Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 198950, 6 pages
http://dx.doi.org/10.1155/2015/198950

Research Article

Scheduling Jobs with Maintenance Subject to Load-Dependent Duration on a Single Machine

Yawei Qi,[1] Long Wan,[1] and Zhigang Yan[2]

[1] *School of Information Technology, Jiangxi University of Finance and Economics, Nanchang 330013, China*
[2] *Accounting School, Jiangxi University of Finance and Economics, Nanchang 330013, China*

Correspondence should be addressed to Yawei Qi; ivey09@163.com

Received 23 August 2014; Accepted 18 September 2014

Academic Editor: Chin-Chia Wu

This paper investigates a scheduling problem on a single machine with maintenance, in which the starting time of the maintenance is given in advance but its duration depends on the load of the machine before the maintenance. The goal is to minimize the makespan. We formulate it as an integer programming model and show that it is NP-hard in the ordinary sense. Then, we propose an FPTAS and point out that a special case is polynomial solvable. Finally, we design fast heuristic algorithms to solve the scheduling problem. Numerical experiments are implemented to evaluate the performance of the proposed heuristic algorithms. The results show the proposed heuristic algorithms are effective.

1. Introduction

We consider the following scheduling problem. Given a set of jobs $\mathscr{J} = \{J_1, J_2, \dots, J_n\}$ to be nonpreemptively processed on a single machine, all of which are available at time zero. To prevent the machine breakdown, we must start maintenance at a prefixed time, but the duration of maintenance depends on the load of the machine before the maintenance, that is, the sum of processing times of jobs that is scheduled before the maintenance, which is a nonnegative and nondecreasing function on the load. The goal is to schedule all the jobs to minimize the maximum completion time (makespan).

For a given schedule, write p_j for the processing time and C_j for the completion time of job J_j, $j = 1, 2, \dots, n$, respectively. Let s denote the starting time of maintenance, which is given in advance, and let l denote the load of the machine before the maintenance and let d denote the duration of maintenance. Then, $d = f(l)$, where $f(\cdot)$ is a nonnegative and nondecreasing function on l (without loss of generality, we assume that function $f(\cdot)$ can be computed in polynomial time). Furthermore, let $C_{\max}$ ($= \max_{j=1,2,\dots,n} C_j$) denote the maximum completion time (i.e., makespan). Using the three-field notation proposed by Graham et al. [1], we denote our problem as $1, \mathrm{LDM} \| C_{\max}$, where "LDM" denotes the maintenance subject to load-dependent duration.

As a practical example, consider the casting production in steel-making plant [2], in which the steel slabs used for downstream hot rolling operation are melted by the caster. Each steel slab can be viewed as a job and the caster can be viewed as a machine. At a given time, we must refill the fuels for the caster, which can be regarded as maintenance. The time for refilling the fuels is its duration, which depends on the consumed fuels before the maintenance.

The above problem roughly falls into a class of scheduling problem which is referred to as scheduling with machine unavailable constraints. Since 1996, scheduling with machine unavailable constraints has been investigated extensively. For the related surveys, we refer the readers to Schmidt [3] and Ma et al. [4]. In the following, we only review the previous works very related to our problem. Lee [5] considered the scheduling problem with unavailable intervals on a single machine, in which the starting times and the durations of the unavailable intervals are given in advance with the goal to minimize the makespan and showed the problem is NP-hard with one unavailable interval and strong NP-hard with multiple unavailable intervals. He et al. [6] proposed a fully polynomial time approximation scheme with one unavailable interval for the problem proposed by Lee [5]. Breit et al. [7] showed that no polynomial time approximation algorithm with a fixed performance ratio exists with two intervals for the problem proposed by Lee [5]. Wu and Lee [8], Gawiejnowicz [9], and Ji et al. [10] extended the problem proposed by Lee [5] to job deterioration case, where the processing time of a job is a simple linear function of its starting time. For other related problems involving machine unavailable constraints, see Lee [11], Lee et al. [12], Hwang et al. [13], Liao et al. [14], Liao and Sheen [15], Zhong et al. [16], Wan [17], Yin et al. [18], Yin et al. [19], and Yin et al. [20].

All of the above reviewing models assume that the duration of maintenance is fixed. But in our model, we assume that the duration is variable, which depends on the load of the machine before the maintenance. As pointed in the practical example, such assumption captures the real setting in steel-making process and our problem extends the previous models.

The paper is organized as follows. In Section 2, we formulate our problem as a integer programming model. In Section 3, we show that our problem is NP-hard and propose an FPTAS for it. In Section 4, we point out that a special case of our problem is polynomially solvable. Finally, we derive the fast heuristic algorithms and carry out the numerical experiment to evaluate their performance in Section 5. Some concluding remarks are given in Section 6.

2. Mathematical Model

In this section, the problem 1, LDM$\|C_{\max}$ is formulated as an integer programming model. The decision variable x_j is a 0-1 variable, $j = 1, 2, \ldots, n$. If $x_j = 1$, then job J_j is scheduled before the maintenance, $j = 1, 2, \ldots, n$. Otherwise, job J_j is scheduled after the maintenance, $j = 1, 2, \ldots, n$. The proposed model is presented as follows.

Integer programming model is

Objective function:

$$\text{Minimize} \quad C_{\max} \tag{1}$$

$$\text{Subject to:} \sum_{j=1}^{n} x_j p_j \le s \tag{2}$$

$$C_{\max} = s + f\left(\sum_{j=1}^{n} x_j p_j\right) + \sum_{j=1}^{n} \left(1 - x_j\right) p_j \tag{3}$$

$$x_j \in \{0, 1\}, \quad j = 1, 2, \ldots, n. \tag{4}$$

The objective function (1) minimizes the makespan, that is, the maximum completion time of the final job. Constraint (2) denotes the sum of processing times of jobs scheduled before the maintenance no larger than the starting time of the maintenance. Constraint (3) denotes the completion time of the final job. Constraint (4) indicates that the range of the decision variables is 0 or 1.

3. An FPTAS

In this section, we first state the computational complexity for our problem. Lee [5] showed that the scheduling problem with an unavailable fixed interval on a single machine to minimize the makespan is NP-hard, which can be viewed as a special case of our problem with the fixed-length duration. Therefore, our general problem 1, LDM$\|C_{\max}$ is also NP-hard. Accordingly, we have the following theorem.

Theorem 1. *The problem* 1, *LDM*$\|C_{\max}$ *is NP-hard.*

Next, we derive a fully polynomial time approximation scheme (FPTAS) for our problem based on the classical 0-1 Min-Knapsack Problem (Min-KP). For completeness, we describe Min-KP in advance here. Given a knapsack and a set of items, the size of the knapsack is given; associated with each item is a given size and a given profit. The problem is to select a subset of items into knapsack so as to minimize the total profit of all unselected items. Kellerer et al. [21] showed that there exists an FPTAS for solving Min-KP.

We now propose an FPTAS for problem 1, LDM$\|C_{\max}$ based on the FPTAS to Min-KP.

Algorithm H.

Step 1. Let $\hat{\varepsilon}$ be a given positive constant and $s_1 = \min_{j\in\{1,2,\ldots,n\}} p_j$. Compute a series of s_i such that $f(s_i) = f(s_1)(1+\hat{\varepsilon})^i$, $i = 2, \ldots, k-1$ with $k - 1 = \lfloor \log_{1+\hat{\varepsilon}}^{f(s)} - \log_{1+\hat{\varepsilon}}^{f(s_1)} \rfloor$. Let $s_0 = 0$, $s_k = s$.

Step 2. For each s_i, $i = 0, 1, 2, \ldots, k$, construct a corresponding Min-KP(i) as follows. Each job in an instance for problem 1, LDM$\|C_{\max}$ is corresponding to a item. The size and the profit of each item are the processing time of its corresponding job. The size of the knapsack is s_i.

Step 3. Invoke the FPTAS for each constructed Min-KP(i), $i = 0, 1, 2, \ldots, k$ [21]; then, we obtain the corresponding items that are selected into the knapsack. Denote them as a corresponding job subset $\mathcal{J}_b^i$. Construct a schedule $\pi(i)$, where the jobs in $\mathcal{J}_b^i$ are scheduled before the maintenance in an arbitrary order, and the remaining jobs $\mathcal{J} \setminus \mathcal{J}_b^i$ are scheduled after the maintenance in an arbitrary order. Let $C_{\max}(\pi(i))$ denote the corresponding objective value, $i = 0, 1, 2, \ldots, k$.

Step 4. From all the constructed schedule $\pi(i)$, $i = 0, 1, 2, \ldots, k$, we choose the schedule with the minimal objective value as the output. Denote it as π; that is,

$$C_{\max}(\pi) = \min_{i=0,1,2,\ldots,k} C_{\max}(\pi(i)). \tag{5}$$

Next, we show that Algorithm H is an FPTAS for problem 1, LDM$\|C_{\max}$.

Let π^* denote an optimal schedule and let l^* denote the sum of processing times of jobs scheduled before the maintenance in the optimal schedule π^*. Let $P = \sum_{j=1}^{n} p_j$. Then, the makespan of the optimal schedule π^* is equal to $s + f(l^*) + P - l^*$; that is, $C_{\max}(\pi^*) = s + f(l^*) + P - l^*$. We split the interval $[0, s]$ into a series of subintervals $[0, s_1], [s_1, s_2], \ldots, [s_{k-1}, s_k]$. Clearly, there exists a subinterval $[s_{i-1}, s_i]$ such that $s_{i-1} \leq l^* \leq s_i$, $i = 1, 2, \ldots, k$. Consider the constructed schedule $\pi(i)$; let l^i denote the sum of processing times of jobs scheduled before the maintenance in the schedule $\pi(i)$. Then, the makespan of the schedule $\pi(i)$ is equal to $s + f(l^i) + P - l^i$; that is, $C_{\max}(\pi(i)) = s + f(l^i) + P - l^i$. According to the choice of s_i, we know $f(l^i) \leq (1+\varepsilon) f(l^*)$. Furthermore, we have $P - l^i \leq (1+\varepsilon)(P - l^*)$, because the schedule $\pi(i)$ is obtained by an FPTAS for Min-KP(i) and π^* is corresponding to a feasible solution for Min-KP(i). In the end, we have $s + f(l^i) + P - l^i \leq (1+\varepsilon)(s + f(l^*) + P - l^*)$; that is, $C_{\max}(\pi(i)) \leq (1+\varepsilon) C_{\max}(\pi^*)$. Consequently, $C_{\max}(\pi) \leq C_{\max}(\pi(i)) \leq (1+\varepsilon) C_{\max}(\pi^*)$. For the computation for each s_i, since $f(\cdot)$ is nonnegative and nondecreasing and can be computed in polynomial time, by binary search it needs at most $\log_2^s$ time. Thus, in all it needs at most $\log_2^{f(s)} \log_2^s$ time. Invoking each Min-KP(i) needs polynomial time. Thus, Algorithm H needs polynomial time.

With the above argument, we have the following theorem.

Theorem 2. *There exists an FPTAS for problem* $1, LDM \| C_{\max}$.

4. A Special Case

Now, we consider a special case for our problem that the function $f(\cdot)$ is differential and $(df/dl)(\cdot) \geq 1$. From the expression on $C_{\max}$ in (3), we know

$$\begin{aligned} C_{\max} &= s + f\left(\sum_{j=1}^{n} x_j p_j\right) + \sum_{j=1}^{n} \left(1 - x_j\right) p_j \\ &= s + f\left(\sum_{j=1}^{n} x_j p_j\right) - \sum_{j=1}^{n} x_j p_j + P. \end{aligned} \tag{6}$$

Applying the well-known Mean-value Theorem [22], we have $C_{\max} = s + P + f(0) + ((df/dl)(\eta) - 1) \sum_{j=1}^{n} x_j p_j$, where $0 < \eta < \sum_{j=1}^{n} x_j p_j$. Because $(df/dl)(\cdot) \geq 1$ and $s + P + f(0)$ is constant, to minimize $C_{\max}$, we set all the x_j, $j = 1, 2, \ldots, n$, to 0, that is, $x_j = 0$, $j = 1, 2, \ldots, n$, which implies that we schedule all the jobs after the maintenance. The optimal makespan is equal to $s + f(0) + P$.

5. Heuristic Algorithms and Numerical Experiments

Although the integer programming model provides the optimal solution, Theorem 1 indicates that the existence of a polynomial time algorithm for solving our problem is impossible. So designing fast heuristic algorithm to obtain near-optimal solution is reasonable. In this section, we propose heuristic algorithms to find approximate solutions. Numerical experiments are implemented to evaluate the effectiveness of these proposed heuristic algorithms.

Heuristic 1 (H1).

Step 1. Order the jobs as $J_1, J_2, \ldots, J_n$ according to nondecreasing processing time; that is, $p_1 \leq p_2 \leq \cdots \leq p_n$.

Step 2. Compute the value k such that $\sum_{j=1}^{k} p_j \leq s \leq \sum_{j=1}^{k+1} p_j$.

Step 3. Construct a series of schedules $\pi_i, i = 1, 2, \ldots, k$, where the jobs scheduled before the maintenance in π_i are $J_1, J_2, \ldots, J_i$ and the remaining jobs are scheduled after the maintenance. Construct schedule π_0, where all the jobs are scheduled after the maintenance in π_0.

Step 4. From all the schedules π_0, π_i, $i = 1, 2, \ldots, k$, choose the best as the output.

Heuristic 2 (H2).

Step 1. Order the jobs as $J_1, J_2, \ldots, J_n$ according to nonincreasing processing time; that is, $p_1 \geq p_2 \geq \cdots \geq p_n$.

Step 2. Compute the value l such that $\sum_{j=1}^{l} p_j \leq s \leq \sum_{j=1}^{l+1} p_j$.

Step 3. Construct a series of schedules $\pi_i, i = 1, 2, \ldots, l$, where the jobs scheduled before the maintenance in π_i are $J_1, J_2, \ldots, J_i$ and the remaining jobs are scheduled after the maintenance. Construct schedule π_0, where all the jobs are scheduled after the maintenance in π_0.

Step 4. From all the schedules $\pi_0, \pi_i, i = 1, 2, \ldots, l$, choose the best as the output.

The heuristic algorithms H1 and H2 are coded in MATLAB language and executed on the computer with 1 G RAM and 512 KB L2 cache. For the experiments, we choose the following parameters to generate randomly the test problems:

> *number of jobs* (n): 50, 100, 200, 400;
>
> *the starting time of the maintenance (s)*: 10, 20, 40, 60;
>
> *processing times of jobs*: generated from the discrete uniform distribution with range $[1, 30]$;
>
> *the duration of the maintenance* ($d = f(l)$): $\lceil 2 + l/2 \rceil$, $\lceil 5 + l/3 \rceil$, $\lceil 10 + l/4 \rceil$, $\lceil 30 + l/8 \rceil$.

To evaluate the performance of the heuristic algorithms, we adopt the trivial lower bound $f(0) + P$. Based on the lower bound, define the relative error ratio as ER $= [(C_{\max}(\mathrm{H}i) - f(0) - P)/(f(0) + P)] \times 100\%$, where $C_{\max}(\mathrm{H}i)$ denotes the makespan of the schedule generated by the heuristic algorithm Hi, $i = 1$, or 2. The Average Error Ratio and the Maximum Error Ratio measured over the lower bound of the makespan are used for the performance evaluation.

For each parameter combination, 200 random problems are generated to evaluate the performance of the heuristics

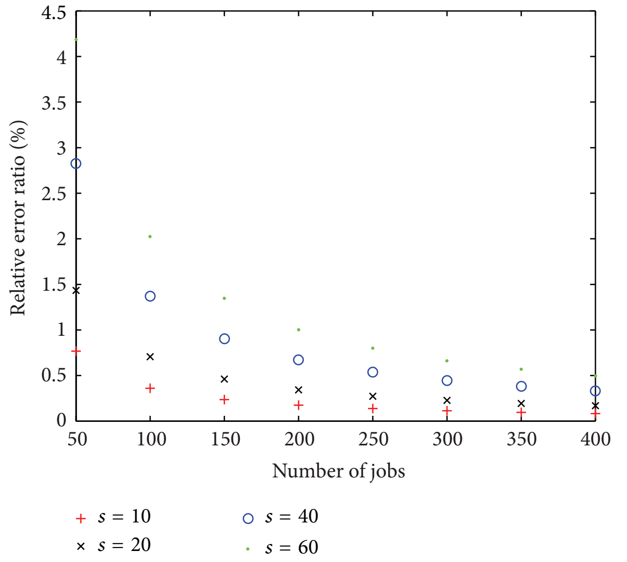

FIGURE 1: Average error ratios of algorithm H1 with $d = f(l) = 2 + l/2$.

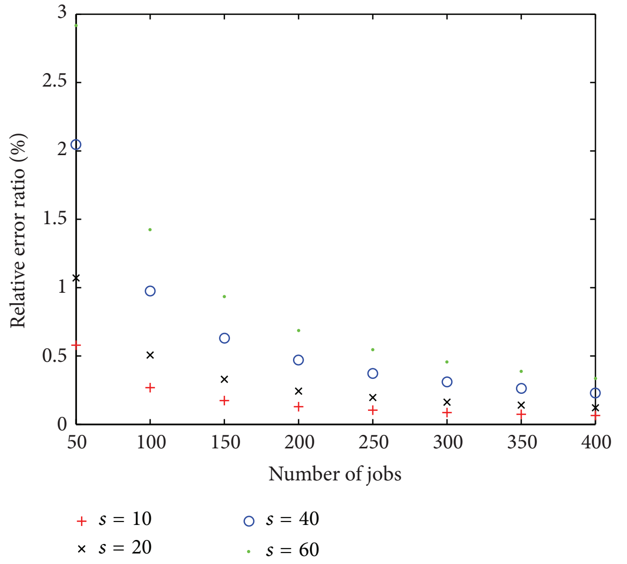

FIGURE 2: Average error ratios of algorithm H1 with $d = f(l) = 5 + l/3$.

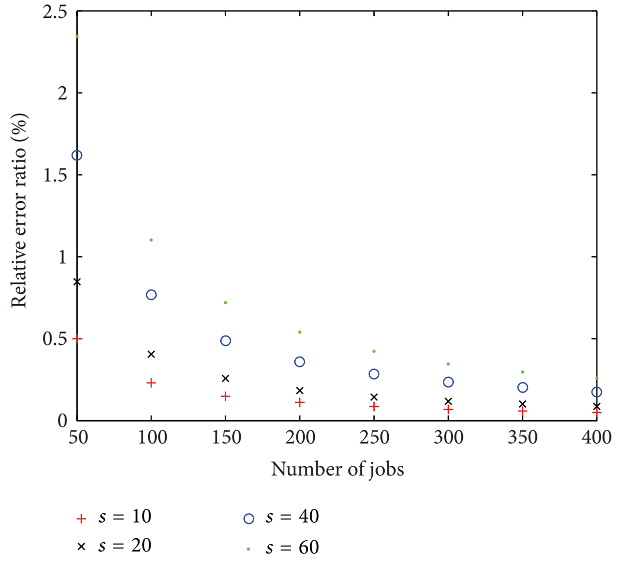

FIGURE 3: Average error ratios of algorithm H1 with $d = f(l) = 10 + l/4$.

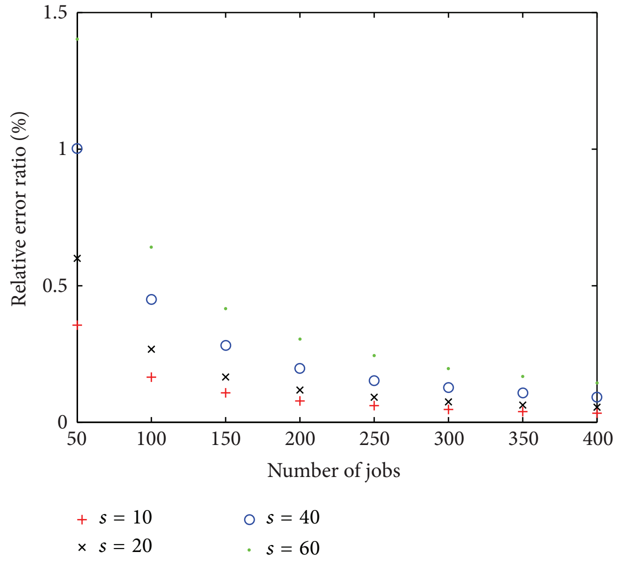

FIGURE 4: Average error ratios of algorithm H1 with $d = f(l) = 300 + l/8$.

algorithms. We do not report CPU times since even the largest problem is solved in a matter of seconds and there are no substantial differences in CPU times between the two algorithms. The evaluation results are showed in Tables 1 and 2.

The results of numerical experiments indicate that Heuristic 1 (H1) dominates Heuristic 2 (H2) in most cases and the performance of Heuristic 1 (H1) is effective in obtaining near-optimal solution. They also display that the error ratios reflect a downward trend when the value of n increases (see Figures 1, 2, 3, and 4). We also observe that the error ratios appear in an upward trend when the value of s increases; that is, the heuristic solutions get worse when the starting time of the maintenance becomes larger (see Figure 5). Furthermore, the heuristic algorithm H1 exhibits minor error ratios when the duration is relatively small (see Figure 6).

6. Concluding Remarks

In this paper, we investigate a scheduling problem on a single machine, in which maintenance on the machine must be performed. The starting time of the maintenance is given beforehand but its duration depends on the load of the machine before the maintenance. The goal is to minimize the

Table 1: Computational results of average error (%).

n	s	$[2 + (l/2)]$		$[5 + (l/3)]$		$[10 + (l/4)]$		$[30 + (l/8)]$	
		H1	H2	H1	H2	H1	H2	H1	H2
50	10	0.76	1.30	0.59	1.29	0.50	1.28	0.36	1.24
	20	1.45	2.59	1.08	2.61	0.86	2.57	0.60	2.50
	40	2.82	3.27	2.03	2.61	1.61	2.32	0.99	1.77
	60	4.17	3.96	2.94	2.69	2.30	2.05	1.40	1.10
100	10	0.36	0.65	0.27	0.65	0.23	0.64	0.16	0.63
	20	0.70	1.30	0.51	1.29	0.40	1.29	0.26	1.27
	40	1.38	1.62	0.98	1.29	0.76	1.16	0.45	0.90
	60	2.04	1.95	1.42	1.30	1.11	0.98	0.64	0.52
200	10	0.17	0.32	0.13	0.32	0.11	0.32	0.08	0.32
	20	0.34	0.65	0.25	0.64	0.19	0.64	0.12	0.64
	40	0.67	0.81	0.47	0.65	0.36	0.58	0.20	0.45
	60	1.00	0.97	0.69	0.64	0.53	0.48	0.31	0.26
400	10	0.08	0.16	0.06	0.16	0.05	0.16	0.03	0.16
	20	0.17	0.32	0.12	0.32	0.09	0.32	0.06	0.32
	40	0.33	0.40	0.23	0.32	0.17	0.29	0.09	0.22
	60	0.49	0.48	0.34	0.32	0.26	0.24	0.15	0.13

Table 2: Computational results of maximum error (%).

n	s	$[2 + (l/2)]$		$[5 + (l/3)]$		$[10 + (l/4)]$		$[30 + (l/8)]$	
		H1	H2	H1	H2	H1	H2	H1	H2
50	10	1.03	1.65	0.94	1.64	0.78	1.58	0.72	1.49
	20	1.83	3.17	1.60	3.46	1.24	3.27	1.11	3.04
	40	3.53	4.01	2.64	3.44	2.14	2.87	1.50	2.26
	60	5.08	5.20	3.85	3.65	2.98	2.81	2.06	1.74
100	10	0.45	0.76	0.36	0.79	0.31	0.73	0.24	0.72
	20	0.89	1.50	0.68	1.51	0.53	1.51	0.41	1.53
	40	1.61	1.89	1.21	1.60	1.04	1.34	0.65	1.16
	60	2.43	2.31	1.72	1.50	1.35	1.22	0.84	0.72
200	10	0.21	0.36	0.15	0.36	0.14	0.37	0.10	0.36
	20	0.39	0.73	0.28	0.72	0.24	0.70	0.18	0.71
	40	0.75	0.90	0.54	0.72	0.43	0.66	0.26	0.51
	60	1.15	1.14	0.78	0.73	0.62	0.55	0.38	0.29
400	10	0.11	0.17	0.07	0.17	0.07	0.0017	0.05	0.18
	20	0.19	0.36	0.14	0.35	0.10	0.35	0.07	0.35
	40	0.36	0.44	0.25	0.36	0.20	0.31	0.12	0.25
	60	0.53	0.53	0.38	0.35	0.29	0.26	0.18	0.14

makespan. We formulate it as an integer programming model and then propose an FPTAS and show that a special case of our problem is polynomially solvable. Finally, we propose fast heuristic algorithms and carry out numerical experiments to evaluate their performance.

Appendix

See Figures 1–6.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work was supported by the Natural Sciences Foundation (Grant no. 20142BAB211017) of Jiangxi Province and the

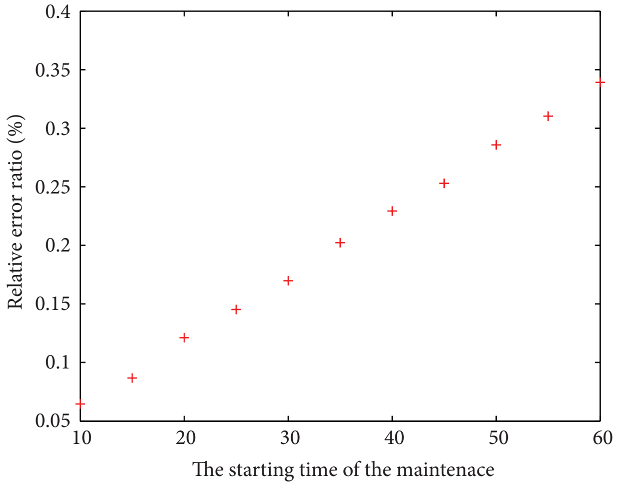

FIGURE 5: Average error ratios of algorithm H1 with $n = 400$ and $d = f(l) = 5 + l/3$.

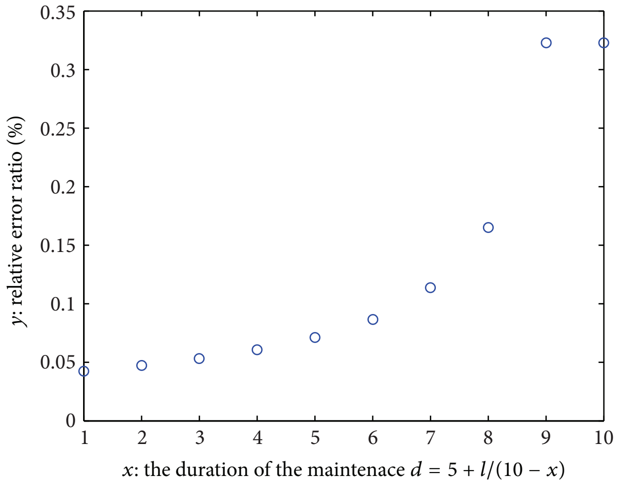

FIGURE 6: Average error ratios of algorithm H1 with $n = 400$ and $d = f(l) = 5 + l/(10 - x)$.

School Subject (Grant no. 06162015 and Grant no. 06102015) of Jiangxi University of Finance and Economics.

References

[1] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. R. Kan, "Optimization and approximation in deterministic sequencing and scheduling: a survey," *Annals of Discrete Mathematics*, vol. 5, pp. 287–326, 1979.

[2] L. Tang and Y. Zhang, "A new lagrangian relaxation algorithm for scheduling dissimilar parallel machines with release dates," *International Journal of Systems Science*, vol. 42, no. 7, pp. 1133–1141, 2011.

[3] G. Schmidt, "Scheduling with limited machine availability," *European Journal of Operational Research*, vol. 121, no. 1, pp. 1–15, 2000.

[4] Y. Ma, C. Chu, and C. Zuo, "A survey of scheduling with deterministic machine availability constraints," *Computers and Industrial Engineering*, vol. 58, no. 2, pp. 199–211, 2010.

[5] C.-Y. Lee, "Machine scheduling with an availability constraint," *Journal of Global Optimization*, vol. 9, no. 3-4, pp. 395–416, 1996.

[6] Y. He, M. Ji, and T. C. E. Cheng, "Single machine scheduling with a restricted rate-modifying activity," *Naval Research Logistics*, vol. 52, no. 4, pp. 361–369, 2005.

[7] J. Breit, G. Schmidt, and V. A. Strusevich, "Non-preemptive two-machine open shop scheduling with non-availability constraints," *Mathematical Methods of Operations Research*, vol. 57, no. 2, pp. 217–234, 2003.

[8] C.-C. Wu and W.-C. Lee, "Scheduling linear deteriorating jobs to minimize makespan with an availability constraint on a single machine," *Information Processing Letters*, vol. 87, no. 2, pp. 89–93, 2003.

[9] S. Gawiejnowicz, "Scheduling deteriorating jobs subject to job or machine availability constraints," *European Journal of Operational Research*, vol. 180, no. 1, pp. 472–478, 2007.

[10] M. Ji, Y. He, and T. C. E. Cheng, "Scheduling linear deteriorating jobs with an availability constraint on a single machine," *Theoretical Computer Science*, vol. 362, no. 1–3, pp. 115–126, 2006.

[11] C.-Y. Lee, "Parallel machines scheduling with non-simultaneous machine available time," *Discrete Applied Mathematics*, vol. 30, no. 1, pp. 53–61, 1991.

[12] C.-Y. Lee, Y. He, and G. Tang, "A note on 'parallel machine scheduling with non-simultaneous machine available time,'" *Discrete Applied Mathematics*, vol. 100, no. 1-2, pp. 133–135, 2000.

[13] H.-C. Hwang, K. Lee, and S. Y. Chang, "The effect of machine availability on the worst—case performance of LPT," *Discrete Applied Mathematics*, vol. 148, no. 1, pp. 49–61, 2005.

[14] C.-J. Liao, D.-L. Shyur, and C.-H. Lin, "Makespan minimization for two parallel machines with an availability constraint," *European Journal of Operational Research*, vol. 160, no. 2, pp. 445–456, 2005.

[15] L.-W. Liao and G.-J. Sheen, "Parallel machine scheduling with machine availability and eligibility constraints," *European Journal of Operational Research*, vol. 184, no. 2, pp. 458–467, 2008.

[16] X. Zhong, J. Ou, and G. Wang, "Order acceptance and scheduling with machine availability constraints," *European Journal of Operational Research*, vol. 232, no. 3, pp. 435–441, 2014.

[17] L. Wan, "Scheduling jobs and a variable maintenance on a single machine with common due-date assignment," *The Scientific World Journal*, vol. 2014, Article ID 748905, 5 pages, 2014.

[18] Y. Yin, T. C. E. Cheng, and C.-C. Wu, "Scheduling with time-dependent processing times," *Mathematical Problems in Engineering*, vol. 2014, Article ID 201421, 2 pages, 2014.

[19] Y. Yin, D. Ye, and G. Zhang, "Single machine batch scheduling to minimize the sum of total flow time and batch delivery cost with an unavailability interval," *Information Sciences*, vol. 274, pp. 310–322, 2014.

[20] Y. Q. Yin, W. H. Wu, T. C. E. Cheng, and C. C. Wu, "Single-machine scheduling with time-dependent and position-dependent deteriorating jobs," *International Journal of Computer Integrated Manufacturing*, 2014.

[21] H. Kellerer, U. Pferschy, and D. Pisinger, *Knapsack Problem*, Springer, Berlin, Germany, 2004.

[22] G. B. Folland, *Advanced Calculus*, Prentice Hall, Englewood Cliffs, NJ, USA, 2002.

Advances in Operations Research
Advances in Decision Sciences
Journal of Applied Mathematics
Algebra
Journal of Probability and Statistics
The Scientific World Journal
International Journal of Differential Equations
Hindawi
International Journal of Combinatorics
Advances in Mathematical Physics
Journal of Complex Analysis
Journal of Mathematics
Mathematical Problems in Engineering
Abstract and Applied Analysis
Discrete Dynamics in Nature and Society
International Journal of Mathematics and Mathematical Sciences
Journal of Discrete Mathematics
Journal of Function Spaces
International Journal of Stochastic Analysis
Journal of Optimization