

## Research Article

# Scheduling Projects with Multiskill Learning Effect

**Hong Zha and Lianying Zhang**

*College of Management and Economics, Tianjin University, No. 92 Weijin Road, Nankai District, Tianjin 300072, China*

Correspondence should be addressed to Lianying Zhang; [tjzly126@126.com](mailto:tjzly126@126.com)

Received 3 November 2013; Accepted 24 December 2013; Published 10 February 2014

Academic Editors: Q. Cheng and J. Yang

Copyright © 2014 H. Zha and L. Zhang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We investigate the project scheduling problem with multiskill learning effect. A new model is proposed to deal with the problem, where both autonomous and induced learning are considered. In order to obtain the optimal solution, a genetic algorithm with specific encoding and decoding schemes is introduced. A numerical example is used to illustrate the proposed model. The computational results show that the learning effect cannot be neglected in project scheduling. By means of determining the level of induced learning, the project manager can balance the project makespan with total cost.

## 1. Introduction

Project scheduling problem is a well-known research field in project management [1], while project scheduling with multiskilled resources is an important extension of it. In the extended problem, activities in the projects are assumed to require multiskills simultaneously to be executed. On the other hand, the resources involved (e.g., human resources) are assumed to possess multiskills and perform one of them at a time. As the assumption of it is closer to the real-world situation, the project scheduling problem with multiskills has attracted much attention in recent years [2–5].

Learning effect might be another nonignorable factor in actual scheduling [6, 7], due to the enormous effect on efficiency of operations. Anzanello and Fogliatto [8] were the pioneer introducing the learning effect into the machine scheduling problem. In addition, Biskup [7] classified the learning effect into two different approaches: the position-based learning and sum-of-processing-time based learning. The position-based learning was considered to be a realistic assumption for machine learning, while the sum-of-processing-time was regarded as a suitable hypothesis for human learning. Most of the researches were carried out based on above-mentioned assumptions. Cheng and Wang [9] simplified linear-log model into linear one in machine scheduling with position-based learning effect. Biskup and Simons [10] proposed the concept of induced learning and investigated to find out an optimal level of induced learning

in machine scheduling. Wang [11] considered the machine scheduling problem affected by position-based learning as well as deteriorating jobs. Kuo and Yang [12] developed the model for single processor (e.g., human worker) scheduling problem with sum-of-processing-time based learning. Rudek [13] analyzed the single processor total weighted completion time scheduling problem with time-dependent learning effect and constructed a heuristic algorithm to solve it. Moreover, a few researchers discussed the scheduling problem where the two learning approaches coexisted in some situation. Wu and Lee [14] as well as Lu et al. [15] developed different models for the scheduling problem with both learning approaches.

Although the machine (processor) scheduling problems with learning effect have been fully investigated, the studies on project scheduling problems with learning effect have just started. Wu and Sun [16] were the first to develop a model for project scheduling and human resources allocation, but the situation where human resources possessed multiskills was not considered in the model. Janiak and Rudek [17] proposed a single processor (human worker) scheduling model with multiskill learning effect. But in the model the experience formula was oversimplified where the experience of a worker was assumed to be either 0 or 1; besides it could not be applied to project scheduling problem directly. Heimerl and Kolisch [18] formulated a work assignment model for multiskilled human resources, and many affecting factors were considered in the learning equation. But the equation took the form of position-based approach, which was unsuitable for human

learning. Furthermore, the induced learning effect was not considered in the above-mentioned three papers. Their models could not assist the manager to determine an optimal level of induced learning.

To fill this gap, we propose a new model for project scheduling with multiskilled human resources, in which the learning formula is improved. In order to solve the proposed model, a genetic algorithm is introduced.

The remainder of the paper is organized as follows. Section 2 formulates the project scheduling problem considering multiskill learning effects. Genetic algorithm for solving this problem is described in Section 3. Numerical example is illustrated and discussed in Section 4. Finally, Section 5 concludes the paper.

## 2. Problem Formulation

We consider a project including  $J$  activities. There are precedence relations between them due to technological requirements. The structure of the project can be depicted as a network  $G = (V, E)$ . The nodes in the network represent activities, and the arcs between them represent precedence relations. Activity 1 is the source node of the network and activity  $J$  is the sink node. They are both dummy activities. Each activity  $j$  ( $j = 2, \dots, J - 1$ ) is performed by multiskilled human workers. Each worker assigned to the activity can only select one of the skills to perform. When the required quantities of each skill are available, the activity can be started. Once it is started, it may take a certain duration and not be preempted.

The efficiency of human workers might be influenced by learning effects during the project implement. Therefore, the actual duration for each activity might be adjusted in project scheduling. There are two kinds of learning, which are autonomous learning and induced learning. The autonomous learning is learning-by-doing, which means higher efficiency can be achieved by repeating similar activities. The induced learning is the proactive investment in technological knowledge, including worker's training, job-related instructions, and handbooks or supervisors. It can facilitate the learning effect so as to shorten the durations. However, extra cost may be also incurred [10]. Hence, the total cost for project may change too.

In this paper, we develop a novel model for the project scheduling problem with multiskilled worker, in which autonomous and induced learning are both considered. It can assist the project manager to determine the start time, the finish time, and resource allocation for each activity. Moreover, combined with the total cost formula, the project manager can balance the project makespan with the total cost. The proposed model is formulated as follows:

$$\min T = f_j \tag{1}$$

$$\text{s.t: } f_j - f_i \geq d_j, \quad i \in P_j \tag{2}$$

$$d_j = \max_{k=1, \dots, K} \left\{ \sum_{l=1}^L \sum_{r=r_{jks}}^{r_{jklf}} d_{lr} \right\} \tag{3}$$

$$d_{lr} = \begin{cases} d_{l1}, & r = 1 \\ \max \left\{ \left( \sum_{n=1}^{r-1} d_{ln} \right)^{\alpha_l}, d_{lm} \right\}, & r \geq 2 \end{cases} \tag{4}$$

$$a_l = \log_2 (1 - x_l) LR_l, \quad 0 \leq x_l \leq x_{lm} < 1 \tag{5}$$

$$\sum_{j \in A_t} as_{jl} \leq \sum_{k=1}^K rs_{kl}, \quad A_t = \{j \mid f_j - d_j < t \leq f_j\} \tag{6}$$

$$j = 1, \dots, J, \quad k = 1, \dots, K, \quad l = 1, \dots, L. \tag{7}$$

In this model, the objective function (1) minimizes the project makespan  $T$ , where  $f_j$  is the finish time of dummy activity  $J$ . Constraint (2) represents the precedence relations, where  $P_j$  denotes the predecessors of activity  $j$ . The duration  $d_j$  in constraint (2) is calculated by formulas (3)–(5).

According to formula (3),  $d_j$  depends on the maximum time that all the workers spend in activity  $j$ . In formula (3),  $d_{lr}$  denotes the time required for a worker to finish the  $r$ th unit of job  $l$ . The ordinal number  $r_{jkl}$  means the accumulative units of job  $l$  that worker  $k$  have done since he finished the first unit of that job in activity  $j$ , while  $r_{jklf}$  means the corresponding accumulative units when he finished the last unit of job  $l$  in activity  $j$ .

Formula (4) is the sum-of-processing-time based learning function, which is proposed by Kuo and Yang [19]. In formula (4),  $d_{l1}$  denotes the time required for a worker to finish the first unit of job using skill  $l$ , which is also the normal time for a worker to finish a unit of that job without learning effect. Parameter  $d_{lm}$  is defined as the minimum value of  $d_{lr}$ . In addition,  $a_l$  denotes the learning index of skill  $l$ , which reflects the learning speed on skill  $l$ .

The value of learning index can be calculated according to formula (5), which is proposed by Biskup and Simons [10]. In formula (5),  $x_l$  denotes the induced learning level of skill  $l$ , and  $x_{lm}$  is defined as the maximum value of  $x_l$ . Parameter  $LR_l$  indicates the autonomous learning ratio of skill  $l$ . When  $x_l$  is equal to 0, it means that there is only autonomous learning effect existing for skill  $l$ . Otherwise, both autonomous and induced learning effects for skill  $l$  are considered.

Constraint (6) represents that the quantity of skills required by activities cannot exceed what the workers can provide. Parameter  $as_{jl}$  denotes the quantity of skill  $l$  required by activity  $j$ , while 0-1 binary variable  $rs_{jl}$  indicates whether worker  $k$  possesses skill  $l$  or not. When  $rs_{jl}$  is equal to 1, worker  $k$  possesses skill  $l$  and vice versa.

Constraint (7) limits the range of  $j, k$ , and  $l$ .

The formula for extra cost incurred by induced learning is given below:

$$C = \sum_{j=1}^J \sum_{k=1}^K \sum_{l=1}^L c_l d_{jkl} + \sum_{l=1}^L b_l x_l^2 + c_{in} T. \tag{8}$$

In formula (9),  $c_l$  denotes the salary paid for a worker using skill  $l$  per day,  $b_l$  indicates the cost coefficient of induced learning for skill  $l$ , and  $c_{in}$  is the daily indirect cost.

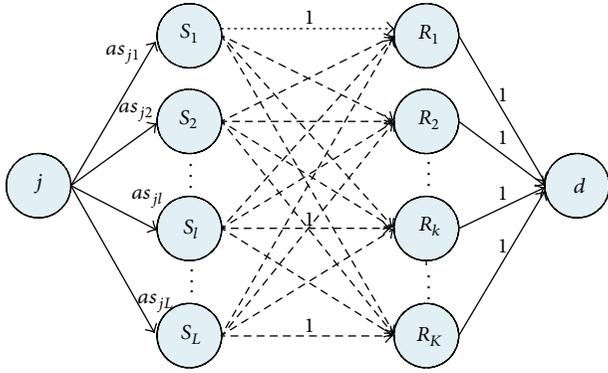


FIGURE 1: Maximum flow model for flexible resource allocation.

### 3. Genetic Algorithm

Genetic algorithm was proposed by Goldberg and Holland [20], which was inspired by Darwin’s theory of evolution. According to the algorithm, the initial population with suitable encoding and decoding can evolve an optimal solution to real-world problem based on the natural principles of natural selection, crossover, and mutation. Due to its excellent performance, Genetic algorithm was widely used for project scheduling problem [21]. In this paper, we applied a genetic algorithm with specific coding and decoding mechanism to solve the above-mentioned formulation. The procedure of the genetic algorithm is described below.

*Step 1 (initial population).* Genetic algorithm begins by means of generating an initial population randomly. Parameter pop is defined as the number of individuals in the population. In this paper, the individuals are encoded in the form of activity list. It is a permutation vector of activities conforming to precedence relation, which represents a feasible solution for schedule [22].

*Step 2 (fitness computation).* To compute the fitness values, we decode the individuals with serial scheduling generation scheme [23], maximum flow algorithm [24], and EFT (early finish time) priority rule.

The serial scheduling generation scheme means the activities are selected according to their order in the list and scheduled at their earliest start period when the available resource is abundant. In order to determine whether the flexible resource allocation at one point is feasible or not, we introduce the maximum flow algorithm proposed by Ford and Fulkerson [24].

The flexible resource allocation can be modeled as a maximum flow network [25, 26], as shown in Figure 1. In the network, activity node  $j$  is the source node, while dummy node  $d$  is the sink node. Between source and sink node, there are skill nodes and resource nodes. The edges connecting activity node and skill nodes reflect the requirement of each skill for activity  $j$ , where the capacity for each edge represents the required quantity of each skill. While the edges linking resource and skill nodes illustrate the resource-skill relations. According to the relations, the capacity of each edge is set to

be either 1 or 0. Besides, the edges connecting resource nodes and dummy nodes show the resource availability at a point in scheduling generation. When the resource is available, the capacity of the edge is set to be 1; otherwise, it is set to be 0. If the maximum flow obtained by the Ford and Fulkerson algorithm is equal to the sum of  $as_{j_l}$ , the resource allocation for activity  $j$  is feasible at this moment.

As there may be multiple feasible options for resource allocation of activity  $j$ , the eventual resource allocation can be determined by EFT priority rule. According to EFT rule, the option with minimized early finish time would be selected.

After decoding, the fitness for individual  $I$  can be calculated by the following:

$$\text{fitness}(i) = \frac{1}{T(i)}. \tag{9}$$

In formula (10),  $T(i)$  denotes the makespan value of individual  $i$ .

*Step 3 (roulette wheel selection).* In the selection stage, pop individuals are selected to reproduce according to their fitness values. With roulette wheel method, an individual can be chosen more than once. The selective probability of individual  $i$  in population can be calculated by the following:

$$p(i) = \frac{\text{fitness}(i)}{\sum_{n=1}^{\text{pop}} \text{fitness}(n)}. \tag{10}$$

*Step 4 (partially matched crossover).* In this step, the partially matched crossover operator is applied on two different individuals from current population with a probability of crossover  $p_{\text{cos}}$ .

For two parents  $i^f$  and  $i^m$  selected from the population, after an integer number  $q_{\text{cos}}$  ( $1 \leq q_{\text{cos}} < J$ ) is randomly generated, a son  $i^s$  and  $i^d$  are produced. For the son  $i^s$ , the first  $q_{\text{cos}}$  activities are taken directly from  $i^f$ , while the remaining activities are arranged according to their order in  $i^m$ , while the daughter takes the first  $q_{\text{cos}}$  activities from mother and rearranges the remaining according to father.

*Step 5 (swap mutation).* In this step, the swap mutation operator is applied on newly generated individuals with a probability of mutation  $p_{\text{mut}}$ . Swap mutation can be described as follows.

A position  $q_{\text{mut}}$  ( $1 \leq q_{\text{mut}} \leq J$ ) is randomly chosen; then are swapped their contents if activity in position  $q_{\text{mut}} + 1$  is not an immediate successor of activity in position  $q_{\text{mut}}$ . Otherwise, another position is randomly selected. This process repeats until the two adjacent activities can be permuted.

*Step 6 (stop criterion).* The algorithm stops if the predefined evolutionary generation  $gen$  is reached. As the stop criterion is satisfied, the optimization result is given as output. Otherwise, the algorithm goes to Step 2 and continues to iterate.

### 4. Computational Experiment

In order to illustrate the model proposed in this paper, a numerical example is given below. As there are no benchmark

TABLE 1: Activities of example project.

Activity no.	Successors	Normal duration (day)	Skill requirement			
			$s_1$	$s_2$	$s_3$	$s_4$
1	2, 3, 4	0	0	0	0	0
2	11	22	0	2	0	2
3	5, 6, 7	25	2	0	0	3
4	9, 10, 11	9	1	0	3	0
5	8	19	0	1	4	0
6	9, 10	13	0	4	0	3
7	9	28	0	1	0	2
8	10	3	0	3	4	0
9	12	19	2	0	0	3
10	12	13	0	4	0	2
11	12	25	0	4	0	1
12	—	0	0	0	0	0

TABLE 2: Worker-skill mapping relationships.

Worker	Skill			
	$s_1$	$s_2$	$s_3$	$s_4$
$w_1$	1	1	1	1
$w_2$	1	0	1	1
$w_3$	1	1	1	0
$w_4$	0	1	1	1
$w_5$	1	1	0	1
$w_6$	1	0	1	0
$w_7$	0	1	1	0
$w_8$	0	1	0	1
$w_9$	1	0	0	1
$w_{10}$	0	0	1	1

examples for project scheduling problem with multiskill learning, we randomly generate an example project with multiskilled workers based on the existing research [2]. After that, the learning parameters and cost parameters are added into the generated example. In the example project, there are 12 activities, and the details of them are listed in Table 1. In addition, 10 workers ( $w_1-w_{10}$ ) are involved in the project. Each worker is assumed to possess at least one of the four skills ( $s_1-s_4$ ). The mapping relationships between the workers and the skills are present in Table 2. Besides, the related learning parameters for each skill are shown in Table 3. Finally, the cost parameters in the numerical are given below:  $b_1 = 2.5 \times 10^6$  \$,  $b_2 = 3.5 \times 10^6$  \$,  $b_3 = 4.5 \times 10^6$  \$,  $b_4 = 3.0 \times 10^6$  \$,  $c_1 = 200$  \$,  $c_2 = 400$  \$,  $c_3 = 500$  \$,  $c_4 = 300$  \$, and  $c_{in} = 1200$  \$.

In the experiment, the parameters of genetic algorithm are set as follows: population size  $pop = 50$ , maximum generation  $gen = 100$ , crossover probability  $p_{cos} = 0.95$ , and mutation probability  $p_{mut} = 0.1$ .

When the induced learning with top level is considered, the optimal solution is 1-3-2-5-8-6-4-7-10-11-9-12. After decoding it, we obtain a schedule with a makespan of 71.5 days. The optimal schedule is shown in Table 4. In addition,

TABLE 3: Learning parameters on skills.

Skill	Autonomous learning ratio	Top level of induced learning	Minimum working time per unit of job (day)
$s_1$	0.96	0.1	0.43
$s_2$	0.95	0.1	0.41
$s_3$	0.93	0.1	0.37
$s_4$	0.98	0.1	0.47

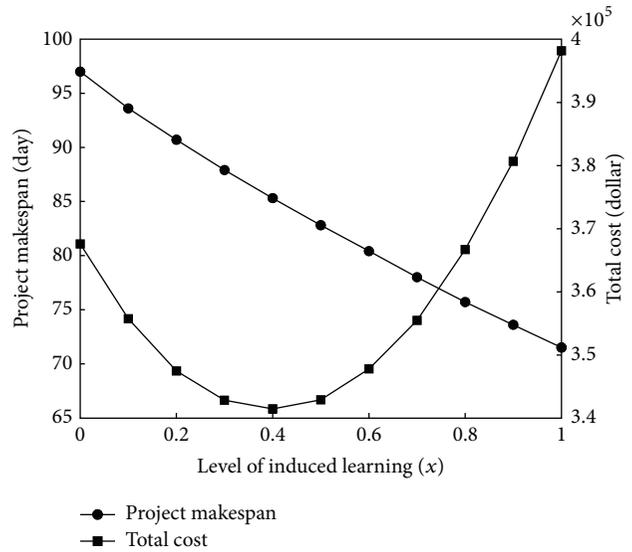


FIGURE 2: Project makespan-induced learning and total cost-induced learning curves.

due to the learning effect, workers allocated to the same activity may have various working times. The details on working time in the aforementioned schedule are present in Table 5. From Tables 4 and 5, we can find out start and finish time for activities, resource allocation, and details on working time. For example, we learn that activity 7 is started on day 43.2 and finished on day 58.1. There are 3 workers ( $w_1-w_3$ ) allocated to the activity. The worker  $w_1$  and  $w_2$  are assigned to use skill  $s_4$ , while worker  $w_3$  uses skill  $s_1$ . In addition, all of them work in the activity for 14.9 days.

When no learning effect is considered in the project scheduling, the optimal solution is 1-2-4-3-11-7-5-6-8-9-10-12. The corresponding project makespan is 114 days and the optimal schedule is shown in Table 6.

Comparing the two schedules, we find that project makespan can be even shortened by 37.2% based on learning effect. The result demonstrated that learning effect cannot be neglected in project scheduling since it shortens the project makespan.

Furthermore, we define  $x$  as the induced learning level of the four skills ( $s_1-s_4$ ). Then, we discuss the changes in project makespan and total cost with  $x$ . The corresponding two curves are shown in Figure 2. It can be seen that the project makespan monotonically decreases when  $x$  increases from 0 to 1. This is because the induced learning can help the workers improve their efficiency throughout the project.

TABLE 4: Optimal schedule with top level of induced learning.

Activity no.	Start time (day)	Finish time (day)	Resource allocation										
			w <sub>1</sub>	w <sub>2</sub>	w <sub>3</sub>	w <sub>4</sub>	w <sub>5</sub>	w <sub>6</sub>	w <sub>7</sub>	w <sub>8</sub>	w <sub>9</sub>	w <sub>10</sub>	
1	0	0	—	—	—	—	—	—	—	—	—	—	—
2	0	15.8	—	—	—	—	—	—	s <sub>2</sub>	s <sub>2</sub>	s <sub>4</sub>	s <sub>4</sub>	—
3	0	17.6	s <sub>4</sub>	s <sub>4</sub>	s <sub>1</sub>	s <sub>4</sub>	s <sub>1</sub>	—	—	—	—	—	—
4	15.8	23	—	—	—	—	—	s <sub>3</sub>	s <sub>3</sub>	—	—	s <sub>1</sub>	s <sub>3</sub>
5	17.6	30.7	s <sub>3</sub>	s <sub>3</sub>	s <sub>3</sub>	s <sub>3</sub>	s <sub>2</sub>	—	—	—	—	—	—
6	33.6	43.2	s <sub>4</sub>	s <sub>4</sub>	s <sub>2</sub>	s <sub>4</sub>	s <sub>2</sub>	—	s <sub>2</sub>	s <sub>2</sub>	—	—	—
7	43.2	58.1	s <sub>4</sub>	s <sub>4</sub>	s <sub>2</sub>	—	—	—	—	—	—	—	—
8	30.7	33.6	s <sub>3</sub>	s <sub>3</sub>	s <sub>3</sub>	s <sub>2</sub>	s <sub>2</sub>	s <sub>3</sub>	s <sub>2</sub>	—	—	—	—
9	58.1	71.5	s <sub>4</sub>	s <sub>4</sub>	s <sub>1</sub>	—	—	s <sub>1</sub>	—	—	—	—	s <sub>4</sub>
10	43.2	51.7	—	—	—	s <sub>2</sub>	s <sub>2</sub>	—	s <sub>2</sub>	s <sub>2</sub>	s <sub>4</sub>	s <sub>4</sub>	—
11	51.7	65.2	—	—	—	s <sub>2</sub>	s <sub>2</sub>	—	s <sub>2</sub>	s <sub>2</sub>	s <sub>4</sub>	—	—
12	71.5	71.5	—	—	—	—	—	—	—	—	—	—	—

TABLE 5: Details on working time with top level of induced learning.

Worker	Working time in each activity (day)												
	1	2	3	4	5	6	7	8	9	10	11	12	
w <sub>1</sub>	0	0	17.6	0	12.6	7.5	14.9	1.6	9.6	0	0	0	0
w <sub>2</sub>	0	0	17.6	0	12.6	7.5	14.9	1.6	9.6	0	0	0	0
w <sub>3</sub>	0	0	16.8	0	12.6	9.6	14.9	1.6	10	0	0	0	0
w <sub>4</sub>	0	0	17.6	0	12.6	7.5	0	2.9	0	8.5	13.1	0	0
w <sub>5</sub>	0	0	16.8	0	13.1	6.8	0	1.7	0	6.3	11.3	0	0
w <sub>6</sub>	0	0	0	5.9	0	0	0	2.8	13.4	0	0	0	0
w <sub>7</sub>	0	14.8	0	6.9	0	6.7	0	1.6	0	6.2	11.2	0	0
w <sub>8</sub>	0	14.8	0	0	0	6.8	0	0	0	6.3	11.3	0	0
w <sub>9</sub>	0	15.8	0	7.2	0	0	0	0	0	7.6	13.5	0	0
w <sub>10</sub>	0	15.8	0	6.9	0	0	0	0	10.4	7.6	0	0	0

On the contrary, the total cost firstly decreases and then increases with increasing of  $x$ . This is because the induced learning can not only reduce cost by shortening the durations but also incur extra cost. When the reduced cost exceeds the incurred, the total cost decreases. Otherwise, the total cost increases.

The results illustrate that each level of induced learning corresponds to a combination of project makespan and total cost. By means of determining the level of induced learning, the project manager can balance them in project scheduling.

### 5. Conclusion

In this paper, we proposed a model for project scheduling problem with multiskill learning effect. In this model, both autonomous and induced learning are considered. Moreover, the learning function is sum-of-process time based, which is a suitable hypothesis for human learning. In order to solve the model, a genetic algorithm is introduced. The individuals in the algorithm are encoded in the form of activity list and decoded with SGS (serial generation scheme), maximum flow

algorithm, and EFT (early finish time) priority rule. Partially matched crossover and swap mutation are applied to make the individuals feasible.

A numerical example is given to illustrate the model. The computational results show that learning effect cannot be neglected in project scheduling. The induced learning can affect both project makespan and total cost. By means of determining the level of induced learning, the project manager can balance them in project scheduling. Due to the uncertainty nature of project, the future research will extend the problem to uncertainty projects environment.

### Conflict of Interests

The authors declare that they have no financial and personal relationships with other people or organizations that can inappropriately influence their work; there is no professional or other personal interests of any nature or kind in any product, service, and/or company that could be construed as influencing the position presented in, or the review of, the paper.

TABLE 6: Optimal schedule considering no learning effect.

Activity no.	Start time (day)	Finish time (day)	Resource allocation										
			$w_1$	$w_2$	$w_3$	$w_4$	$w_5$	$w_6$	$w_7$	$w_8$	$w_9$	$w_{10}$	
1	0	0	—	—	—	—	—	—	—	—	—	—	—
2	0	22	$s_4$	$s_4$	$s_2$	$s_2$	—	—	—	—	—	—	—
3	9	34	—	—	—	—	$s_4$	$s_1$	—	$s_4$	$s_1$	$s_4$	—
4	0	9	—	—	—	—	$s_1$	$s_3$	$s_3$	—	—	—	$s_3$
5	47	66	$s_3$	$s_3$	$s_3$	$s_2$	—	$s_3$	—	—	—	—	—
6	66	79	$s_4$	$s_4$	$s_2$	$s_4$	$s_2$	—	$s_2$	$s_2$	—	—	—
7	34	62	—	—	—	—	$s_4$	—	—	$s_2$	$s_4$	—	—
8	79	82	$s_3$	$s_3$	$s_3$	$s_2$	$s_2$	$s_3$	$s_2$	—	—	—	—
9	82	101	$s_4$	$s_4$	$s_1$	$s_4$	$s_1$	—	—	—	—	—	—
10	101	114	$s_4$	$s_4$	$s_2$	$s_2$	$s_2$	—	$s_2$	—	—	—	—
11	22	47	$s_2$	$s_4$	$s_2$	$s_2$	—	—	$s_2$	—	—	—	—
12	114	114	—	—	—	—	—	—	—	—	—	—	—

**Acknowledgment**

This paper is supported by the National Natural Science Foundation of China (no. 71272146).

**References**

[1] S. Hartmann and D. Briskorn, “A survey of variants and extensions of the resource-constrained project scheduling problem,” *European Journal of Operational Research*, vol. 207, no. 1, pp. 1–14, 2010.

[2] E. Alba and J. Francisco Chicano, “Software project management with GAs,” *Information Sciences*, vol. 177, no. 11, pp. 2380–2401, 2007.

[3] H. Li and K. Womer, “Scheduling projects with multi-skilled personnel by a hybrid MILP/CP benders decomposition algorithm,” *Journal of Scheduling*, vol. 12, no. 3, pp. 281–298, 2009.

[4] C. Heimerl and R. Kolisch, “Scheduling and staffing multiple projects with a multi-skilled workforce,” *OR Spectrum*, vol. 32, no. 2, pp. 343–368, 2010.

[5] V. Yannibelli and A. Amandi, “A knowledge-based evolutionary assistant to software development project scheduling,” *Expert Systems with Applications*, vol. 38, no. 7, pp. 8403–8413, 2011.

[6] M. J. Anzanello and F. S. Fogliatto, “Learning curve models and applications: literature review and research directions,” *International Journal of Industrial Ergonomics*, vol. 41, no. 5, pp. 573–583, 2011.

[7] D. Biskup, “A state-of-the-art review on scheduling with learning effects,” *European Journal of Operational Research*, vol. 188, no. 2, pp. 315–329, 2008.

[8] M. J. Anzanello and F. S. Fogliatto, “Learning curve models and applications: literature review and research directions,” *International Journal of Industrial Ergonomics*, vol. 41, no. 5, pp. 573–583, 2011.

[9] T. E. Cheng and G. Wang, “Single machine scheduling with learning effect considerations,” *Annals of Operations Research*, vol. 98, no. 1–4, pp. 273–290, 2000.

[10] D. Biskup and D. Simons, “Common due date scheduling with autonomous and induced learning,” *European Journal of Operational Research*, vol. 159, no. 3, pp. 606–616, 2004.

[11] J.-B. Wang, “A note on scheduling problems with learning effect and deteriorating jobs,” *International Journal of Systems Science*, vol. 37, no. 12, pp. 827–833, 2006.

[12] W.-H. Kuo and D.-L. Yang, “Minimizing the total completion time in a single-machine scheduling problem with a time-dependent learning effect,” *European Journal of Operational Research*, vol. 174, no. 2, pp. 1184–1190, 2006.

[13] R. Rudek, “The single processor total weighted completion time scheduling problem with the sum-of-processing-time based learning model,” *Information Sciences*, vol. 199, pp. 216–229, 2012.

[14] C.-C. Wu and W.-C. Lee, “Single-machine and flowshop scheduling with a general learning effect model,” *Computers and Industrial Engineering*, vol. 56, no. 4, pp. 1553–1558, 2009.

[15] Y.-Y. Lu, C.-M. Wei, and J.-B. Wang, “Several single-machine scheduling problems with general learning effects,” *Applied Mathematical Modelling*, vol. 36, no. 11, pp. 5650–5656, 2012.

[16] M.-C. Wu and S.-H. Sun, “A project scheduling and staff assignment model considering learning effect,” *International Journal of Advanced Manufacturing Technology*, vol. 28, no. 11–12, pp. 1190–1195, 2006.

[17] A. Janiak and R. Rudek, “A note on a makespan minimization problem with a multi-ability learning effect,” *Omega*, vol. 38, no. 3–4, pp. 213–217, 2010.

[18] C. Heimerl and R. Kolisch, “Work assignment to and qualification of multi-skilled human resources under knowledge depreciation and company skill level targets,” *International Journal of Production Research*, vol. 48, no. 13, pp. 3759–3781, 2010.

[19] W.-H. Kuo and D.-L. Yang, “Minimizing the makespan in a single machine scheduling problem with a time-based learning effect,” *Information Processing Letters*, vol. 97, no. 2, pp. 64–67, 2006.

[20] D. E. Goldberg and J. H. Holland, “Genetic algorithms and machine learning,” *Machine Learning*, vol. 3, no. 2, pp. 95–99, 1988.

[21] R. Kolisch and S. Hartmann, “Experimental investigation of heuristics for resource-constrained project scheduling: an update,” *European Journal of Operational Research*, vol. 174, no. 1, pp. 23–37, 2006.

[22] K. Moumene and J. A. Ferland, “Activity list representation for a generalization of the resource-constrained project scheduling

- problem,” *European Journal of Operational Research*, vol. 199, no. 1, pp. 46–54, 2009.
- [23] R. Kolisch, “Serial and parallel resource-constrained project scheduling methods revisited: theory and computation,” *European Journal of Operational Research*, vol. 90, no. 2, pp. 320–333, 1996.
- [24] L. Ford and D. Fulkerson, “Maximal flow through a network,” *Canadian Journal of Mathematics*, vol. 8, pp. 399–404, 1956.
- [25] J. Yang and Z. Fei, “Bipartite graph based dynamic spectrum allocation for wireless mesh networks,” in *Proceedings of the 28th International Conference on Distributed Computing Systems Workshops (ICDCS '08)*, pp. 96–101, IEEE press, June 2008.
- [26] C. Liu, G. Chang, and J. Jia, “A new channel assignment algorithm in wireless mesh network,” in *Recent Advances in Computer Science and Information Engineering*, vol. 127 of *Lecture Notes in Electrical Engineering*, pp. 511–516, 2012.