Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 805261, 5 pages
http://dx.doi.org/10.1155/2013/805261

Research Article

Two-Agent Single-Machine Scheduling with Resource-Dependent Starting Times

Peng Liu and Xiaoyu Tian

School of Management, Shenyang University of Technology, Shenyang 110870, China

Correspondence should be addressed to Peng Liu; liup7802@163.com

Received 22 May 2013; Revised 15 July 2013; Accepted 15 July 2013

Academic Editor: Yunqiang Yin

We consider several two-agent scheduling problems with resource consumption on a single machine, where each of the agents wants to minimize a measure dependent on its own jobs. The starting time of each job of the first agent is related to the amount of resource consumed. The objective is to minimize the total amount of resource consumption of the first agent with the restriction that the makespan or the total completion time of the second agent cannot exceed a given bound U. The optimal properties and the optimal polynomial time algorithms are proposed to solve the scheduling problems.

1. Introduction

Recently, there has been a growing interest in the literature to study multiagent scheduling problems, in which different agents share a common processing machine, and each agent wants to minimize a cost function dependent on its own job. Scheduling with multiple agents is firstly introduced by Baker and Smith [1] and Agnetis et al. [2]. Baker and Smith [1] manage to find a schedule that minimizes a combination of multiple agents' objective functions. Agnetis et al. [2] study several scheduling problems with two agents that have requirements of their own objective functions. Some researchers pay their attention to the different machine environments and job characteristics of the scheduling problems with multiagent. Yuan et al. [3] show that two dynamic programming recursions in Baker and Smith [1] are incorrect and give a polynomial time algorithm for the same problems. Cheng et al. [4] extend one of the problems of Agnetis et al. [2] to the multiagent scheduling problem with minimizing the total weighted number of tardy jobs on a single machine. They show that the problem is strongly NP-complete in general, and they study a special case in which the number of agents is fixed. Cheng et al. [5] consider the multiagent scheduling problem on a single machine, where the agent's objective functions are of the max form. They study the feasibility model and the minimality model. Leung et al. [6] study a two-agent scheduling environment with m identical parallel machines and generalize the results of Baker and Smith [1] and Agnetis et al. [2] by including the total tardiness objective, allowing for preemptions, and considering jobs with different release dates. Gawiejnowicz et al. [7] consider a single-machine two-agent scheduling problem with proportionally deteriorating job processing times. Yin et al. [8] consider several two-agent scheduling problems with assignable due dates on a single machine. Gerstl and Mosheiov [9] consider the scheduling problems with two competing agents to minimize the weighted earliness-tardiness. Cheng et al. [10] study a two-agent single-machine scheduling problem with release times to minimize the total weighted completion time. Zhao and Lu [11] consider two models of two-agent scheduling problems on identical machines. Yin et al. [12] provide a branch-and-bound procedure for a single-machine earliness scheduling problem with two agents. Yu et al. [13] investigate several single-machine two-synergetic-agent scheduling problems. For other recent scheduling problems with two agents, the reader is referred to Lee et al. [14], Wu et al. [15], Feng et al. [16], and Wu et al. [17].

On the other hand, the scheduling problems with resource consumption have received a considerable attention in the last two decades. The classical scheduling problems have traditionally been considered under the assumptions that job processing times and release times are constant parameters and that the resource allocation issue is ignored. Cheng and Janiak [18] study the resource optimal control

of job completion on a single machine with a constraint on maximum job completion time. Li [19] considers the problem of scheduling a set of jobs on a single machine where the release time of a job is related to the amount of resources consumed. Vasilev and Foote [20] investigate a single-machine scheduling problem where the resources consumed depend on the release times of jobs. Kaspi and Shabtay [21] consider the problem of scheduling jobs on a single machine where job processing times are controllable through the allocation of a common limited resource. Wang and Cheng [22] consider the single-machine scheduling problem with resource-dependent release times and processing times. Shabtay and Kaspi [23] study the parallel machine scheduling problems with a convex resource consumption function. Yedidsion et al. [24] provide an optimization algorithm for the single-machine problem where the processing time of each job is resource-dependent. Wang et al. [25] study the single-machine scheduling problem with learning effect and resource-dependent processing times. Wei et al. [26] consider the single-machine scheduling with time and resource-dependent processing times. J. B. Wang and M. Z. Wang [27] consider the single-machine scheduling to minimize total convex resource consumption with a constraint on total weighted flow time. Wang et al. [28] consider the scheduling problem with processing times dependent on position, starting time, and allotted resources. Wang et al. [29] consider single-machine scheduling problems with deteriorating jobs and resource allocation in a group technology environment. Hsu and Yang [30] analyze unrelated parallel machine scheduling resource allocation problems with position-dependent deteriorating jobs.

However, to the best of our knowledge, the multiagent scheduling problem with the consideration of resource-dependent starting times has hardly been studied in the literature. These two categories of scheduling problems have been extensively and separately researched over the last two decades. In this paper, we study the two-agent scheduling problems on a single machine with resource-dependent starting times, where the goal is to find a schedule that minimizes the objective function of one agent with the restriction that the objective function of the other agent cannot exceed a given bound. The problems under consideration fall into the category of scheduling problems with resource consumption and multiple agents.

The remainder of this paper is organized as follows. In Section 2, we describe the proposed problems. In Sections 3 and 4, we develop the optimal polynomial time algorithms for the two-agent single-machine scheduling problems. Section 5 gives some concluding remarks.

2. Problem Description

We now describe our problems formally. There are two families of independent and nonpreemptive jobs $J^A = \{J_1^A, J_2^A, \ldots, J_{n_A}^A\}$ and $J^B = \{J_1^B, J_2^B, \ldots, J_{n_B}^B\}$ to be processed on a common single machine. All jobs are available for processing at time zero. The jobs in J^A and J^B are called A-agent's jobs and B-agent's jobs, respectively. Associated with each job J_h^A, let p_h^A denote the processing time, $h = 1, 2, \ldots, n_A$. The starting time s_h^A is related to the amount of the resource f_h^A, consumed on job J_h^A. We assume that $f_h^A = f(s_h^A)$, where $f : R^+ \to R^+$, called the resource consumption function, is a strictly increasing continuous function to A-agent's jobs. Associated with each job J_k^B, let p_k^B denote the processing time, $k = 1, 2, \ldots, n_B$. Let π indicate a feasible schedule of the $n = n_A + n_B$ jobs. Let $C_k^B(\pi)$ denote the completion time of B-agent's job J_k^B under schedule π. The objective function of agent A is to minimize the total amount of resource consumption $\sum_{h=1}^{n_A} f(s_h^A)$. The objective function of agent B is to minimize the makespan $C_{\max}^B = \max_{k=1,2,\ldots,n_B}\{C_k^B(\pi)\}$ or the total completion time $\sum_{k=1}^{n_B} C_k^B(\pi)$.

The goal is to minimize the total amount of resource consumption $\sum_{h=1}^{n_A} f(s_h^A)$ of agent A with the restriction that the makespan $C_{\max}^B$ or the total completion time $\sum_{k=1}^{n_B} C_k^B$ of agent B cannot exceed a given bound U. If the value U is too small, an instance of the scheduling problem may not have feasible solutions. If there is at least one feasible solution, we say that the instance is *feasible*. According to the three-field notation $\psi_1|\psi_2|\psi_3$ of Graham et al. [31], the two scheduling problems are denoted as $1\|\sum_{h=1}^{n_A} f(s_h^A) : C_{\max}^B \le U$ and $1\|\sum_{h=1}^{n_A} f(s_h^A) : \sum_{k=1}^{n_B} C_k^B \le U$, respectively.

3. Problem $1\|\sum_{h=1}^{n_A} f(s_h^A) : C_{\max}^B \le U$

In this section, we develop an optimal polynomial time algorithm to solve the problem $1\|\sum_{h=1}^{n_A} f(s_h^A) : C_{\max}^B \le U$. The following Lemmas 1 and 2 are easily obtained, and we omit the details of the proof.

Lemma 1. *Given a sequence $\pi = \{J_1^B, J_2^B, \ldots, J_{n_B}^B, J_1^A, J_2^A, \ldots, J_{n_A}^A\}$ and a constant U, define $C_\pi^B = \sum_{k=1}^{n_B} p_k^B$. Then, if $U < C_\pi^B$, the sequence π corresponds to an infeasible schedule.*

Lemma 2. *Given a sequence $\pi = \{J_1^A, J_2^A, \ldots, J_{n_A}^A, J_1^B, J_2^B, \ldots, J_{n_B}^B\}$ and a constant U, define $C_\pi^B = \sum_{h=1}^{n_A} p_h^A + \sum_{k=1}^{n_B} p_k^B$. Then, if $U \ge C_\pi^B$, the schedule corresponding to the sequence π is feasible.*

Now we can define bounds for the constraint U. Define $\underline{U} = \sum_{k=1}^{n_B} p_k^B$ and $\overline{U} = \sum_{h=1}^{n_A} p_h^A + \sum_{k=1}^{n_B} p_k^B$.

Lemma 3. *If $U < \underline{U} = \sum_{k=1}^{n_B} p_k^B$, there are no feasible schedules. If $U \ge \overline{U} = \sum_{h=1}^{n_A} p_h^A + \sum_{k=1}^{n_B} p_k^B$, there are feasible schedules for all possible sequences.*

From now on, we will always assume that $\underline{U} \le U \le \overline{U}$ in this problem.

Lemma 4. *An optimal schedule exists in which the A-agent's jobs are processed in the nondecreasing order of processing times p_h^A.*

Proof. The resource consumption function f is a strictly increasing continuous function to A-agent's jobs. Since

starting A-agent's jobs later consumes more resources, A-agent's jobs should be processed as early as possible. Hence, A-agent's jobs should be processed in a nondecreasing order of p_h^A. □

Lemma 5. *An optimal schedule exists in which the B-agent's jobs are consecutively processed.*

Proof. The makespan of agent B is the maximum completion time of B-agent's jobs on the single machine; that is, the makespan of agent B is the completion time of B-agent's last job. Using a pairwise job interchange argument, we can consecutively process B-agent's jobs and consolidate all B-agent's jobs into a block. □

Next, an algorithm to determine an optimal schedule of the problem $1\|\sum_{h=1}^{n_A} f(s_h^A) : C_{\max}^B \le U$ is developed as follows.

Algorithm 6.

Step 1. Set $i \leftarrow 1$.

Step 2. Arrange the A-agent's jobs as $\{J_1^A, J_2^A, \ldots, J_{n_A}^A\}$ according to the nondecreasing order of p_h^A and denote all B-agent's jobs as a dummy job $B1$.

Step 3. Define sequence $S = \{J_1^A, J_2^A, \ldots, J_{n_A-1}^A, J_{n_A}^A, B1\}$ and calculate the makespan $C_{\max}^B$ for agent B. If $C_{\max}^B \le U$, then the sequence S is an optimal schedule.

Step 4. Select the dummy job $B1$ to the $(n_A - i)$th position in the sequence $\{J_1^A, J_2^A, \ldots, J_{n_A-i}^A, J_{n_A-i+1}^A, \ldots, J_{n_A}^A\}$ and calculate the makespan $C_{\max}^B$ for agent B. If $C_{\max}^B \le U$, then the sequence $S_i = \{J_1^A, J_2^A, \ldots, J_{n_A-i}^A, B1, J_{n_A-i+1}^A, \ldots, J_{n_A-1}^A, J_{n_A}^A\}$ is an optimal schedule, and stop. Otherwise, go to Step 5.

Step 5. If $i < n_A$ and $C_{\max}^B > U$, then $i \leftarrow i + 1$, and go to Step 4. Otherwise, stop.

Theorem 7. *Algorithm 6 generates an optimal schedule for the problem $1\|\sum_{h=1}^{n_A} f(s_h^A) : C_{\max}^B \le U$ in $O(n_A \log n_A + n_B)$ time.*

Proof. The proof of optimality is straightforward from the results of Lemmas 1–5. We now turn to time complexity. The time to sequence the jobs of set J^A according to the nondecreasing order of p_h^A is $O(n_A \log n_A)$. Creating dummy job $B1$ incurs $O(n_B)$ operations. The solution requires finding locations for the dummy job $B1$ among the A-agent's jobs sequenced by the nondecreasing order of p_h^A. So, the overall computational complexity of Algorithm 6 is bounded by $O(n_A \log n_A + n_B)$. This completes the proof. □

4. Problem $1\|\sum_{h=1}^{n_A} f(s_h^A) : \sum_{k=1}^{n_B} C_k^B \le U$

For the problem $1\|\sum_{h=1}^{n_A} f(s_h^A) : \sum_{k=1}^{n_B} C_k^B \le U$, we have the following lemmas that are similar to Lemmas 1, 2, and 3.

Lemma 8. *Given a sequence $\pi = \{J_1^B, J_2^B, \ldots, J_{n_B}^B, J_1^A, J_2^A, \ldots, J_{n_A}^A\}$ and a constant U, define $C_\pi^B = \sum_{k=1}^{n_B}(n_B - k + 1)p_k^B$. Then, if $U < C_\pi^B$, the sequence π corresponds to an infeasible schedule.*

Proof. If $U < C_\pi^B$, then

$$\begin{aligned}\sum_{k=1}^{n_B} C_k^B &= \sum_{k=1}^{n_B}\left(s_k^B + p_k^B\right)\\ &= \sum_{k=1}^{n_B}\left(s_1^B + \sum_{i=1}^{k-1} p_i^B + p_k^B\right)\\ &= \sum_{k=1}^{n_B}\left(s_1^B + \sum_{i=1}^{k} p_i^B\right) = \sum_{k=1}^{n_B} s_1^B + \sum_{k=1}^{n_B}\sum_{i=1}^{k} p_k^B\\ &= n_B s_1^B + \sum_{k=1}^{n_B}(n_B - k + 1)\, p_k^B = C_\pi^B > U,\end{aligned} \tag{1}$$

which implies that the sequence π is infeasible. □

Lemma 9. *Given a sequence $\pi = \{J_1^A, J_2^A, \ldots, J_{n_A}^A, J_1^B, J_2^B, \ldots, J_{n_B}^B\}$ and a constant U, define $C_\pi^B = n_B \sum_{h=1}^{n_A} p_h^A + \sum_{k=1}^{n_B}(n_B - k + 1)p_k^B$. Then, if $U \ge C_\pi^B$, the schedule corresponding to the sequence π is feasible.*

Proof. If $U \ge C_\pi^B$, then

$$\begin{aligned}\sum_{k=1}^{n_B} C_k^B &= \sum_{k=1}^{n_B}\left(s_k^B + p_k^B\right)\\ &= \sum_{k=1}^{n_B}\left(s_1^B + \sum_{i=1}^{k-1} p_i^B + p_k^B\right)\\ &= \sum_{k=1}^{n_B}\left(s_1^B + \sum_{i=1}^{k} p_i^B\right)\\ &= \sum_{k=1}^{n_B} s_1^B + \sum_{k=1}^{n_B}\sum_{i=1}^{k} p_k^B\\ &= n_B \sum_{h=1}^{n_A} p_h^A + \sum_{k=1}^{n_B}(n_B - k + 1)\, p_k^B = C_\pi^B \le U,\end{aligned} \tag{2}$$

which implies that the sequence π is feasible. □

Now we can define bounds for the constraint U. Let the processing times of the B-agent's jobs be ordered in the nondecreasing order $p_1^B \le p_2^B \le \cdots \le p_{n_B}^B$. Then, define $\underline{U} = \sum_{k=1}^{n_B}(n_B - k + 1)p_k^B$ and $\overline{U} = n_B \sum_{h=1}^{n_A} p_h^A + \sum_{k=1}^{n_B} k p_k^B$. These constants play an important role in this problem.

Lemma 10. *If $U < \underline{U} = \sum_{k=1}^{n_B}(n_B - k + 1)p_k^B$, there are no feasible schedules. If $U \ge \overline{U} = n_B \sum_{h=1}^{n_A} p_h^A + \sum_{k=1}^{n_B} k p_k^B$, there are feasible schedules for all possible sequences.*

In view of these properties, the analysis in the following section will be confined to the case in which $\underline{U} \le U \le \overline{U}$.

Lemma 11. *An optimal schedule exists in which the A-agent's jobs are processed in the nondecreasing order of processing times* p_h^A.

Proof. The proof is same to Lemma 4, and we omit the details. □

Lemma 12. *An optimal schedule exists in which the B-agent's jobs are consecutively processed in the nondecreasing order of processing times* p_k^B.

Proof. Using a pairwise job interchange argument, we can consecutively process B-agent's jobs in the nondecreasing order of processing times p_k^B. □

Next, an algorithm to determine an optimal schedule of the problem $1\|\sum_{h=1}^{n_A} f(s_h^A) : \sum_{k=1}^{n_B} C_k^B \leq U$ is developed as follows.

Algorithm 13.

Step 1. Set $i \leftarrow 1$.

Step 2. Arrange the A-agent's jobs as $\{J_1^A, J_2^A, \ldots, J_{n_A}^A\}$ according to the nondecreasing order of p_h^A, and denote all B-agent's jobs sequenced by the nondecreasing order of p_k^B as a dummy job $B1$.

Step 3. Define sequence $S = \{J_1^A, J_2^A, \ldots, J_{n_A-1}^A, J_{n_A}^A, B1\}$ and calculate the total completion time $\sum_{k=1}^{n_B} C_k^B$ for agent B. If $\sum_{k=1}^{n_B} C_k^B \leq U$, then the sequence S is an optimal schedule.

Step 4. Select the dummy job $B1$ to the $(n_A - i)$th position in the sequence $\{J_1^A, J_2^A, \ldots, J_{n_A-i}^A, J_{n_A-i+1}^A, \ldots, J_{n_A}^A\}$ and calculate $\sum_{k=1}^{n_B} C_k^B$ for agent B. If $\sum_{k=1}^{n_B} C_k^B \leq U$, then the sequence $S_i = \{J_1^A, J_2^A, \ldots, J_{n_A-i}^A, B1, J_{n_A-i+1}^A, \ldots, J_{n_A-1}^A, J_{n_A}^A\}$ is an optimal schedule, and stop. Otherwise, go to Step 5.

Step 5. If $i < n_A$ and $\sum_{k=1}^{n_B} C_k^B > U$, then $i \leftarrow i + 1$, and go to Step 4. Otherwise, stop.

Theorem 14. *Algorithm 13 generates an optimal schedule for the problem* $1\|\sum_{h=1}^{n_A} f(s_h^A) : \sum_{k=1}^{n_B} C_k^B \leq U$ *in* $O(n_A \log n_A + n_B \log n_B)$ *time.*

Proof. The proof of optimality is straightforward from the results of Lemmas 8–12. We now turn to time complexity. The times to sequence the jobs of sets J^A and J^B according to the nondecreasing orders of p_h^A and p_k^B are $O(n_A \log n_A)$ and $O(n_B \log n_B)$, respectively. Creating dummy job $B1$ incurs $O(n_B)$ operations. So, the overall computational complexity of Algorithm 13 is bounded by $O(n_A \log n_A + n_B \log n_B)$. This completes the proof. □

5. Conclusions

In this paper, we introduce a new scheduling model in which both resource-dependent starting times and two agents exist simultaneously. The objective is to minimize the total amount of resource consumption of the first agent with the restriction that the makespan or the total completion time of the second agent does not exceed a given bound. We propose the optimal properties and the optimal polynomial time algorithms for the considered scheduling problems.

Future research may be directed to analyze the problems with other objective functions such as minimizing the number of late jobs, the total weighted completion time, and tardiness. An interesting research topic is also to consider other resource consumption functions. Finally, the future research direction is to analyze the scheduling problem with more than two agents or in other machine environments.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (Grant no. 71001074), the Program for Liaoning Excellent Talents in University (Grant no. WJQ2013003), and the Science Research Foundation, Department of Education, Liaoning Province of China (Grant no. W2010302).

References

[1] K. R. Baker and J. C. Smith, "A multiple-criterion model for machine scheduling," *Journal of Scheduling*, vol. 6, no. 1, pp. 7–16, 2003.

[2] A. Agnetis, P. B. Mirchandani, D. Pacciarelli, and A. Pacifici, "Scheduling problems with two competing agents," *Operations Research*, vol. 52, no. 2, pp. 229–242, 2004.

[3] J. J. Yuan, W. P. Shang, and Q. Feng, "A note on the scheduling with two families of jobs," *Journal of Scheduling*, vol. 8, no. 6, pp. 537–542, 2005.

[4] T. C. E. Cheng, C. T. Ng, and J. J. Yuan, "Multi-agent scheduling on a single machine to minimize total weighted number of tardy jobs," *Theoretical Computer Science*, vol. 362, no. 1–3, pp. 273–281, 2006.

[5] T. C. E. Cheng, C. T. Ng, and J. J. Yuan, "Multi-agent scheduling on a single machine with max-form criteria," *European Journal of Operational Research*, vol. 188, no. 2, pp. 603–609, 2008.

[6] J. Y. T. Leung, M. Pinedo, and G. Wan, "Competitive two-agent scheduling and its applications," *Operations Research*, vol. 58, no. 2, pp. 458–469, 2010.

[7] S. Gawiejnowicz, W. C. Lee, C. L. Lin, and C. C. Wu, "Single-machine scheduling of proportionally deteriorating jobs by two agents," *Journal of the Operational Research Society*, vol. 62, no. 11, pp. 1983–1991, 2011.

[8] Y. Yin, S. R. Cheng, T. C. E. Cheng, C. C. Wu, and W. H. Wu, "Two-agent single-machine scheduling with assignable due dates," *Applied Mathematics and Computation*, vol. 219, no. 4, pp. 1674–1685, 2012.

[9] E. Gerstl and G. Mosheiov, "Scheduling problems with two competing agents to minimized weighted earliness-tardiness," *Computers & Operations Research*, vol. 40, no. 1, pp. 109–116, 2013.

[10] T. C. E. Cheng, Y. H. Chung, S. C. Liao, and W. C. Lee, "Two-agent singe-machine scheduling with release times to minimize the total weighted completion time," *Computers & Operations Research*, vol. 40, no. 1, pp. 353–361, 2013.

[11] K. Zhao and X. Lu, "Approximation schemes for two-agent scheduling on parallel machines," *Theoretical Computer Science*, vol. 468, pp. 114–121, 2013.

[12] Y. Q. Yin, C. C. Wu, W. H. Wu, C. J. Hsu, and W. H. Wu, "A branch-and-bound procedure for a single-machine earliness scheduling problem with two agents," *Applied Soft Computing*, vol. 13, pp. 1042–1054, 2013.

[13] X. Yu, Y. Zhang, D. Xu, and Y. Yin, "Single machine scheduling problem with two synergetic agents and piece-rate maintenance," *Applied Mathematical Modelling*, vol. 37, no. 3, pp. 1390–1399, 2013.

[14] W. C. Lee, Y. H. Chung, and Z. R. Huang, "A single-machine bi-criterion scheduling problem with two agents," *Applied Mathematics and Computation*, vol. 219, no. 23, pp. 10831–10841, 2013.

[15] W. H. Wu, J. Y. Xu, W. H. Wu, Y. Q. Yin, I. F. Cheng, and C. C. Wu, "A tabu method for a two-agent single-machine scheduling with deterioration jobs," *Computers & Operations Research*, vol. 40, no. 8, pp. 2116–2127, 2013.

[16] Q. Feng, J. J. Yuan, H. L. Liu, and C. He, "A note on two-agent scheduling on an unbounded parallel-batching machine with makespan and maximum lateness objectives," *Applied Mathematical Modelling*, vol. 37, no. 10-11, pp. 7071–7076, 2013.

[17] C. C. Wu, W. H. Wu, J. C. Chen, Y. Q. Yin, and W. H. Wu, "A study of the single-machine two-agent scheduling problem with release times," *Applied Soft Computing*, vol. 13, no. 2, pp. 998–1006, 2013.

[18] T. C. E. Cheng and A. Janiak, "Resource optimal control in some single-machine scheduling problems," *IEEE Transactions on Automatic Control*, vol. 39, no. 6, pp. 1243–1246, 1994.

[19] C. L. Li, "Scheduling to minimize the total resource consumption with a constraint on the sum of completion times," *European Journal of Operational Research*, vol. 80, no. 2, pp. 381–388, 1995.

[20] S. H. Vasilev and B. L. Foote, "On minimizing resource consumption with constraints on the makespan and the total completion time," *European Journal of Operational Research*, vol. 96, no. 3, pp. 612–621, 1997.

[21] M. Kaspi and D. Shabtay, "Convex resource allocation for minimizing the makespan in a single machine with job release dates," *Computers & Operations Research*, vol. 31, no. 9, pp. 1481–1489, 2004.

[22] X. Wang and T. C. E. Cheng, "Single machine scheduling with resource dependent release times and processing times," *European Journal of Operational Research*, vol. 162, no. 3, pp. 727–739, 2005.

[23] D. Shabtay and M. Kaspi, "Parallel machine scheduling with a convex resource consumption function," *European Journal of Operational Research*, vol. 173, no. 1, pp. 92–107, 2006.

[24] L. Yedidsion, D. Shabtay, and M. Kaspi, "A bicriteria approach to minimize maximal lateness and resource consumption for scheduling a single machine," *Journal of Scheduling*, vol. 10, no. 6, pp. 341–352, 2007.

[25] D. Wang, M. Z. Wang, and J. B. Wang, "Single-machine scheduling with learning effect and resource-dependent processing times," *Computers and Industrial Engineering*, vol. 59, no. 3, pp. 458–462, 2010.

[26] C. M. Wei, J. B. Wang, and P. Ji, "Single-machine scheduling with time-and-resource-dependent processing times," *Applied Mathematical Modelling*, vol. 36, no. 2, pp. 792–798, 2012.

[27] J. B. Wang and M. Z. Wang, "Single-machine scheduling to minimize total convex resource consumption with a constraint on total weighted flow time," *Computers & Operations Research*, vol. 39, no. 3, pp. 492–497, 2012.

[28] J. B. Wang, M. Z. Wang, and P. Ji, "Scheduling jobs with processing times dependent on position, starting time, and allotted resource," *Asia-Pacific Journal of Operational Research*, vol. 29, no. 5, Article ID 1250030, 15 pages, 2012.

[29] D. Wang, Y. Z. Huo, and P. Ji, "Single-machine group scheduling with deteriorating jobs and allotted resource," *Optimization Letters*, 2013.

[30] C. J. Hsu and D. L. Yang, "Unrelated parallel-machine scheduling with position-dependent deteriorating jobs and resource-dependent processing time," *Optimization Letters*, 2013.

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