Computers and Mathematics with Applications 64 (2012) 2111–2117

Contents lists available at SciVerse ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Two meta-heuristic algorithms for solving multi-objective flexible job-shop scheduling with parallel machine and maintenance constraints☆

Vahid Majazi Dalfard [a], Ghorbanali Mohammadi [b,*]

[a] *Young Researchers Club, Kerman Branch, Islamic Azad University, Kerman, Iran*

[b] *Shahid Bahonar University, Department of Industrial Engineering, Kerman, Iran*

ARTICLE INFO

Article history:
Received 22 September 2011
Received in revised form 13 March 2012
Accepted 11 April 2012

Keywords:
Scheduling
Parallel machine
Meta-heuristics
LINGO software

ABSTRACT

There are different reasons, such as a preventive maintenance, for the lack of machines in the planning horizon in real industrial environments. This paper focuses on the multi-objective flexible job-shop scheduling problem with parallel machines and maintenance cost. A new mathematical modeling was developed for the problem. Two meta-heuristic algorithms, a hybrid genetic algorithm and a simulated annealing algorithm, were applied after modeling the problem. Then, solutions of these meta-heuristic methods were compared with solutions obtained by using the software LINGO for small-scale, medium-scale, and large-scale problems in terms of time and optimality. The results showed that the applied hybrid genetic and simulated annealing algorithms were much more effective than the solutions obtained using LINGO. Finally, solutions using the simulated annealing approach were compared with solutions of the hybrid genetic algorithm.

1. Introduction

In general, two major causes are known for the unavailability of a machine: accidents (machine failure or downtime) and security considerations (preventive sheet and overhaul prescheduled) [1]. The Multi-Objective Flexible Dynamic Job-Shop with Parallel Machines (MO-FDJSPM) problem with maintenance constraints is an optimization problem in discrete space. Since the problem is naturally non-convex and non-linear, the problem usually has a local optimum [2]. To solve the scheduling problem, meta-heuristics methods are the first choice. The genetic algorithm approach has better performance than other methods. Some researchers believe that the scheduling algorithm approach for solving these optimization problems is appropriate. Studies showed that premature convergence properties and being trapped in local optimal points are the two shortcomings of classical genetic algorithms [3]. Despite the maintenance constraint, the purpose of the current paper is to model and provide an efficient method for solving the MO-FDJSPM problem. The research problem considers maintenance constraints while considering a dynamic manufacturing environment, operational flexibility due to parallel machines, and a multi-criteria objective function.

In the planning horizon, there are various causes for machines to break down. These causes may be due to unforeseen damage and disabilities, preventive maintenance for specified periods of time, and overhaul operations [4,5]. Machines may also be out service due to unscheduled events [1]. Thus, lack of access to a machine includes two categories of stochastic and deterministic accesses. Adiri et al. showed that the single-machine scheduling problem with the certain period of

☆ The paper has been evaluated according to the old Aims and Scope of the journal.

* Corresponding author. Tel.: +98 9125700730.
E-mail address: ghorbanalim@yahoo.co.uk (G. Mohammadi).

0898-1221/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2012.04.007

inaccessibility to the machine is an NP-hard problem [6]. Preemption of operations means that the interruption of the process is allowed in order to carry out operational activities or maintenance and that there is no penalty for this interruption (the process could be continued without a penalty) [7]. Partial interruption of operations of the work in a certain time window is not possible unless the process is done after the time window [8]. In the regime of continuing the process in detail (S-R), the process can be done from the rest of process not from the scratch whenever the machine becomes available [9–13]. Schmidt, for the first time, introduced the parallel machine scheduling problem with a certain period of inaccessibility to different machines and the possibility of preemption [14]. Graves studied the single-machine scheduling problem with an indefinite period of inaccessibility and regime type of production (S-R) [15]. Schmidt studied single-machine, parallel-machine, and flow-shop scheduling problems with limitations regarding the accessibility of the machines [16]. Blazewicz et al. solved the scheduling problem for a flow shop with two machines and many periods of inaccessibility to each machine in a production type R using meta-heuristic algorithms [17]. Considering limited access to the machine and type R production, Xie and Wang developed a scheduling problem for a two-stage flexible flow shop with parallel machines [18]. Cheng and Liu presented the flow-shop scheduling problem considering no waiting and constraints on each machine [19]. Kubzin and Strusevich introduced a scheduling problem for a flow shop with two machines with no waiting and constraints on each machine, and then they solved it using an approximation algorithm [20]. Xie and Wang developed a scheduling problem for a flow shop with parallel machines while considering accessibility constraints to the machines in the type R production regime [21]. Breit developed the single-machine scheduling problem with constraints regarding access to the machine and the N-R production regime [22]. Wu and Lee presented a single-machine scheduling problem with the learning effect and a limitation on accessibility to machines in the R and N-R production regimes [23]. Zribi and Kamel discussed the job-shop scheduling problem with multi-purpose machines and limited access to machines with a type R production regime [24]. Zegordi and Rahimi analyzed the single-objective scheduling problem while considering limited access to machines with the N-R production regime and then solved it through applying a classical genetic algorithm [25].

2. Mathematical model

2.1. Problem definition

A mathematical model, mixed-integer nonlinear programming, is developed in this study. The model is comprised of N jobs and L processing steps in a dynamic workshop. Each work needs a number of processing steps to get finished. For example, the job i has o_i operations with a specific sequence. Each part i is released to the shop at a certain non-zero time r_i. Workstation l has a number of parallel machines m_l working at different rates. The operation o_{ik} is done on machine m_{lj} of available machines M_{ikl} and in the pre-specified workstation w_l with processing time p_{ikl}. R_{lj} maintenance activities need to be carried on machine m_{lj} during the planning horizon. Moreover, maintenance activity PM_{ljr} is completed in time t_{ljr} in a certain time window. Following are lists of the parameters and the decision variables, and the mathematical model.

Parameters.

o_i: Number of job i
o_{ik}: Operation kth of job i
r_i: Release time of ith part to the shop
w_l: Workstation l
m_l: Number of parallel machines at workstation l
m_{lj}: Machine j from workstation l
p_{iklj}: Processing time o_{ik} on m_{lj}
t_{mlj}: Availability time of m_{lj}
R_{lj}: Number of maintenance activities on m_{lj}
PM_{ljr}: Maintenance activities r on m_{lj}
t_{ljr}: Completion time of PM_{ljr}
U^E_{ljr}: Earliest completion time of PM_{ljr}
U^L_{ljr}: Latest completion time of PM_{ljr}
$[U^E_{ljr}, U^L_{ljr}]$: Window of time to complete PM_{ljr}
M_{ikl}: Set of machines available for processing operations o_{ik} at workstation l

Decision variables

c_{ik}: Completion time of o_{ik}
u_{ljr}: Completion time of PM_{ljr}

$$\upsilon_{ikljr} = \begin{cases} 1 & \text{if } c_{ik} \text{ do before doing } c_{ik} \\ 0 & \text{otherwise} \end{cases}$$

$$x_{iklj} = \begin{cases} 1 & \text{if } m_{lj} \text{ selected because of doing } c_{ik} \\ 0 & \text{otherwise} \end{cases}$$

$$y_{ikhq} = \begin{cases} 1 & \text{if } o_{ik} \text{ do before } o_{hq} \\ 0 & \text{otherwise.} \end{cases}$$

Objective function problems:

$$F = a_1F_1 + a_2F_2 + a_3F_3 \tag{1}$$

$$F_1 = C_{\max} = \mathrm{Max}\{C_i | i = 1, \dots, N\} \tag{2}$$

$$F_2 = \bar{F} = \frac{1}{N}\sum_{i=0}^{N} \max\{C_i - r_i\} \tag{3}$$

$$F_3 = \bar{T} = \frac{1}{N}\sum_{i=0}^{N} \max\{\beta_i(C_i - d_i), 0\}. \tag{4}$$

Constraints problem:

$$(C_k - C_{i,k-1}) \geq P_{iklj} \cdot X_{iklj} \tag{5}$$

$$k = 2, 3, O_i; \quad \forall i, l, j$$

$$(C_{ik} - C_{hq} - P_{iklj}) \cdot X_{iklj} \cdot X_{hqlj} \cdot (1 - y_{ikhq}) \geq 0 \tag{6}$$

$$\forall \{i, h | i \neq h\}, k, q, l, j$$

$$(C_{hq} - C_{ik} - p_{hqlj}) \cdot x_{iklj} \cdot x_{hqlj} \cdot y_{ikhq} \geq 0 \tag{7}$$

$$\forall \{i, h | i \neq h\}, k, q, l, j$$

$$(C_{ik} - u_{ljr} - P_{iklj} - r_i) \cdot X_{iklj} \cdot (1 - v_{ikljr}) \geq 0 \tag{8}$$

$$\forall i \cdot k, l, j, r$$

$$(u_{ljr} - C_{ik} - t_{ljr} - r_i) \cdot X_{iklj} \cdot v_{ikljr} \geq 0 \tag{9}$$

$$\forall i \cdot k, l, j, r$$

$$\sum_{l=1}^{L} \sum_{j \in M_{iklj}} X_{iklj} = 1 \quad \forall i, k \tag{10}$$

$$U_{ljr}^{E} \leq u_{ljr} \leq U_{ljr}^{L} \quad \forall l, j, r \tag{11}$$

$$C_{ik} \geq 0 \quad \forall i, k \tag{12}$$

$$u_{ljr} \geq 0 \quad \forall l, j, r \tag{13}$$

$$X_{iklj}, y_{ikhq}, v_{ikljr} \in \{0, 1\} \tag{14}$$

$$\forall \{i, h | i \neq h\}, k, q, l, j, r. \tag{15}$$

2.2. Model description

Eq. (1) shows the objective function of the problem as the minimization of the weighted sum of three equations (2) to (4) with determinant coefficients a_1, a_2, a_3. It should be noted that, in real manufacturing environments, values of a_1, a_2, a_3 are determined by experts, and these values could vary from one industry to another. In the present research, equal priorities are assumed; thus a_1, a_2, a_3 are all equal to $\frac{1}{3}$. In addition, the value of the penalty for late delivery of each unit of parts is set to 1 ($\beta_j = 1$). The inequalities of Eq. (5) ensure that the sequences of operations for different jobs do not interfere with each other. The set of constraints (6) and (7) simultaneously ensures that operations performed on a machine do not interfere with each other. The inequalities shown in Eqs. (8) and (9) express that maintenance activities and processing cannot be done on a machine concurrently. The set of these equations implicitly indicates the dynamics of manufacturing environment as parts are released to the shop at different times (in dynamic manner). Eq. (10) indicates that an available machine in each workstation for each operation is selected. The inequality of Eq. (11) states that the maintenance activities should be completed in the specified time window.

2.3. Complexity of the problem

Since the FDJSP with flexible operations is strongly NP-hard [26], the MO-FDJSPM with flexibility of parallel machines in a dynamic manufacturing environment with maintenance constraints is strongly NP-hard.

2.4. Method comparison

Now, the proposed algorithms of performance are evaluated in two modes. Since the flexible job-shop problem case is simple, the investigated number of courses available for the machine is zero ($R = 0$). Therefore, the meta-heuristic algorithm methods are compared with each other.

3. Simulated annealing (SA) algorithm

The main components of SA for implementation are as follows.

3.1. Creating the initial answer

1000 random answers each of length *h* were created and the answer with the best objective function was selected as the start point.

3.2. Initial temperature

Temperature as a parameter plays an important role for accepting or rejecting objective functions. The starting temperature should give enough room, in the early stages, for the selection of many undesirable answers. By doing so, both the possibility of development and variation of answers are guaranteed.

Actually the initial temperature tells us the range in which the answer can be gotten worse. In other words, it determines the probability of deterioration. The acceptance probability of each worsening the answer is $e^{\frac{-\text{Index of deterioration of the answer}}{\text{Temperature}}}$.

To have the index almost independent of the problem size, it must be set equal to $\frac{\text{Scale of deterioration of the objective function}}{\text{Objective function}}$.

3.3. Determining the rate of temperature decrease

For less probability of accepting unfavorable answers, the temperature should be decreased. This is achieved by changing temperature function $T_k = \alpha T_{k-1}, \alpha \prec 1$. In this paper, $\alpha = 0.95$ is selected.

3.4. Determining the way of creating a neighborhood

First, two time units were changed. Relocation would be acceptable if better results were obtained and the size of deterioration of objective function were found if it deteriorates. The index of deterioration of the objective function is obtained through the following equation:

$$\frac{\text{Scale of deterioration of the objective function}}{\text{Objective function}} = \text{Index of deterioration of the objective function}$$

A random number between 0 and 1 was generated through a uniform distribution.

If $e^{(-\text{Index of deterioration of the answer/Temperature})} > \text{rand}(0, 1)$, then the answer is deteriorated. Otherwise, another neighborhood will be chosen.

3.5. Determining the number of neighborhoods reviewed at each temperature

More iteration is necessary for better answers. These iterations should be determined in way to minimize the runtime. In addition, solutions must be favorable. Within the scope of this paper, the number of the iteration is constant, and equal to 1000.

3.6. Scale stop

The runtime of the calculation depends on the scale stop. Efficiency of the scale in determining the desired answer is noticeable. The algorithm ends when the answers at each temperature remain unchanged on increasing the temperature. This is called freezing state. This status is assumed as the scale stop. In this paper, the final temperature is assumed to be 0.002. Fig. 1 shows a diagram for simulated annealing.

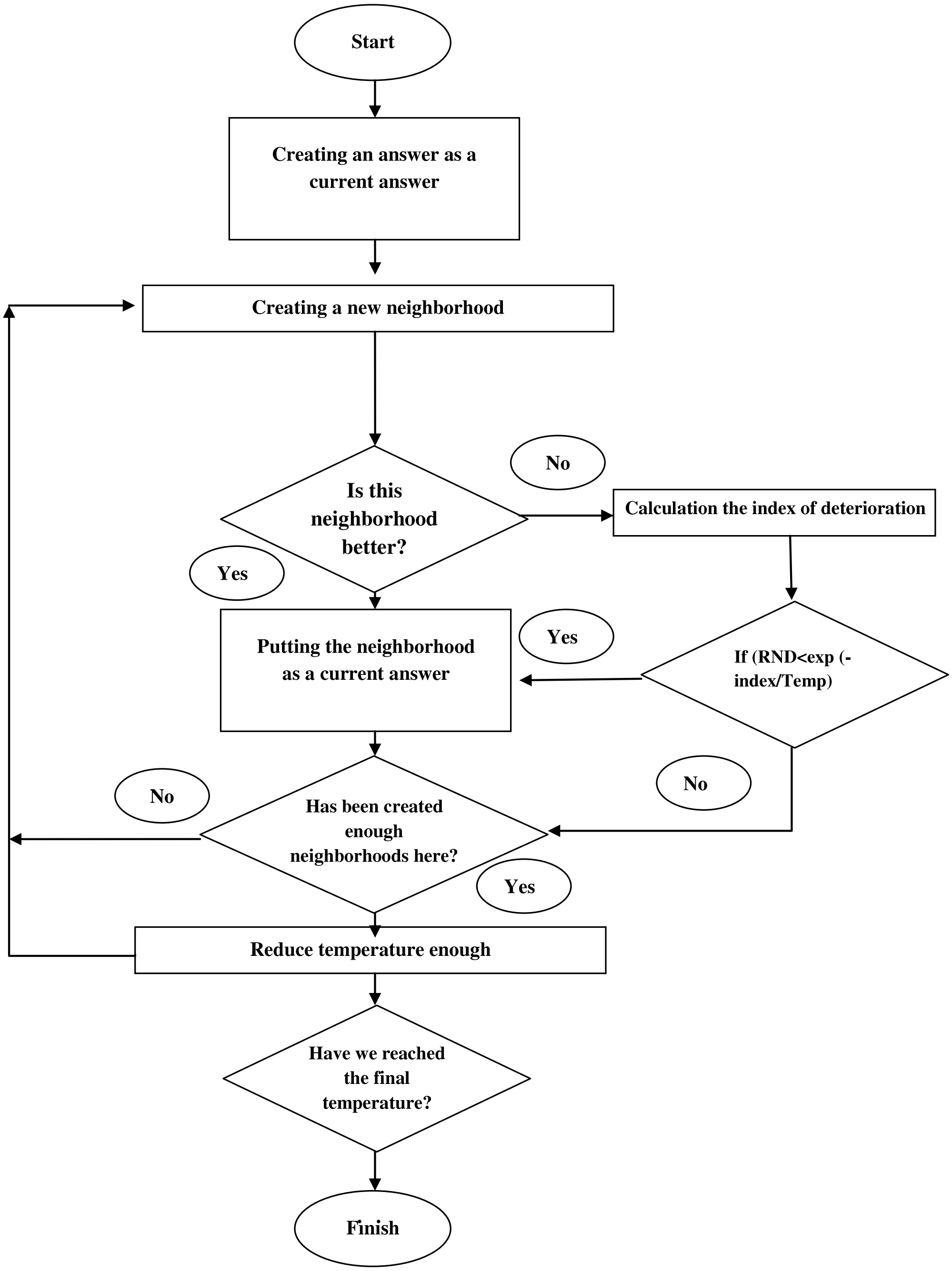

Fig. 1. A diagram used for simulated annealing.

4. Hybrid genetic algorithm (HGA)

The hybrid genetic algorithm (HGA) is used as a globally accepted search technique which is the same as simple genetic algorithm with a nuance of generation of an initial solution. In the HGA, some heuristics help to generate an initial feasible solution and then the procedure of simple genetic algorithm is used by the population according to population size. The HGA is described as follows [27].

Step 1: initialization and evaluation

(a) Generation of a initial sequence with special heuristics (SH) is known as one of the chromosomes of the population and the first step in this algorithm.

(b) Sequences are generated randomly as per the population size (Ps)

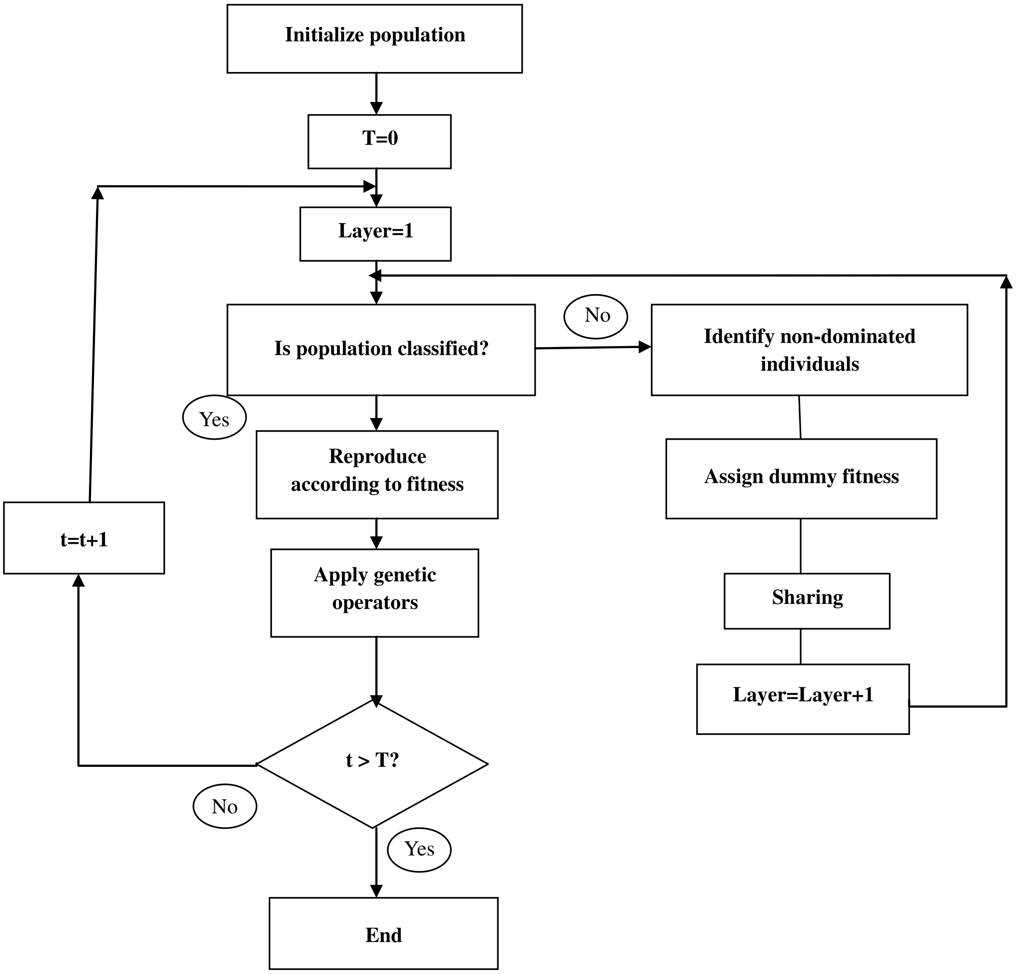

Fig. 2. Diagram of the HGA used in this problem.

(c) Combination of initial sequences obtained by special heuristics with a sequence that was generated randomly in order to form an equal population size sequence.

Step 2: Reproduction

A set of new populations is created through the algorithm. At each generation individuals are used by the algorithm to generate the next generation. In this process, the following steps are carried out in the algorithm.

(a) Fitness computation lets us score each member of the current population.
(b) Parents are selected based on the fitness function.
(c) The best-fitted individuals in the current population are used as an elite population which is utilized in the next population.
(d) Offspring are produced by crossing over from the pair of parents or a single parent is changed randomly (mutation).
(e) The current population and children are replaced to form the next generation.

Step 3: stopping limit.

A stopping condition is used to terminate the algorithm for certain numbers of generations [27]. A diagram of the HGA is shown in Fig. 2.

5. Results

12 experiments for solving multi-objective flexible job-shop scheduling with parallel machines and maintenance constraints in different dimensions, each produced randomly, were carried out using the SA and the HGA. For programming the SA and the HGA, MATLAB 7.5 was used. For running the algorithms, a PC (3.2 PIV, 2 GB RAM) was used. The results of experiments are listed in Table 1.

Table 1
The values of the objective functions obtained by the HGA and the SA.

Name of problem	Size of problem	Number of machines	Number of jobs	GA solution	Computational time GA (s)	SA solution	Computational time SA (s)	LINGO solution	Computational time (s)
Vmd1	Small	10	10	586	0.3	588	0.1	586	0.1
Vmd2		10	20	1,341	0.4	1,342	0.2	1,341	1
Vmd3		15	20	1,724	0.5	1,727	0.5	1,724	8.5
Vmd4		15	25	2,405	0.9	2,411	0.8	2,405	81
Vmd5	Medium	20	20	2,836	1.15	2,841	1.05	2,836	399.1
Vmd6		20	30	5,586	3.8	5,595	2.8	5,586	1748
Vmd7		30	40	13,802	7.9	13,824	6.1	13,802	6005
Vmd8		30	50	16,856	12.5	16,865	11.8	-	Until 3 h
Vmd9	Large	75	50	38,404	32	38,492	28.4	-	Until 5 h
Vm10		75	100	81,835	71	81,897	57	-	Until 5 h
Vmd11		100	100	122,741	93	122,993	69.4	-	Until 10 h
Vmd12		100	200	286,341	166	286,987	122	-	Until 10 h

6. Discussion and conclusions

A multi-objective flexible job-shop scheduling problem with parallel machines in the dynamic job shop combined with limitations on maintenance for the machines has been introduced. Two meta-heuristic algorithms were proposed for solving the structure based on the characteristics of the MO-FDJSPM, with maintenance constraint. One disadvantage of the classical genetic algorithm is that it was developed for parameters that control the dynamic changes during the optimization process. Numerical experiments were designed in 3 parts (small, medium and large problems). The efficiency of the applied algorithms in solving the problem was examined. The results of developing the genetic algorithm and using the simulated annealing algorithm indicate more speed and precision of such algorithms than obtained from LINGO software. Some future research directions could involve applying a time window for maintenance and also applying an accessibility constraint in the stochastic and constant forms concurrently.

References

[1] A. Allahverdi, H. Aydilek, Heuristics for the two-machine flowshop scheduling problem to minimize maximum lateness with bounded processing times, Comput. Math. Appl. 60 (2010) 1374–1384.
[2] T.C.E. Cheng, Minimizing the maximum deviation of job completion time about a common due-date, Comput. Math. Appl. 14 (1987) 279–283.
[3] A.B. Gur Mosheiov, A. Sariga, Scheduling a maintenance activity to minimize total weighted completion-time, Comput. Math. Appl. 57 (2009) 619–623.
[4] X. Zhanga, X. Yuan, Yanbin Yuan, Improved hybrid simulated annealing algorithm for navigation scheduling for the two dams of the three gorges project, Comput. Math. Appl. 56 (2008) 151–159.
[5] Y.B. Yunqiang, X. Dehua, Some single-machine scheduling problems with general effects of learning and deterioration, Comput. Math. Appl. 61 (2011) 100–108.
[6] I. Adiri, J. Bruno, E. Frostig, A.H.G. Rinnooy-Kan, Single machine flow-time scheduling with a single breakdown, Acta Inform. 26 (1989) 679–696.
[7] H. Aydilek, A. Allahverdi, Two-machine flowshop scheduling problem with bounded processing times to minimize total completion time, Comput. Math. Appl. 59 (2010) 684–693.
[8] X. Jinxing, X. Wang, Complexity and algorithms for two-stage flexible flowshop scheduling with availability constraints, Comput. Math. Appl. 50 (2005) 1629–1638.
[9] J.B. Wanga, Single-machine scheduling with general learning functions, Comput. Math. Appl. 56 (2008) 1941–1947.
[10] J.B. Wanga, Single machine scheduling with decreasing linear deterioration under precedence constraints, Comput. Math. Appl. 58 (2009) 95–103.
[11] J.B. Wanga, D. Wanga, L.Y. Wanga, L. Lin, Na Yina, W.W. Wanga, Single machine scheduling with exponential time-dependent learning effect and past-sequence-dependent setup times, Comput. Math. Appl. 57 (2009) 9–16.
[12] J.B. Wanga, W. Ming-Zheng, Single machine multiple common due dates scheduling with learning effects, Comput. Math. Appl. 60 (2010) 2998–3002.
[13] V. Mania, C.C. Pei, H.C. Shih, Hsin Chenc, Bi-criteria single machine scheduling problem with a learning effect: Aneja–Nair method to obtain the set of optimal sequences, Comput. Math. Appl. 58 (2009) 39–47.
[14] G. Schmidt, Scheduling on semi-identical processors, Z Oper. Res. Meth. Model. Oper. Res. 28 (1984) 153–162.
[15] S. Graves, A review of production scheduling, Oper. Res. 29 (1999) 646–675.
[16] G. Schmidt, Scheduling with limited machine availability, European J. Oper. Res. 121 (2000) 1–15.
[17] J. Blazewicz, E. Pesch, M. Sterna, The disjunctive graph machine representation of the job shop scheduling problem, European J. Oper. Res. 127 (2000) 317–331.
[18] J. Xie, X. Wang, Branch and bound algorithm for flexible flowshop with limited machine availability, Asian Info. Sci. Life. 1 (2002) 241–248.
[19] T.C.E. Cheng, Z. Liu, Approximability of two- machine flow shop scheduling with availability constraints, Oper. Res. Lett. 31 (2003) 319–322.
[20] M.A. Kubzin, V.A. Strusevich, Two-machine flow shop no-wait scheduling with machine maintenance, A Quart. J. Oper. Res. 3 (2005) 303–313.
[21] J. Xie, X. Wang, Complexity and algorithms for two-stage flexible flow shop scheduling with availability constraints, Comput. Math. Appl. 50 (2005) 1629–1638.
[22] J. Breit, Improved approximation for non-preemptive single machine flow-time scheduling with an availability constraint, European J. Oper. Res. (2006) 1–9.
[23] C.C. Wu, W.C. Lee, A note on single-machine scheduling with learning effect and an availability constraint, Int. J. Adv. Manuf. Tech. 33 (2006) 540–544.
[24] N. Zribi, A.E. Kamel, P. Borne, Minimizing the makespan for the MPM job-shop with availability constraints, Int. J. Prod. Econ. 112 (2007) 1–10.
[25] H. Zegordi, M. Rahimi, Job shop scheduling with maintenance constraint, J. Sharif. 46 (2010) 131–137.
[26] I. Kacem, S. Hammadi, P. Borne, Approach by localization and multi objective evolutionary optimization for flexible job-shop scheduling problems, IEEE Trans. Syst. Man. Cybern. B Cybern. 32 (2002) 1–13.
[27] D. Ashwani, C. Pankaj, Hybrid genetic algorithm for multicriteria scheduling with sequence dependent set up time, Int. J. Eng. 3 (2010) 510–520.