

A genetic algorithm for preemptive scheduling of a single machine

Amir-Mohammad Golmohammadi^{a*}, Hamid Bani-Asadi^b, Hamed Jafar Zanjani^c and Hamid Tikani^d

^aPh.D Student of Industrial Engineering, Department of Industrial Engineering, University of Yazd, Yazd, Iran

^bGraduate student of Industrial Engineering, Department of Industrial Engineering, Naragh Branch, Islamic Azad University, Naragh, Iran

^cAssistant Professor, Department of Industrial Engineering, Naragh Branch, Islamic Azad University, Naragh, Iran

^dGraduate student of Industrial Engineering, Department of Industrial Engineering, University of Yazd, Yazd, Iran

CHRONICLE

Article history:

Received November 4 2015

Received in Revised Format

February 21 2016

Accepted March 21 2016

Available online

March 22 2016

Keywords:

Preemption

Single machine scheduling

Work in process

Genetic algorithm

ABSTRACT

This paper presents a mathematical model for scheduling of a single machine when there are preemptions in jobs. The primary objective of the study is to minimize different objectives such as earliness, tardiness and work in process. The proposed mathematical problem is considered as NP-Hard and the optimal solution is available for small scale problems. Therefore, a genetic algorithm (GA) is developed to solve the problem for large-scale problems. The implementation of the proposed model is compared with GA for problems with up to 50 jobs using three methods of roulette wheel sampling, random sampling and competition sampling. The results have indicated that competition sampling has reached optimal solutions for small scale problems and it could obtain better near-optimal solutions in relatively lower running time compared with other sampling methods.

© 2016 Growing Science Ltd. All rights reserved

1. Introduction

Nowadays, there is a growing trend on shortage of resources, equipment, investments and facilities. Therefore, it is necessary to have optimal allocation of resources using different scheduling techniques (Liao & Cheng, 2007). In fact, a good scheduling plays essential role on development of more efficient production systems. Single machine scheduling has been widely used in different industries such as cement, petrochemical, etc. Earliness and tardiness are two important minimization strategies in most scheduling problems. In fact, these types of penalties are common in industrial settings where early job completion may cause the cash commitment to resources in a time frame earlier than required, giving rise to early completion penalties. Tardiness penalties may occur for different reasons such as loss of customer goodwill, opportunity expenses of lost sales, and direct cash penalties. Earliness expenses may hurt the performance measure, and this non-regularity discourages one from looking for solutions to this type of problem. Davis and Kanet (1993) investigated a single-machine scheduling problem by considering penalties for early and tardy completion of jobs.

* Corresponding author.

E-mail: amir.m.golmohammadi@stu.yazd.ac.ir (A.-M. Golmohammadi)

Szwarc and Mukhopadhyay (1996) presented an efficient technique to solve an earliness-tardiness single machine n job scheduling model where idle times was allowed and the earliness and tardiness penalties were proportional to the processing times of the jobs. Lenstra et al. (1997) showed that a single job scheduling problem is NP-Hard. Ahmadizar and Farhadi (2015) considered a single-machine scheduling problem in which jobs were released in various times but delivered in batches and a due window was related to each job. They aimed to schedule the jobs, to form them into batches and to select the delivery date of each batch to minimize the sum of earliness, tardiness, holding as well as delivery costs. Bülbül et al. (2007) investigated a single machine earliness/tardiness scheduling problem with general weights, ready times and due dates and the method is based on a time-indexed preemptive relaxation of the problem.

Hendel et al. (2009) studied the concept of preemption in scheduling, with earliness and tardiness penalties where the earliness costs was dependent on the start times of the jobs. They proposed an efficient representation of dominant schedules to measure the most appropriate schedule for a given representation. Sourd and Kedad-Sidhoum (2003) addressed the one-machine problem where the jobs had distinct due dates, earliness costs, and tardiness costs. Sun et al. (1999) investigated the NP-hard problem of scheduling N independent jobs on a single machine with release dates, due dates, sequence dependent setup times, and no preemption to minimize the weighted sum of squared tardiness. Valente (2008) presented an exact method for the single machine scheduling problem with linear early and quadratic tardy penalties.

Valente and Gonçalves (2009) proposed a genetic algorithm method for the single machine scheduling problem with linear earliness and quadratic tardiness penalties. Khorshidian et al. (2011) is associated with the concept of preemption in just-in-time single machine scheduling problem, with permissible machine idle time. Runge and Sourd (2009) investigated a new technique for the one-machine earliness-tardiness scheduling problem where jobs could be interrupted. They also proposed a timing method and a local search algorithm based on this timing to reach a good feasible solutions. M'Hallah (2007) proposed a hybrid heuristic to minimize total earliness and tardiness on a single machine. Kazemi et al. (2012) considered preemption and idle time in a single machine scheduling problem with just-in-time (JIT) approach. Wang et al. (2011) considered a single-machine scheduling problem with linear decreasing deterioration in which the due dates were determined by the equal slack method.

2. The proposed study

In this paper, we propose a single machine job preemptive scheduling with earliness tardiness constraints. In addition, we also consider the cost of work in process.

Assumption

1. Preemption and interruption are permitted.
2. At any time up to one work can be assigned to each machine.
3. Processing time of each job is deterministic.
4. There are certain number of jobs at each time.
5. Work in process and its cost are considered during the scheduling.
6. Start time, processing time and due dates are known in advance.
7. If operations of a job is interrupted, a cost of work in process is considered.
8. There is machine breakdown.
9. There is a cost for job preemption.

Indices

N : Number of jobs, $i = 1, \dots, N$

J : Number of periods, $j = 1, \dots, J$

Parameters

P_i	Processing time of job i
D_i	Due date of job i
α_i	Penalty of earliness of job i
β_i	Penalty of tardiness of job i
γ_i	Penalty for maintaining job i
η_i	Penalty for facing preemption of job i
θ_i	Penalty of earliness of each earliness job i
π_i	Penalty of tardiness of each tardiness job i
M	A big number

Variables

C_i	Completion time of job i
S_i'	Starting time of job $i, j = 2$
S_i''	Starting time of job $i, j = 1$
S_i	Starting time of job i
E_i	Earliness permitted for job i
T_i	Tardiness permitted for job i
x_{ij}	A binary variable, which is one if job i is processed at period j , and zero, otherwise
σ_{ij}	A binary variable, which is one if $x_{ij} = 1$ and zero, otherwise
Y_i	A binary variable, which is one if there is earliness on job i , and zero, otherwise
K_i	A binary variable, which is one if there is tardiness on job i , and zero, otherwise

Mathematical model

$$\min z = \sum_{i=1}^N \alpha_i E_i + \sum_{i=1}^N \beta_i T_i + \frac{1}{2} \sum_{i=1}^N \eta_i \left[\left(\sum_{i=1}^N |x_{ij} - x_{i,j+1}| \right) + x_{ij} + x_{iL} - 2 \right] + \sum_{i=1}^N (C_i - P_i - S_i) \gamma_i + \sum_{i=1}^N \theta_i Y_i + \sum_{i=1}^N \pi_i K_i \quad (1)$$

subject to

$$E_i = \max(0, D_i - C_i) \quad \forall i = 1, \dots, N \quad (2)$$

$$T_i = \max(0, C_i - D_i) \quad \forall i = 1, \dots, N \quad (3)$$

$$\sum_{i=1}^N x_{ij} \leq 1 \quad \forall j = 1, \dots, J \quad (4)$$

$$\sum_{j=1}^J x_{ij} = P_i \quad \forall i = 1, \dots, N \quad (5)$$

$$D_i - C_i \leq M \cdot Y_i \quad \forall i = 1, \dots, N \quad (6)$$

$$C_i - D_i \leq M \cdot K_i \quad \forall i = 1, \dots, N \quad (7)$$

$$C_i = \max_j (j \cdot x_{ij}) \quad \forall i = 1, \dots, N \quad (8)$$

$$S_i' = \min_j ((j-1) \sigma_{ij}) \quad \forall i = 1, \dots, N \quad \forall j = 2, \dots, J \quad (9)$$

$$S_i'' = \min_j (j \sigma_{ij}) - 1 \quad \forall i = 1, \dots, N \quad j = 1 \quad (10)$$

$$S_i = \min(S'_i, S''_i) \quad \forall i = 1, \dots, N \quad \forall j = 2, \dots, J \quad (11)$$

$$\sigma_{ij} = M(1 - x_{ij}) + 1 \quad \forall i = 1, \dots, N \quad \forall j = 2, \dots, J \quad (12)$$

$$Y_i, K_i, x_{ij} \in \{0,1\} \quad \forall i = 1, \dots, N \quad \forall j = 2, \dots, J \quad (13)$$

$$E_i, T_i, \sigma_{ij}, S_i, S'_i, S''_i, C_i \geq 0 \quad \forall i = 1, \dots, N \quad \forall j = 2, \dots, J$$

The objective function given in Eq. (1) consists of five parts of earliness, tardiness, job preemption, work in process, the number of earliness and the number of tardiness. Eq. (2) and Eq. (3) are associated with jobs with earliness and tardiness, respectively. According to Eq. (4), in each period, only one job needs to be processed and according to Eq. (5), each job is divided into P_i process in each period. Eq. (6) and Eq. (7) indicate the number of earliness and tardiness for each job. Eq. (8) shows the time the process of each job ends. Eqs. (9-11) demonstrate the starting time of jobs. Eq. (12) determines the relationship between σ_{ij} and x_{ij} and finally Eq. (13) determines the nature of variables in terms of binary/continuous. Note that $J = \max\{\max(D_i), \sum_{i=1}^N P_i\}$, which makes the set of feasible solution to be non-empty and Table 1 presents the input data used for the next section.

Table 1
The input data

<i>job i</i>	P_i	d_i	α_i	β_i	γ_i
<i>i</i> = 1	20	82	9	17	16
<i>i</i> = 2	65	126	6	8	11
<i>i</i> = 3	15	162	12	3	6
<i>i</i> = 4	5	135	1	19	7
<i>i</i> = 5	50	112	18	11	10

3. Genetic algorithm

As explained earlier, the proposed model of this paper is formulated as mixed integer programming and is classified as an NP-Hard problem (Lenstra et al., 1997). Therefore, we use genetic algorithm (GA) to solve the resulted model for problems in large-scales. GA is known as the most famous metaheuristic algorithm in the world. Holland (1992) is believed to be the first who introduced GA and it has been widely implemented in different classes of problems. GA is considered as a metaheuristic search, which mimics the process of natural selection. The processed is routinely implemented to generate useful solutions to optimization and search problems. It belongs to the bigger class of evolutionary algorithms (EA), which create solutions to optimization problems based on techniques inspired by natural evolution, including inheritance, mutation, selection, and crossover. The chromosome of the proposed GA method is a string where each cell represents a job as follows,

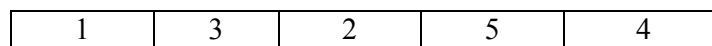


Fig. 1. The chromosome

The population of the proposed GA method consists of J elements with $\sum_{i=1}^I p_i < J \leq \max_i(D_i)$. The fitness function is evaluated based on Eq. (1). In addition, the earliness and tardiness are computed as follows,

$$E_i = \max(0, D_i - C_i) \quad (14)$$

$$T_i = \max(0, C_i - D_i) \quad (15)$$

In addition, the number of preemption is evaluated by $[(\sum_{i=1}^N |x_{ij} - x_{i,j+1}|) + x_{ij} + x_{iL} - 2]$, time for maintaining work in process is computed by $(C_i - P_i - S_i)$, and finally, $D_i - C_i \leq M.Y_i$ and $C_i - D_i \leq M.K_i$ are used to compute the earliness and tardiness. To select the parents, we use three methods of roulette wheel sampling, random sampling and competition sampling. Moreover, we use three methods of one-point, two-point and three-point crossover operators shown in Fig. 2, Fig. 3 and Fig. 4 as follows,

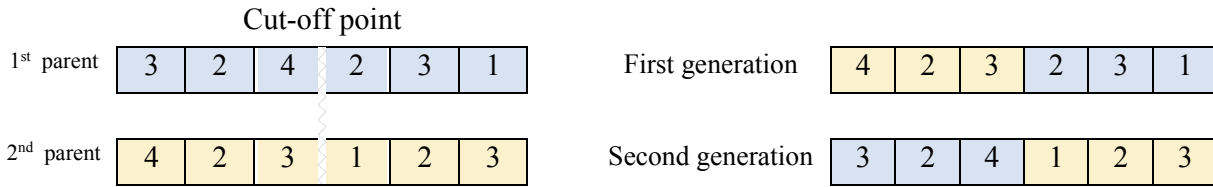


Fig. 2. one-point crossover

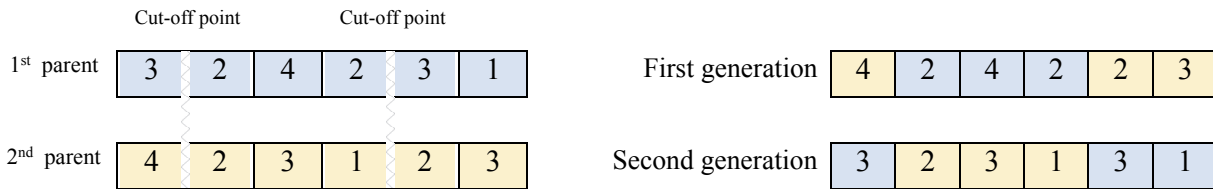


Fig. 3. Two-point crossover

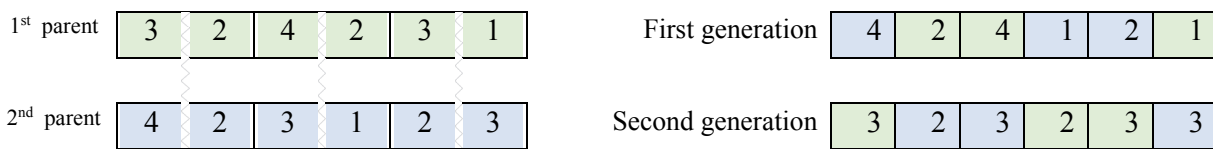


Fig. 4. Three-point crossover

The operations of the mutation operator is performed to escape from local solutions. Fig. 5 demonstrates the process of mutation operation used in this paper.



Fig. 5. Mutation operation

Parameter setting

The proposed study of this paper uses GAMS software package to solve the problems in small and medium sizes and GA method is implemented for large scale problems. In this paper, problems with 4,

6, and 8 jobs are considered small size, problems with 10, 15 and 20 jobs are considered medium size and problems with 30, 40 and 50 jobs are considered as large scale problems.

The processing time follows a uniform distribution with [1 20], earliness and tardiness times also follow uniform distribution with [1 10] and finally, due dates are defined as follows,

$$[d_{min} - \lambda, d_{min} + \lambda]$$

$$d_{min} = P \times (1 - TEF)$$

$$P = \sum_{i=1}^n p_i$$

$$\lambda = P \times \left(\frac{RDD}{2}\right)$$

Here, TEF and RDD are considered for suitable delivery time and the values of 0.2 and 0.5 are considered for TEF and 0.5 and 0.8 are considered for RDD.

Parameter tuning

In order to improve the performance of the proposed study, we use Taguchi method (Montgomery, 2008). For crossover probability (P_c) we use three levels of 0.8, 0.85 and 0.9. In addition, for mutation probability (P_m) we use three levels of 0.03, 0.06 and 0.09. Number of population N_p is considered in two levels of 100 and 200. Moreover, number of generation is considered in one level with $N_g = 100$. Finally, $P_c = 0.8, P_m = 0.06, N_p = 100$ are considered as default values. Fig. 6 shows details of the implementation of the Taguchi method.

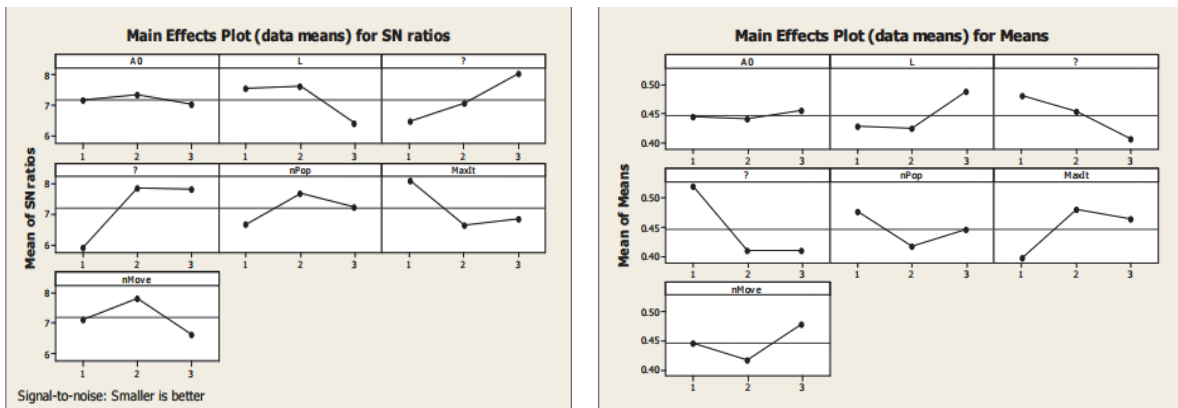


Fig. 6. The results of Taguchi method

All computations for the implementation of GA have been coded in MATLAB2011 and GAMS2 23.9 (CPLEX) on a personal computer with AMD 4600 GHz with one gigabyte memory.

4. Results

Table 2 presents details of the implementation of the proposed study in GAMS and GA with three methods of roulette wheel sampling, random sampling and competition sampling. In Table 2, the first column shows the problem number, the second column presents the number of jobs and optimal solution (O.S) and execution time (CPU) for three methods used in GA are given next. Finally, the optimal

solution are also presented in the last two columns in terms of objective function and CPU time. All times are given in minutes and hours.

Table 2

The results of the implementation of GAMS and GA

Prob.	Jobs	Competition		Roulette wheel		Random sampling		GAMS	
		O.S	CPU	O.S	CPU	O.S	CPU	O.S	CPU
J# 1	4	58	3	58	6	58	6	58	1
J# 2	4	154	4	154	3	154	2	154	3
J# 3	6	155	6	155	6	155	7	155	50
J# 4	6	184	4	184	4	184	3	184	18
J# 5	8	341	14	341	12	341	14	341	20:32
J# 6	8	280	17	280	18	280	17	280	7:52
J# 7	10	196	22	196	21	196	19	196	1:34:41
J# 8	10	269	17	269	19	269	23	269	2:18:57
J# 9	15	281	1:09	281	1:02	281	1:12	281	4:26:48
J#10	15	618	54	618	1:09	618	1:05	611	3:05:12
J#11	20	733	2:00	733	2:25	733	3:56	---	5:00:00
J#12	20	831	2:31	835	2:16	837	3:08	---	5:00:00
J#13	30	1368	7:16	1389	19:35	1403	9:54	---	5:00:00
J#14	30	1392	15:30	1397	15:39	1395	18:43	---	5:00:00
J#15	40	2147	28:34:00	2101	30:26:00	2214	29:25:00	---	5:00:00
J#16	40	1678	25:46:00	1519	24:27:00	1568	27:38:00	---	5:00:00
J#17	50	6526	50:48:00	6568	1:37:43	6634	1:17:32	---	5:00:00
J#18	50	5396	47:58:00	5412	1:23:34	5425	1:25:48	---	5:00:00

As we can observe from the results of Table 2, both GA and GAMS could solve the problems in small and medium sizes, successfully and they reached optimal solutions. However, for large-scale problems, only GA could solve the problem. The other observation from the results of Table 2 is that competition method used in GA to generate initial population could reach better solution in less amount of time compared with other two sampling techniques.

4. Conclusion

In this paper, we have presented a mathematical model for scheduling of a single machine when there were preemptions in jobs. The primary objective of the study was to minimize different objectives such as earliness, tardiness, work in process, etc. The proposed mathematical problem was NP-Hard and the optimal solution was available for small scale problems. Therefore, a genetic algorithm (GA) has been developed to solve the problem for large-scale problems. The implementation of the proposed model was compared with GA for problems with up to 50 jobs using three methods of roulette wheel sampling, random sampling and competition sampling. The results have indicated that competition sampling has reached optimal solutions for small scale problems and it could obtain better near-optimal solutions in relatively lower running time compared with other sampling methods.

Acknowledgement

The authors would like to thank the anonymous referees for constructive comments on earlier version of this paper.

References

- Ahmadizar, F., & Farhadi, S. (2015). Single-machine batch delivery scheduling with job release dates, due windows and earliness, tardiness, holding and delivery costs. *Computers & Operations Research*, 53, 194-205.

- Bülbül, K., Kaminsky, P., & Yano, C. (2007). Preemption in single machine earliness/tardiness scheduling. *Journal of Scheduling*, 10(4-5), 271-292.
- Davis, J. S., & Kanet, J. J. (1993). Single-machine scheduling with early and tardy completion costs. *Naval Research Logistics (NRL)*, 40(1), 85-101.
- Hendel, Y., Runge, N., & Sourd, F. (2009). The one-machine just-in-time scheduling problem with preemption. *Discrete Optimization*, 6(1), 10-22.
- Holland, J., (1992). *Adaptation in natural and artificial systems*: MIT Press. Cambridge, MA.
- Kazemi, M., Nikoofarid, E., Aalaei, A., & Kia, R. (2012). Just-in-time preemptive single machine problem with costs of earliness/tardiness, interruption and work-in-process. *International Journal of Industrial Engineering Computations*, 3(3), 321-336.
- Khorshidian, H., Javadian, N., Zandieh, M., Rezaeian, J., & Rahmani, K. (2011). A genetic algorithm for JIT single machine scheduling with preemption and machine idle time. *Expert Systems with Applications*, 38(7), 7911-7918.
- Lenstra, J. K., Kan, A. R., & Brucker, P. (1977). Complexity of machine scheduling problems. *Annals of Discrete Mathematics*, 1, 343-362.
- Liao, C. J., & Cheng, C. C. (2007). A variable neighborhood search for minimizing single machine weighted earliness and tardiness with common due date. *Computers & Industrial Engineering*, 52(4), 404-413.
- M'Hallah, R. (2007). Minimizing total earliness and tardiness on a single machine using a hybrid heuristic. *Computers & Operations Research*, 34(10), 3126-3142.
- Montgomery, D. C. (2008). *Design and analysis of experiments*. John Wiley & Sons.
- Runge, N., & Sourd, F. (2009). A new model for the preemptive earliness–tardiness scheduling problem. *Computers & Operations Research*, 36(7), 2242-2249.
- Sourd, F., & Kedad-Sidhoum, S. (2003). The one-machine problem with earliness and tardiness penalties. *Journal of scheduling*, 6(6), 533-549.
- Sun, X., Noble, J. S., & Klein, C. M. (1999). Single-machine scheduling with sequence dependent setup to minimize total weighted squared tardiness. *IIE transactions*, 31(2), 113-124.
- Szwarc, W., & Mukhopadhyay, S. K. (1996). Earliness and tardiness single machine scheduling with proportional weights. *Journal of Global Optimization*, 9(3-4), 227-238.
- Valente, J. M. (2008). An exact approach for the single machine scheduling problem with linear early and quadratic tardy penalties. *Asia-Pacific Journal of Operational Research*, 25(02), 169-186.
- Valente, J. M., & Gonçalves, J. F. (2009). A genetic algorithm approach for the single machine scheduling problem with linear earliness and quadratic tardiness penalties. *Computers & Operations Research*, 36(10), 2707-2715.
- Wang, X. R., Huang, X., & Wang, J. B. (2011). Single-machine scheduling with linear decreasing deterioration to minimize earliness penalties. *Applied Mathematical Modelling*, 35(7), 3509-3515.