints (8 to 10) determine the sequence of operations that are performed on a machine. If two operations o_{ij} and o_{hg} want to be processed on Kth machine, so it is at the right side of constraint (7). Therefore, o_{ij} is processed before or after o_{hg}. Constraints (9 and 10) also show the onset of each of these two operations. Constraint (11) indicates the start of operations on each machine. Constraint (12) also implies the flexibility of each machine.
As it is shown in equation (8), the model has a nonlinear form. To convert non-linear to linear model in this study, the following transformations are presented. With the help of the following transformations INLP model will become ILP model.

$$Z_{ijhgk} + Z_{hgijk} = f_{ijhgk}$$

$$f_{ijhgk} \le v_{ijk} + v_{hgk} - 1$$

$$f_{ijhgk} \le v_{ijk}$$

$$f_{ijhgk} \le v_{hgk}$$

3. Scatter Search (SS) Algorithm

Flexible job shop problem that is a development on the classic job shop model is known as NP-hard problems. Due to the complexity of this type of problem, it requires much time to solve by exact algorithms. In such cases, meta-heuristic algorithms are usually employed. For example, in a study, Zhang et al., introduced the performance genetic algorithms that seek to minimize the time to complete the job. In the presented algorithm, two policies of local optimal and global selection were used to create high-quality solutions. Li et al., in 2011 have introduced tabu search algorithm for solving FJSP problem to minimize the job completion times. A set of four rules for the allocation of machines and scheduling operations are applied to produce high-quality solutions. Then the local search which is defined based on the neighborhood structure was used to produce better solutions (Li, et al., 2010). Bagheri and Zandieh considered the flexible job shop problem with setup times. They divided their problem into two sub-problems of allocation of machines and sequence of operations. Then meta-heuristics algorithm of neighborhood search was used to solve the problem (Bagheri and Zandieh, 2011),. In this study, job shop scheduling problem is developed by adding controllable time and given the complexity of flexible job shop problem, the meta-heuristic algorithm for scatter search was used to solve the mathematical model.
Scatter search algorithm is a heuristic technique which has been effectively applied for NP-hard optimization problems (Glover et al., 2000). The algorithm is used for problems with continuous and discrete variables in the case that the target functions is a single-objective or multiple objective. The success of the scatter search algorithm discussed in literature (Chinneck, 2004). SS was first used by Glover as a heuristic method for integer programming in 1967. Unlike genetic algorithm, the scatter search algorithm uses the method of producing varied solutions on a set of small solutions, and preserves solutions as a tabu search algorithm (Haq, et al., 2007). Another advantage of this method is that by producing various responses and searching around any point, avoids to produce the same results and can maintain the efficient solutions. SS flexible framework allows to use various improvements with numerous ways to get the effective solutions (Yin et al., 2010).

3.1. Scatter Search (SS) Stages

Before the introduction of the scatter search algorithm stages, the way of displaying solution in the algorithm should be specified. Each meta-heuristic method requires an approach to display the results. With regard to the structure of the problem, the problem can be defined in three sub-problems:

- Sub problem of sequence of operations
- Sub problem of allocation
- Sub problem of process mode

So the representation form of solutions should be so that shows all three sub problems, in this study, the following method is used for each sub-problem, a layer is designed and the number of elements of each layer equals to all operations of the job.

Operation sequencing	3	2	1	2	3	1	1	2
	O_{31}	O_{21}	O_{11}	O_{22}	O_{31}	O_{12}	O_{13}	O_{23}

Machine allocation	1	2	1	2	3	1	4	2
	M_1	M_2	M_1	M_2	M_3	M_1	M_4	M_2

Figure 1. Solution representation

Sequence layer is formed so that for number of operations of first job, number 1 is created in the layer and will be placed randomly in the array. In this case, first 1 appeared in the array represents O_{11}, the second number 1 represents O_{12} etc. This procedure is repeated for all jobs as the layer is completed.

Layer of operations allocation represents the machine allocated to the operation, so that the first element indicates the machine allocated to operation O_{11}. If the layer of operation processing is equal to 1, it means that it is processed at an intensive time; otherwise, by giving zero, operation is processed in the normal state with a longer time but with lower costs.

3.2. Diversification

The most effective methods of producing various solutions are those that are capable of producing a series of solutions to meet the balance between the diversity and quality (Blum and Roli, 2003). SS algorithm in the first step constitutes a population of versatile solutions that in this study, the following method was applied to determine the sequence of operations (Glover, 1998).

Suppose that an initial permutation is given T, the permutation is applied as initial core. $T = (1,2,\dots,n)$, its next sequence is defined as $T(h:s)$ *where s will be a positive number between* h *to* 1*, so the desired sequence will be as follows*:

$$T(h:s) = (s, s+h, s+2h, s+3h, \dots, s+rh) \tag{15}$$

Where r is the largest positive integer for which the relation $s + rh \le n$ is established. So permutation $T(h)$ for $h \le n$ will be as follows:

$$T(h) = (\ T(h:h), T(h:h-1), \dots, T(h:1)\ ) \tag{16}$$

This method is used as the basis for generating permutations, then the permutations will become the sequence of operations.
Vector T is used as the primary core for production of random permutation as follows:

$T = (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16)$

If $h = 5$ is selected, therefore we have:

$$T(5:5) = (5,10,15)$$

$$T(5:4) = (4,9,14)$$

$$T(5:3) = (3,8,13)$$

$$T(5:2) = (2,7,12)$$

$$T(5:1) = (1,6,11,15)$$

Therefore, random produced permutation $T(5)$ will be as follows:

$$T(5) = (5,10,15,4,9,14,3,8,13,2,7,12,1,6,11,15)$$

It is better that h chooses two close values to the square root of n, because they minimize the dependency between the primary core and new solution. So the preferred values for h will be between 1 to $n/2$.

3.3. Improvement

The purpose of this phase is the transition of the primary solution to an effective solution. Common practice in this section is local search by considering the stop condition rule, so that if improvement is not close to the answer, the search procedure will be stopped. The possibility of using other improvement methods that are able to escape from the local optimum as TS, SA and VNS exist for scatter search algorithm (Laguna, et al., 2003). In the proposed algorithm, a local search method is used for all three sub-layers in which two places are randomly selected in the sub-layer and their contents are replaced with each other.

3.4. Reference Set Update

Solution update in the reference collection during SS reps will be based on the quality of the solutions and the distance between them. Due to the structure of SS, it is essential to know the distance between two solutions. Measure of distance helps to understand how the solutions are apart from each other, and how they are distributed in the search space. Disjunctive graph is one of the most well-known concepts for modeling scheduling problems and issues related to sequencing machines. So this study has benefited from disjunctive graph, and the distance between the two solutions has been calculated using it (Barzegar and Motameni, 2011).

The flexible job shop scheduling problem can be modeled as disjunctive graph $G = (V, C \cup D)$, in which:

V: is a set of nodes that shows any job operations. Furthermore, it includes two special nodes of beginning and ending that shows the beginning of scheduling and ending it respectively.

C: is a set of arcs that shows the technological limitations of operation sequences.

D: is a pair of arcs which indicates pair operations that must be done on a machine.

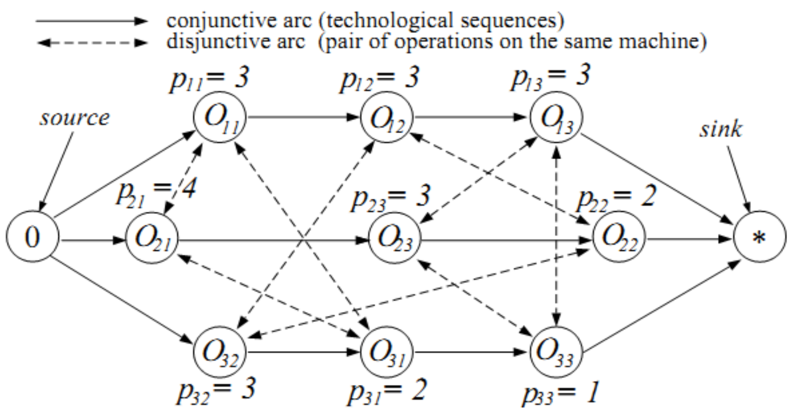


Figure 2. An example of disjunctive graph (Shirinivas et al., 2010)

Generally, the reference set consists of a limited number of members of primary population. The set includes $b_1 = 10$ *of high quality solutions* and $b_2 = 10$ *solution*s with the highest distance standards between the solutions.

3.5. Subset Generation

As mentioned previously, the distance criterion calculates the distance between the two specified solutions. The solutions in this question are the scheduling programs. As an example let us to consider two scheduling programs a and b in Figure 3 that are two hypothetical solutions. According to the disjunctive graph, this distance between the two scheduling programs includes different orders of operations that are performed on each machine (Shirinivas et al., 2010).
In other words, the distance is equal to the sum of disjoint arcs which directions are different in scheduling program a and b. Figure 3 shows the distance between the two scheduling programs. Two disjoint arcs that are specified in the Figure b are different from a. The distance between the disjunctive graph between scheduling solutions a and b is equal to 2.

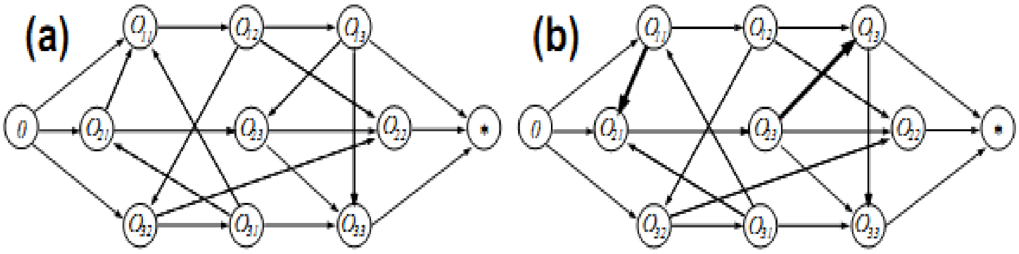


Figure 3. Distance between the two scheduling solutions (Karabati, et al., 1995)

After defining the distance between the two solutions, it is required to generate a reference set. As mentioned in the previous section, reference set consists *of* $b_1 + b_2$ as following: first b_1 the best solutions obtained is chosen and added to the reference set, at the stage, b_2 members of the solutions generated have the most distance with the existing solutions and will be added to the reference set. Thus, the set will have a balance of high quality, scattered and diverse solutions. In this stage, inputs are the reference set solutions. The usual method for the stage produces all possible solutions from the solution set. In SS, several methods rather than one can be used to generate solutions. However generally, the methods are limited to the production of solutions from two members of the reference set. So, the two solution subsets are selected from reference set and are combined with each other.

3.6. Solution Combination

After choosing binary series, the combination operator will be performed on it:

- The sequence sub-problem: things are placed in two categories: $J_1$and J_2. The jobs placed in J_1 are transferred to first child and the rest empty spaces of the second parent will be placed in the first child.
- The allocation sub-problem: In this operator, two points are randomly selected and in the layer of machine allocation, the location between these two points is transferred to children and the end parts are taken from the relevant parent.
- The processing sub-problem: In this type of operator, two points are randomly selected in processing mode layer. The elements between these two points are transferred to the opposite child and the rest of the child elements are taken from the parent.

So for each of the above sub-layers, combination operator is used and two parents generate two children. The children will be replaced in the population, if the objective function value is not worse.

3.7. Reference Set Update

After generating sets, the reference set is updated as follows: first, a local search is implemented on the solution obtained by the method. If we have not gained any improvement, the solution is compared with the worst reference solution with reference qualityb_1. If its objective function value was better than the quality of the worst results will be added to the set and takes. If the generated solution was not better than the high quality set. In the second step, the algorithm checks if the distance of generated solution from the high quality set exceeds the permitted limits of the second set. Then the solution will be replaced in the second set.

3.8. Stop Criteria

Each meta-heuristic algorithm requires a stop condition to do a search in the solution space. The stop criterion may be defined based on convergence of solutions in search space. In a manner that if the distance between the objective function of two solutions is always less than a number, the algorithm is stopped or may be defined based on the number of iterations of the algorithm. In this algorithm, stopping criterion is defined based on the number of iteration of algorithm and is considered as100.

4. Numerical Results

According to the authors' knowledge, so far flexible job shop problem has not been studied with the controllable times in the literature. However, due to the lack of a similar problem in the literature and this point that so far a meta-heuristic method has not been used to solve this kind of problems, the authors have tried to use genetic algorithm to evaluate and assess the scatter search algorithm which is used by Zhang et al to solve the flexible job shop problem (Zhang et al., 2011). To generate the delay time of sample problems from the introduced relationships in reference (Zribi et al., 2006) [32] have been used so that:

$$t_{ijk}^{N} \sim U[a,b] \qquad a > 1 \tag{17}$$

$$x \sim U[1, t_{ijk}^{N} - 1] \tag{18}$$

$$t_{ijk}^{C} = t_{ijk}^{N} - x \tag{19}$$

$$d_i = r_i + \beta \sum_{j=1}^{J_i} \bar{P}_{ij} \tag{20}$$

$$\beta = \frac{N * M}{1000} + 0.5 \tag{21}$$

Where $\bar{P}_{ij}$ is the average processing time of operation and O_{ij} and M are the number of machines and N is the number of jobs. Table 1 represents the results of comparing two scatter search algorithm and genetic algorithm. Each problem is solved in 10 repetitions with each algorithm. To compare two algorithms, they used two factors of mean solution time and the mean value of the objective function. The results of the study suggest that for numerical problem to be solved, the objective function value obtained by the scatter search algorithm is better.

Table 1. Comparison of SS and GA results

Problem	Jobs	Machines	Operations	GA		SS algorithm	
				Total Cost	Run time (s)	Total Cost	Run time (s)
TP01	4	4	4	2400	93	2400	79
TP02	6	6	5	6850	110	6850	83
TP03	10	10	6	14150	132	12350	96
TP04	12	12	8	15850	154	15900	113
TP05	15	14	10	29950	198	23000	149
TP06	18	18	15	65807	256	6280	202
TP07	20	20	20	158964	369	127975	289
TP08	28	28	28	196045	752	170045	579
TP09	35	35	35	465074	995	353312	879
TP10	50	50	50	900234	1670	770346	1378

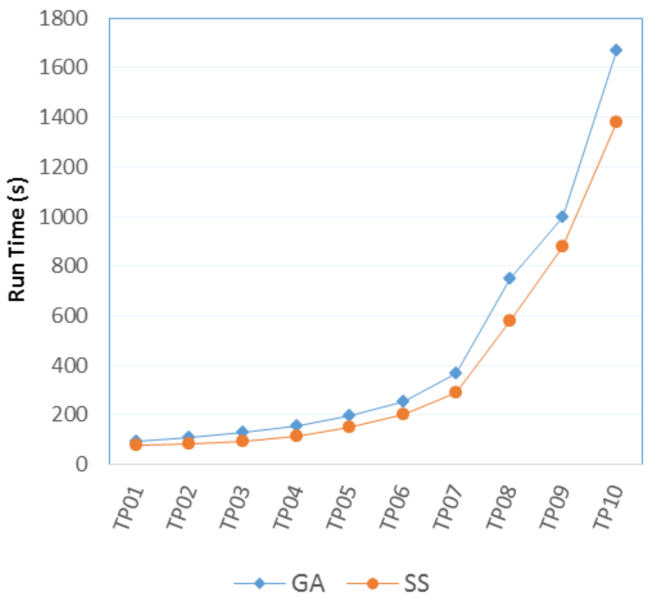


Figure 4 . Run time of algorithms

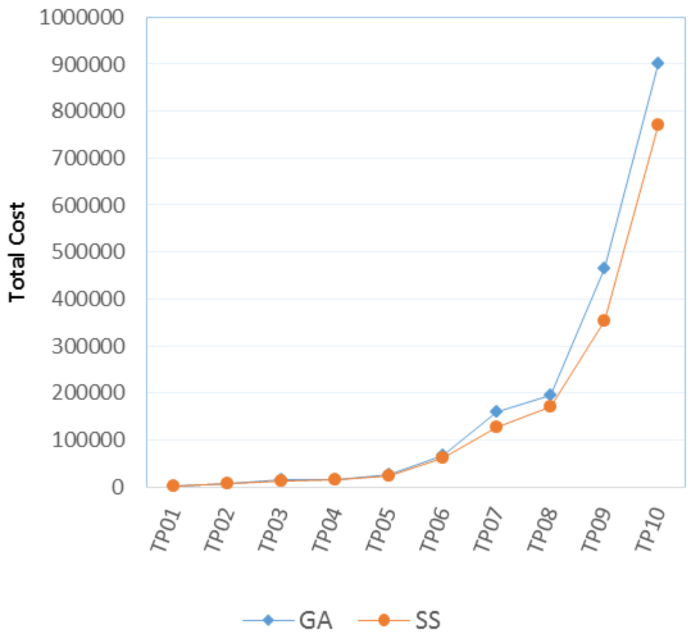


Figure 5. Accuracy of algorithms

As displayed in Figures 4 and 5 on the performance of the algorithm, the issues can be pointed out that genetic algorithm for sample problems dedicated more solution time. For values of the objective function obtained for algorithms, the scatter search algorithm reduced on average 10% of the value of the objective function and provided a better solution.

5. Conclusion

In the real world, the operation times are not fixed and with the allocation of resources and spending more costs, it is then possible to control the allocated time. In this study, for the first time, ILP model is presented for the flexible job shop problem with controllable time. A model that strikes a balance between the cost of delays and cost of compression of operations helps the organization to be committed to the delivery date of services / products at lower costs and customer satisfaction. At the end, in the present research, scatter search algorithm is developed to solve this problem.
ILP model in this paper was presented for the first time for flexible job shop problem, while the operations time is controlled. Always a delay in the delivery of services causes costs to firms. Therefore, companies seek to reduce costs by adopting the proposed approach, as suggested in this study, which introduces by controllable times and meta-heuristic scatter search algorithm for obtaining efficient solutions are presented. After reviewing the results of the algorithm, the scatter search algorithm on average showed a 10% improvement in the objective function than the genetic algorithm in the literature.
For future research, with regard to the acceptance and rejection rate of jobs, the organizations can be assisted by not to accept orders that do not profit them. Moreover, as another point, the problem can be combined with the problem of finding the optimal number of resources.

References

Akturk, M.S. and T. Ilhan, (2011) *Single CNC machine scheduling with controllable processing times to minimize total weighted tardiness.* Computers & Operations Research, 38(4): p. 771-781.

Bagheri, A. and M. Zandieh, (2011) *Bi-criteria flexible job-shop scheduling with sequence-dependent setup times—Variable neighborhood search approach".* Journal of Manufacturing Systems, 2011. 30(1): p. 8-15.

Barzegar, B. and H. Motameni, (2011) *Optimality of the flexible job shop scheduling system based on Gravitational Search Algorithm".* JOURNAL OF ADVANCES IN COMPUTER RESEARCH .

Blum, C. and A. Roli, (2003) *Metaheuristics in combinatorial optimization: Overview and conceptual comparison.* ACM Computing Surveys (CSUR), 35(3): p. 268-308.

Brucker, P. and R. Schlie, (1990) *Job-shop scheduling with multi-purpose machines".* Computing, 45(4): p. 369-375.

Chinneck, J.W., (2004) *Practical optimization: a gentle introduction".* Electronic document.

Dauzère-Pérès, S. and J. Paulli, (1997) *An integrated approach for modeling and solving the general multiprocessor job-shop scheduling problem using tabu search".* Annals of Operations Research, 70: p. 281-306.

Feng, M., et al., (2008) *A Grouping Particle Swarm Optimization Algorithm for Flexible Job Shop Scheduling Problem,* 332-336.

Garey, M.R., D.S. Johnson, and R. Sethi, (1976) *The complexity of flowshop and job-shop scheduling".* Mathematics of Operations Research, 1, 117-129.

Glover, F., M. Laguna, and R. Martí, (2000) *Fundamentals of scatter search and path relinking".* Control and cybernetics, 39(3): p. 653-684.

Glover, F. (1998) *A template for scatter search and path relinking*. in *Artificial evolution*. Springer.

Giglio, D. (2015) Optimal control strategies for single-machine family scheduling with sequence-dependent batch setup and controllable processing times, Journal of Scheduling, 18(5), 525-543.

Haq, A.N., et al., (2007) *A scatter search approach for general flowshop scheduling problem* .The International Journal of Advanced Manufacturing Technology, 31(7-8): p. 731-736.

Herroelen, W., B. De Reyck, and E. Demeulemeester, (1998) *Resource-constrained project scheduling: a survey of recent developments.* Computers & Operations Research, 25(4) 279-302.

Janiak, A., (1989) *Minimization of resource consumption under a given deadline in the two-processor flow-shop scheduling problem.* Information Processing Letters, 32(3): p. 101-112.

Jiang, S., Liu, M., Hao, J., Qian, W. (2015) A bi-layer optimization approach for a hybrid flow shop scheduling problem involving controllable processing times in the steelmaking industry, Computers and Industrial Engineering, 87, 518-531.

Karabati, S., P. Kouvelis, and G. Yu, (1995) *The discrete resource allocation problem in flow lines.* Management Science. 41(9): p. 1417-1430.

Kayvanfar, V., I. Mahdavi, and G.M. Komaki, (2011) *Single machine scheduling with controllable processing times to minimize total tardiness and earliness.* Computers & Industrial Engineering.

Koca, E., Yaman, H., Aktürk, M.S. (2015) Stochastic lot sizing problem with controllable processing times, Omega, 53, 1-10.

Luo, C. (2015) Single machine batch scheduling problem to minimize makespan with controllable setup and jobs processing times, Numerical Algebra, Control and Optimization, 5(1), 71-77.

Laguna, M., R. Martín, and R.C. Martí, (2003) *Scatter search: methodology and implementations in C*. Vol. 24.: Springer.

Li, J.-Q., et al., (2010) *A hybrid tabu search algorithm with an efficient neighborhood structure for the flexible job shop scheduling problem".* The International Journal of Advanced Manufacturing Technology, 52(5-8): p. 683-697.

Mokhtari, H., I.N.K. Abadi and A. Cheraghalikhani, (2011a) *A multi-objective flow shop scheduling with resource-dependent processing times: Tradeoff between makes pan and cost of* *resources*. Int.J. Prod. Res., 49, 5851-5875.

Mokhtari, H., Nakhai Kamal Abadi, I., Zegordi, S.H., (2011b) *Production Capacity Planning and Scheduling in a No-Wait Environment with Controllable Processing Times: An integrated modeling approach"*, Expert Systems with Applications, 38, 12630-12642.

Mokhtari, H. (2015) *Designing an efficient bi-criteria iterated greedy heuristic for simultaneous order scheduling and resource allocation: a balance between cost and lateness measures"*, Neural Computing and Applications, 26, 1085-1101.

Shioura, A., Shakhlevich, N.V., Strusevich, V.A. (2015) Optimal control strategies for single-machine family scheduling with sequence-dependent batch setup and controllable processing times, Mathematical Programming, 153(2), 495-534.

Shirinivas, S.G., S. Vetrivel, and D. N.M.Elango, (2010) *Applications of graph theory in computer science an overview".* International Journal of Engineering Science and Technology, Vol. 2(9), 2010, 4610-4621

Tay, J.C. and N.B. Ho (2008) *Evolving dispatching rules using genetic programming for solving multi-objective flexible job-shop problems".* Computers & Industrial Engineering, 54(3): p. 453-473.

Vokurka, R.J. and S.W.O. Leary-Kelly, (2000) *A review of empirical research on manufacturing flexibility.* Journal of Operations Management 2000.

Wang, L., et al ,. (2011) *An effective artificial bee colony algorithm for the flexible job-shop scheduling problem".* The International Journal of Advanced Manufacturing Technology, 60(1-4): p. 303-315.

Wang, J.-B., (2006) *Single machine scheduling with common due date and controllable processing times".* Applied Mathematics and Computation, 174(2): p. 1245-1254.

Xia, W. and Z. Wu, (2005) *An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems.* Computers & Industrial Engineering, 2005. 48(2): p. 409-425.

Yin, P.-Y., et al., (2010) *Cyber swarm algorithms–improving particle swarm optimization using adaptive memory strategies.* European Journal of Operational Research, 201, 377-389.

Zhang, G., L. Gao, and Y. Shi, (2011) *An effective genetic algorithm for the flexible job-shop scheduling problem.* Expert Systems with Applications, 2011. 38(4): 3563-3573.

Zribi, N., et al., (2006) *Minimizing the total tardiness in a flexible job-shop.* World Automation Congress.