RESEARCH NOTE

A MATHEMATICAL MODEL OF A MULTI-CRITERIA PARALLEL MACHINE SCHEDULING PROBLEM: A GENETIC ALGORITHM

R. Tavakkoli-Moghaddam*, F. Jolai, Y. Khodadadeghan and M. Haghnevis

Department of Industrial Engineering, Faculty of Engineering
University of Tehran, P.O. Box 11365/4563, Tehran, Iran
tavakoli@ut.ac.ir - fjolai@ut.ac.ir - ykhodadad@engmail.ut.ac.ir - haghnevis@engmail.ut.ac.ir

*Corresponding Author

(Received: November 16, 2005 – Accepted in Revised Form: June 1, 2006)

Abstract This paper presents a new mathematical model for a multi-criteria parallel machine scheduling problem minimizing the total earliness and tardiness penalties as well as machine costs. Machines are defined as unrelated parallel machines, so they have different speeds. To solve such a NP-hard problem, a meta-heuristic method based on genetic algorithms is proposed and developed. New operators are defined and applied in order to improve the quality of solutions. A number of test problems are carried out and the associated computational results are represented. The results show that the proposed algorithm is effective**.**

Key Words Multi-criteria parallel machine scheduling, Earliness and tardiness penalties, Machine costs, Genetic algorithms

چکیده این مقاله یک مدل برنامه‌ریزی ریاضی را برای مساله زمانبندی ماشین‌های موازی چند معیاره به منظور حداقل کردن کل جریمه‌های زودکرد و دیرکرد و نیز هزینه‌های ماشین ارائه می‌دهد. ماشین‌ها به صورت ماشین‌های موازی نامرتبط تعریف می‌شوند که در نتیجه سرعت‌های مختلفی خواهند داشت. برای حل چنین مساله چند جمله‌ای سخت، یک الگوریتم فرا ابتکاری بر اساس الگوریتم ژنتیک پیشنهاد شده است. عملگرهای جدیدی برای آن تعریف شده است تا کیفیت حل بهبود یابد. تعدادی مساله حل شده و نتایج محاسباتی آنها ارائه شده است. این نتایج نشان می‌دهند که الگوریتم ژنتیک پیشنهادی به‌طور موثری عمل می‌کند.

1. INTRODUCTION

There is a need to consider multi criteria in a wide range of scheduling problems [1]. Parallel scheduling considering multi criteria has been receiving an increasing amount of attention in recent years. In the current trends towards just-in-time (JIT) manufacturing strategies, the completion of both early and tardy jobs is undesired. The objectives related to the associated earliness and tardiness penalties have become increasingly popular [2]. In the current business environment, the competitiveness of manufacturing companies is being determined by their ability to quickly respond to the rapidly changing commercial areas and to produce high quality products at lower costs. Manufacturing companies are striving to achieve these capabilities through automation and innovative concepts, e.g., just-in-time (JIT), quick response (QR), group technology (GT), and total quality management (TQM) [3]. Due to the increased pressure towards JIT policies, many manufacturing firms are faced with the need to develop schedules that complete each customers order by the desired deadline i.e., both earliness and tardiness in job completion times are important. Clearly, in a JIT environment, there are penalties associated with both. For instance, penalties in regard to early deliveries, may refer to holding costs, deterioration costs in the case of perishable goods, and the like, if customers do not pick up a delivery until the due date. Likewise, tardiness can refer to discounts that may be offered to compensate customers for, a late delivery. These penalties can be different for different jobs based on the value and/or priority of jobs/customers [4]. We face a problem of sequencing and scheduling all the jobs to be processed on time in which machines are in parallel with.

Most researches suppose that parallel machines are identical, so their speeds and other factors are the same, but in the real environments there are old and new machines with different factors. Furthermore in each environment and industry, there are some costs such as maintenance costs, running costs (electricity and water usage), and the like for each machine. It is important to know which machines should be selected with the lowest costs.

The majority of scheduling researches assume that setup times are negligible or part of the processing time. While, this assumption simplifies the analysis and/or reflects certain applications; it adversely affects the solution quality for many applications which require the explicit treatment of setup [5]. By knowing that the scheduling of independent jobs with a common due date on a single machine has already been an NP-hard problem [2]. There fore, a parallel machine scheduling problem is an NP-hard one, even for the least complex single objective problem [6]. When a job tardiness penalty weights are arbitrary positive numbers, the problem is NP-hard in a strong sense and it is NP-hard in an ordinary sense when all weights are equal [6]. Many research efforts have concentrated on a number of heuristic approaches. Thus, we use a meta-heuristic method based on genetic algorithms to schedule unrelated parallel machines with sequence dependent setup times.

Cao et al. [7] considered a parallel machine selection and job scheduling to minimize the machine cost and job tardiness. They developed a heuristic algorithm to find the optimal or near optimal solutions based on a tabu search method. They also assumed that setup times are set to zero. Bilge et al. [8] considered a scheduling problem for a set of independent jobs with sequence dependent setup times done on a set of uniform parallel machines in such a way that total tardiness is minimized (assuming that all jobs have non-identical due dates and arrival times).

A tabu search (TS) approach is employed to tackle this complex problem. In order to obtain a robust search mechanism, several key components of TS such as candidate list strategies, tabu classifications, tabu tenure, and intensification /diversification strategies are investigated. In this study, in addition to distinct due dates and preparation times, features such as sequence dependent setup times and different processing rates for machines are incorporated into the classical model.

Liaw et al. [9] considered a scheduling problem for a given set of independent jobs on unrelated parallel machines to minimize the total weighted tardiness. As this problem is difficult to solve, a branch-and-bound algorithm incorporated with various dominance rules along with efficient lower and upper bounds is proposed to find an optimal solution. Min and Cheng [10] and Cochran et al. [6] presented a kind of genetic algorithms based on a sectional code to minimize the total cost of assignment of due date, earliness, and tardiness in a scheduling problem in order to find the optimal common due dates and the optimal scheduling policy by determining the job number and their processing order on each machine. Also, a simulated annealing method and the iterative heuristic fine-tuning operator are introduced into the genetic algorithm so as to construct three kinds of hybrid genetic algorithms with good performance.

Koulamas and Kyparisis [11] considered a uniform parallel machine scheduling problem to minimize the maximum lateness. They showed that an extension of the EDD rule to a uniform parallel machine setting yields the maximum lateness value which does not exceed the optimal value by more than the maximum job processing time. Azizoglu and Kirca [12] considered the NP-hard problem of scheduling jobs on identical parallel machines to minimize the total tardiness. They proposed a branch-and-bound algorithm that incorporates the properties along with an efficient lower bounding scheme.

In this paper, we consider a meta-heuristic algorithm to minimize the sum of earliness and tardiness penalties as well as machine costs in parallel machine scheduling. Sequence dependent setup times are also considered and parallel machines are unrelated. The rest of the paper is organized as follows: In Section 2, we propose a new mathematical model. Section 3 describes the suggested genetic algorithms. Section 4 shows the computational results. Finally, the conclusion is discussed.

2. MATHEMATICAL MODELING

In this paper, we consider N jobs on a number of unrelated parallel machines selected from a set of M potential machines so as to minimize the sum of earliness and tardiness penalties as well as machine costs. If a machine is selected to process any of the jobs (at least one job), a machine cost will be incurred in which they are independent. Different machines may operate at different speeds where different jobs have different earliness and tardiness penalties. Each machine processes one job at a certain time and each job should be completed on one machine. It means that preemption is not allowed. Each job has its own distinct due date that is fixed by the customer. Sequence dependent setup times are considered and triangular law of inequality, i.e., $s_{ijm} + s_{jkm} \geq s_{ikm}$ for all jobs $i, j,$ and k on machine m is satisfied.

We propose an integrated mixed-integer model for this problem. First, we give notations and variables:

i, j job indicates where job 0 is a dummy job which is always at the first position on a machine $(i, j = 0, 1, ..., N)$

e_i earliness penalty of job i

t_i tardiness penalty of job i

β_k machine cost ($k = 1, \ldots, K$)

k machine index ($k = 1, \ldots, K$)

C_i completion time of job i

E_i earliness of job i

T_i tardiness of job i

d_i due date of job i

L large positive number

P_{ik} process time of job i on machine k $(i = 1, \ldots N; k = 1, \ldots, K)$

S_{ijk} setup time to switch from job i to job j on machine k

$$x_{ijk} = \begin{cases} 1 & \text{if job } j \text{ immediately follows job } i \text{ on machine } k \\ 0 & \text{otherwise} \end{cases}$$

$$y_{jk} = \begin{cases} 1 & \text{if job } j \text{ is assigned to machine } k \\ 0 & \text{otherwise} \end{cases}$$

$$z_k = \begin{cases} 1 & \text{if machine } k \text{ is selected} \\ 0 & \text{otherwise} \end{cases}$$

Applying the above notations and variables, the scheduling model is formulated as follows:

$$Min \sum_{k=1}^{K} \beta_k z_k + \sum_{i=1}^{N} (e_i E_i + t_i T_i) \quad (1)$$

s.t.

$$C_i + E_i - T_i = d_i \qquad i = 1, ..., N \quad (2)$$

$$\sum_{\substack{i=0 \\ i \neq j}}^{N} \sum_{k=1}^{K} x_{ijk} = 1 \qquad j = 1, ..., N \quad (3)$$

$$\sum_{\substack{i=0 \\ i \neq j}}^{N} x_{ijk} = y_{jk} \qquad j = 1, ..., N \ ; \ k = 1, ..., K \quad (4)$$

$$\sum_{\substack{j=1 \\ j \neq i}}^{N} x_{ijk} \leq y_{ik} \qquad j = 1, ..., N \ ; \ k = 1, ..., K \quad (5)$$

$$\sum_{k=1}^{K} y_{ik} = 1 \qquad i = 1, ..., N \quad (6)$$

$$\sum_{j=1}^{N} x_{0jk} \leq z_k \qquad k = 1, ..., K \quad (7)$$

$$\sum_{j=1}^{N} x_{0jk} > z_k - 1 \qquad k = 1, \ldots, K \qquad (8)$$

$$C_j - C_i + L(1-x_{ijk}) \geq P_{jk} + S_{ijk} \qquad i = 1, \ldots, N ; j = 1, \ldots, N, \; i \neq j \qquad (9)$$

$$C_i \geq \sum_{k=1}^{K} (S_{0ik} + P_{ik}) \times y_{ik} \qquad i=1,\ldots, N ; k=1, \ldots, K \qquad (10)$$

$$Y_{ik}, X_{ijk}, z_k \in \{0, 1\} \qquad i = 1, \ldots, N ; j = 1, \ldots, N; k = 1, \ldots, K ; i \neq j \qquad (11)$$

$$C_i, E_i, T_i \geq 0 \qquad i = 1, \ldots, N \qquad (12)$$

In this proposed model, Equation 1 is the objective function consisting of three terms. These terms are the machine cost, the total weighted earliness penalties, and the total weighted tardiness penalties, respectively. Equation 2 calculates the earliness or tardiness of job i based on its completion time. Equation 3 ensures that a job must be processed at one and only one position on a machine. Equation 4 illustrates that if job j is assigned to Machine *k*, it should come after one of the jobs including job 0. Equation 5 states that if job j is assigned to Machine *k*, at least one job will immediately follow. Equation 6 ensures that each job assigned to one machine in which preemption is not allowed. Equation 7 ensures that if a machine is not selected, no job should go on it. Equation 8 guarantees that if a machine is selected, at least one job should be assigned to it. Equations 9 and 10 establish the relationship between the completion times of jobs *i* and *j* according to setup times as long as both jobs are assigned to the same machine. Equations 11 and 12 define zero-one variables and nonnegative variables, respectively. By applying data given in [4], we solve the proposed model in a small size by the Lingo software. The number of jobs is 4, the number of machines is 2, and objective function value is 93. The sequence on the first machine is 1→ 4. The sequence on the second machine is 2→ 3.

3. GENETIC ALGORITHMS

Genetic algorithms (GAs) were developed by Holland in 1975 as artificial adaptive systems to simulate the natural evolution. Because of their effectiveness and efficiency in searching complex search spaces, they are increasingly used to handle NP-hard problems. The heart of GAs is the crossover operator which progressively constructs near-optimum solutions from good partial solutions [2]. In this paper regarding its NP-Hard nature, we apply a genetic algorithm approach to improve the objective function. We use binary coding for representing chromosomes. Each chromosome is made of zero-one genes. The length of the chromosome is the sum of the number of machines and the number of jobs multiplied by the number of machines *((N+1)*M)*. Because each machine has N+1 place, one place for job 0 and N place for other jobs. The first N+1 genes belong to machine number 1, the second N+1 genes belong to machine number 2, and so on. If a gene is equal to one, it means that a job is assigned to the corresponding machine. And if a gene is equal to zero, it means that a job is not assigned to that machine. Each group of genes belonging to one machine show the jobs assigned to that machine and the sequence on the machine. For example, we suppose that there are 3 machines and 4 jobs to be scheduled. The corresponding

	Machine 1	Machine 2	Machine 3
Position numbers on each machine	0 1 2 3 4	0 1 2 3 4	0 1 2 3 4
A Sample Chromosome	1 0 1 0 1	1 0 0 1 0	1 1 0 0 0

Figure 1. Chromosome coding and representation.

chromosome is represented as shown in Figure 1.

As it is shown, genes assigned to machine number 1 are 1, 0, 1, 0, and 1, respectively. Job 0 is assigned to each machine, so this gene is equal to one on each machine. In this example, we can see that job 2 and job 4 are assigned to machine number 1, Job 3 is assigned to Machine 2, and job 4 is assigned to Machine 3.

The initial chromosomes are created and developed at random. The digits (values of genes) ensure that each job is assigned to one machine. Some of genetic algorithm operators are presented and developed to be used in GA applications to a class of combinatorial optimization problems. Different mutation and crossover operators are described as follows:

3.1. Mutation Operators

3.1.1. Mutation for jobs First, a machine is randomly selected then two jobs are selected at random on that machine and replaced with each other (i.e., they are swapped) as depicted in Figure 2. The new created offspring (i.e., child) is checked to ensure that each job is assigned to one machine. If not, a repair strategy is applied and run in order to create feasible chromosome.

3.1.2. Mutation for machines First, a job number is randomly selected then two machines are selected at random. The selected job is replaced on two machines as depicted in Figure 2. If these two strings do not ensure the rule of the problem, a repair strategy is applied and run in order to create feasible chromosomes.

3.2. Crossover Operators

3.2.1. Simple crossover We consider two parents as the initial chromosomes. A random value, r, is created in the range of $[1, (N+1)*M]$. All digits from the first one to the r^{th} one are replaced on two chromosomes. Then, both offspring are checked with the rule to create feasible chromosomes.

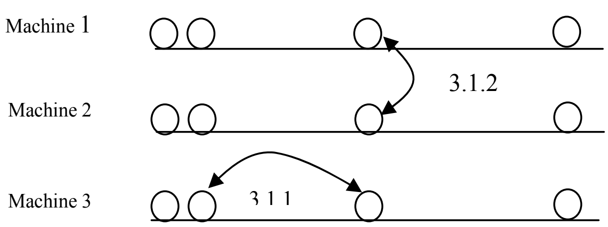

Figure 2. Mutation operators.

3.2.2. Partial crossover Two random values in the range of $[1, (N+1)*M]$ are created. If the two values are not equal, the digits between these values are replaced on two chromosomes. Then, both offspring should be checked to create feasible chromosome.

3.3. New Operators

3.3.1. Machine replacement Two machines are randomly selected and all the jobs on one of these machines are replaced by all the jobs on the other machine as shown in Figure 3.

3.3.2. Job replacement Two jobs are randomly selected and replaced on each machine as shown in Figure 4.

The proposed algorithm has been developed and coded by the Visual Basic 6.0 programming

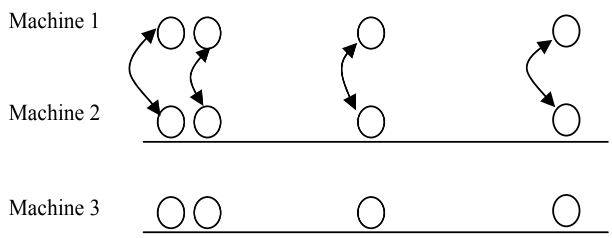

Figure 3. Operator of machine replacement.

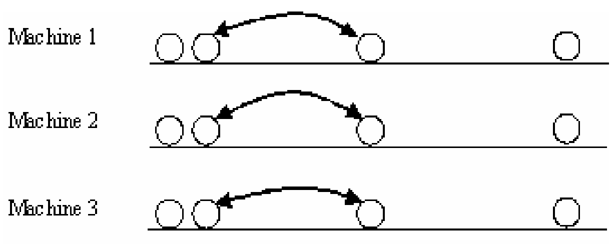

Figure 4. Operator of job replacement.

TABLE 1. Solving Some Examples Taken from Related Articles by the Proposed Algorithm.

Reference	Number of jobs	Number of machines	Sequence	Objective function value	Iteration needed
[7]	2	2	M2:1 - 2	41	5
[7]	4	2	M1: 1-2-3-4	81	8
[7]	6	3	M2: 4 – 5 M3: 1-2-3-6	104.5	15
[4]	4	2	M1: 2-3 M2: 1-4	45	12
[4]	10	3	M1: 1-2-3-5-7 M2: 4-6-8-9-10	1548.8	256
[3]	5	2	M1: 5 M2: 1-2-3-4	1727	15

TABLE 2. Comparing Some Randomly Generated Problem Results by Proposed Algorithm with Lingo.

Number of jobs	Number of machines	Objective function value	Time (Sec.)	Objective function value	Time (Sec.)	Iterations needed by proposed algorithm
		Lingo		Proposed algorithm		
5	2	146	11080	146	27sec	14
10	2	1185	4080	1185	40sec	29
15	2	1426.8	20100	1426.8	55sec	31
10	3	97	3900	97	47sec	21
20	3	2940	21900	2940	59sec	48

language. This proposed GA has run on Pentium IV with a 2.4 processor power. By using data given in some related articles which are simplified models of this paper, we apply this algorithm for a number of test problems and examples. Then, optimal results are obtained as illustrated in Table 1. In this paper, the new model is presented which considers setup times and earliness penalties. Setup times and earliness penalties are created from a uniform distribution [0,4,10], respectively. Some numerical examples are generated, and results of proposed algorithm are compared with Lingo software results shown in Table 2. To apply the operators, a random value is developed in each iteration. Each operator uses a specified range, so this random value specifies which operator should be used for each iteration. The initial population size concludes 20 chromosomes. The fitness of a chromos x in a population is given by the fitness function $f(x) = z_{max} - z(x)$, where $z(x)$ is the objective function value associated with x and z_{max} is the maximum objective function value observed in the population. Table 3 shows the different states in order to apply GA operators. Each state has three ranges for three kinds of operators and each kind has two sub-operators which are equally selected. Table 3 also shows the objective function values in iterations and for different states. The best objective function value is equal to 1783.2.

Figure 5 shows the graphical view of Table 3. State "e" shown by "×" symbol has reached better objective values (*i.e.*, they are very fast). States "a" and "c" are variant, so they may not be suitable states. States "d" and "f" seems to improve the

TABLE 3. Comparison of The Objective Function Values for Different States and Iterations.

State	Probability	Iterations							
		500	1500	2500	3500	5000	7000	7000	5000
a	0-0.2 0.2-0.8 0.8-1	1808.9	1855.5	1783.2	1789.5	1783.9	1789.5	1789.5	1783.9
b	0-0.7 0.7-0.8 0.8-1	1893.4	1836.4	1789.5	1783.2	1785.2	1783.2	1783.2	1785.2
c	0-0.3 0.3-0.5 0.5-1	1893.7	1883	1783.9	1801.9	1783.9	1785.2	1785.2	1783.9
d	0-0.1 0.1-0.5 0.5-1	1914.7	1823.4	1783.2	1783.2	1783.9	1783.2	1783.2	1783.9
e	0-0.3 0.3-0.7 0.7-1	1988.6	1785.2	1789.5	1783.9	1789.5	1785.2	1785.2	1789.5
f	0-0.5 0.5-0.7 0.7-1	1972.5	1855.5	1783.9	1785.2	1783.2	1783.9	1783.9	1783.2

objective function quite well. Each curve shows how much a state can improve the objective function and how much it can be effective.

4. CONCLUSION

In this survey, a new and complex mathematical model for selecting machines and scheduling jobs is proposed for a multi-criteria parallel machine scheduling problem. In this model, sequence dependent setup times and unequal penalties for earliness and tardiness are considered. Machines are supposed to be unrelated, so they have different speeds. This model is solved by applying a meta-heuristic method based on genetic algorithms. New operators are defined and different states to apply them are examined. This proposed algorithm is applied for previous models which have fewer criterions and it shows effective results. Regarding the results achieved from the different states applied to the algorithm, we can realize which ranges can improve the objective function value of the model more than the others. As this model is a comprehensive model, it can be applied to a single machine scheduling problem in which there are a number of identical parallel machines. In this model, preemption is not allowed but there are some industries which this can be studied and analyzed. Considering dynamic parameters such as dynamic due dates, can be considered in a future research.

5. REFERENCES

1. Gupta, J. N. D. and Ruiz-Torres, A. J., "Theory and methodology minimizing make-span subject to minimum total flow-time on identical parallel machines", ***European Journal of Operational Research***, Vol. 125, (2000), 370-380.
2. Sivrikaya-Serifoğlu, F. and Ulusoy, G., "Parallel machine scheduling with earliness and tardiness penalties", ***Computers and Operations Research***, Vol. 26, No. 8, (1999), 773-787.
3. Yi, Y. and Wang, D. W., "Soft computing for scheduling with batch setup times and earliness-tardiness penalties on parallel machines", ***J. of Intelligent Manufacturing***, Vol. 14, Nos. 3-4, (2003), 311-322.

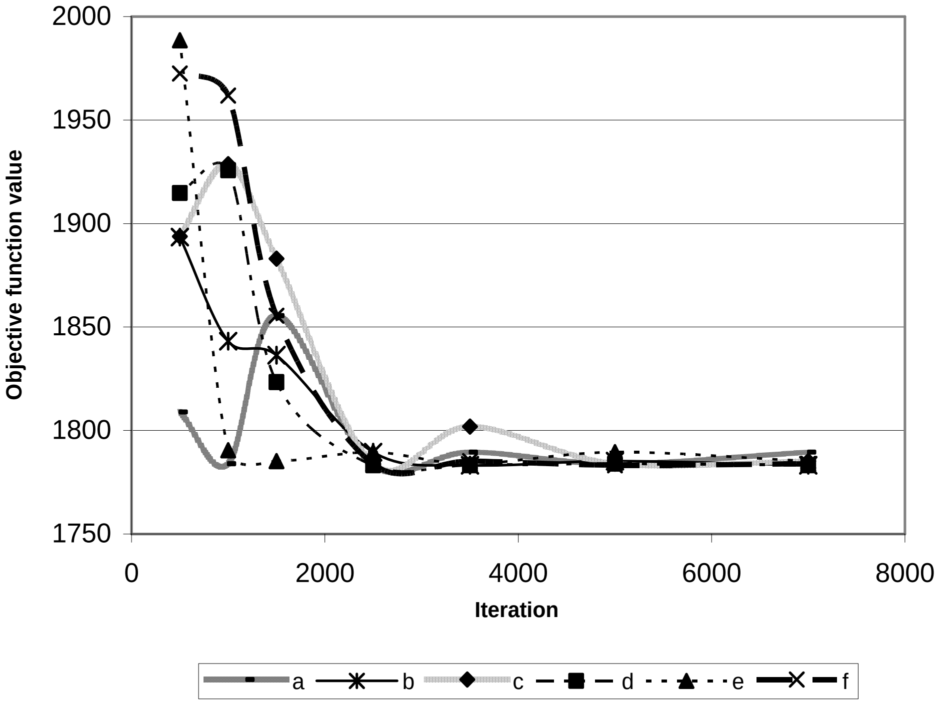

Figure 5. Different states for applying the operators.

4. Balakrishnan, N., Kanet, J. J. and Sir Sridharan, V., "Early/tardy scheduling with sequence dependent setups on uniform parallel machines", ***Computers and Operations Research***, Vol. 26, (1999), 127-141.
5. Allahverdi, A., Gupta, J.N.D. and Aldowaisan, T., "A review of scheduling research involving setup considerations", ***OMEGA, Int. J. Management Science***, Vol. 27, (1999), 219-239.
6. Cochran, J. K., Horng, S. M. and Fowler, J.W., "A multi-population genetic algorithm to solve multi objective scheduling problems for parallel machines", ***Computers and Operations Research***, Vol. 30, (2003), 1087-1102.
7. Cao, D., Chen, M. and Wan, G., "Parallel machine selection and job scheduling to minimize machine cost and job tardiness", ***Computers and Operations Research***, Vol. 32, No. 8, (2005), 1995-2012.
8. Bilge, Ü., Kıraç, F., Kurtulan, M. and Pekgün, P., "A tabu search algorithm for parallel machine total tardiness problem", ***Computers and Operations Research***, Vol. 31, No. 3, (2004), 397-414.
9. Liaw, C. F., Lin, Y. K., Cheng, C. Y. and Chen, M., "Scheduling unrelated parallel machines to minimize total weighted tardiness", ***Computers and Operations Research***, Vol. 30, (2003), 1777-1789.
10. Min, L. and Cheng, W., "Genetic algorithms for the optimal common due date assignment and the optimal scheduling policy in parallel machine earliness/ tardiness scheduling problems", ***Robotics and Computer-Integrated Manufacturing***, Vol. 22, No. 4, (2006), 279-287.
11. Koulamas, C. and Kyparisis, G. J., "Scheduling on uniform parallel machines to minimize maximum lateness", ***Operations Research Letters***, Vol. 26, No. 4, (2000), 175-179.
12. Azizoglu, M. and Kirca, O., "Tardiness minimization on parallel machines, ***International Journal of Production Economics***, Vol. 55, No. 2, (1998), 163-168.
13. Tavakkoli-Moghaddam, R., Moslehi, G., Vasei, M. and Azaron , A., "Optimal scheduling for a single machine to minimize the sum of maximum earliness and tardiness considering idle insert", ***Applied Mathematics and Computation***, Vol. 167, (2005), 1430–1450.