Int J Adv Manuf Technol (2006) 28: 1190–1195
DOI 10.1007/s00170-004-2465-0

ORIGINAL ARTICLE

Muh-Cherng Wu · Shih-Hsiung Sun

A project scheduling and staff assignment model considering learning effect

Received: 27 July 2004 / Accepted: 25 October 2004 / Published online: 21 September 2005

Abstract In a multi-project environment, we sometimes need to periodically schedule the tasks for each project and assign staff to the tasks. Such a decision-making problem has been studied in literature; however, learning effect of staff has not been considered in previous studies. This research formulates a mixed nonlinear program for project scheduling and staff allocation problems, which considers learning effect of staff. The objective function is to minimize outsourcing costs. A genetic algorithm (GA) is proposed to solve the problem. Experiments for solving various sizes of test problems has been carried out to validate the proposed GA.

Keywords Genetic algorithm · Learning effect · Project scheduling · Staff allocation

1 Introduction

In a research and development (R&D) department, several projects may have to be simultaneously implemented in a certain time period. Scheduling multiple projects is complicated due to the limitation of staff resources and the efficiency variation among staff. Effectively scheduling project tasks and allocating staff to these tasks is therefore very important. Such a decision-making problem is called the project-scheduling and staff-allocation problem.

Campbell et al. [1, 2] assume that the schedule of projects is available and develop mathematical models for the staff-allocation problem, which includes modelling the variation in staff efficiency. That is, each staff has different efficiency in performing different tasks. These mathematical models for staff allocation, being integer programs, require long computation time. Therefore, some studies aimed to develop heuristic methods to solve the problems more efficiently [3–5].

Bassett [6] proposed a mixed integer linear programming model for solving the project-scheduling and staff-allocation problem, which is distinct in two characteristics. First, the capabilities of staff are different; that is, some tasks can only be performed by certified staff. Second, each staff has a limited number of working days. To solve the integer program efficiently, a heuristic method has been proposed.

Some other works, through empirical studies, aimed to investigate the behavioural factors associated with staff allocation. Hendriks et al. [7] characterized two behavioural factors that greatly influence the effect of staff allocation. One factor, called project scatter, indicates that the efficiency of a team may decrease if the number of team staff is more than needed. The other factor, called resource dedication, denotes that the dedication of a staff to a particular task will increase efficiency.

Another behavioural study, by Hankawa et al. [8], reveals that learning effect does exist in software development. Learning effect denotes that staff will perform a task more efficiently if he/she stays on the task longer. The learning effect of engineers, though very important for project scheduling and staff allocation, has not been included in the mathematical model published in previous literature.

This paper presents a mathematical model for a project-scheduling and staff-allocation problem that considers learning effect. This problem can be characterized as follows. Many projects are performed in a time horizon, which covers several time periods. At the end of each period, staff can be reallocated to different tasks. The decision, in which the learning effect of staff should be considered, is to schedule the tasks of each project and assign staff to these tasks for each period. The objective function is to minimize outsourcing costs. The proposed model is a mixed non-linear integer program and is solved by a proposed genetic algorithm.

The remainder of this paper is organized as follows. Section 2 presents a model for characterizing the learning effect of R&D engineers. Section 3 describes the mathematical model

M.-C. Wu (✉) · S.-H. Sun
Department of Industrial Engineering and Management,
National Chiao Tung University,
Hsin-Chu, Taiwan, R.O.C.
E-mail: mcwu@cc.nctu.edu.tw
Fax: +886-35-720-610

for project scheduling and staff allocation. Section 4 states the genetic algorithm for solving the problem. Experiments for justifying the solution method are discussed in Sect. 5. Some concluding remarks are present in the last section.

2 Learning effect modeling

Wright pioneered the study of learning effect (also called learning curve) [9]. Learning effect denotes that the efficiency of staff will improve by doing more. That is, the longer staff works on a particular task, the more efficient will be the staff on that task. Learning curve has been widely applied to various areas, and a comprehensive survey has been published [10].

Smith and Larsson investigated the learning effect of a heart transplant facility [11]. Their study revealed that the cumulative average resource consumption decreases exponentially over x, the total number of operation cases, and finally reaches a steady state. The task of heart transplant is much like a task of R&D in engineering. Based on Smith and Larsson's work, we propose a learning curve as follows, for modeling the efficiency of an R&D staff (p) working on a project task (k).

$$\bar{E}_n = \bar{E}_1 n^b ,$$

- n total number of time periods spent by staff p on task k
- $\bar{E}_n$ cumulative average efficiency when staff p has spent n periods on task k
- b $b = -\frac{\ln(r)}{\ln 2}$; $(0 < r \leq 1)$, b is called learning factor, and r is called learning percentage. The smaller is the value of r, the larger is the value of b, and the higher is the learning effect.

The proposed learning curve for the case of $r = 0.95$ is shown in Fig. 1. The figure shows that the cumulative average efficiency, starting from 0.8, increases over time and finally reaches 1.0. Notice that $\bar{E}_n$ represents the cumulative average efficiency of staff for working n days. Let E_n represent the efficiency of staff at the n-th day. Then,

$$E_n = \frac{\bar{E}_n\, n - \bar{E}_{n-1}(n-1)}{n-(n-1)} = \bar{E}_n\, n - \bar{E}_{n-1}(n-1) .$$

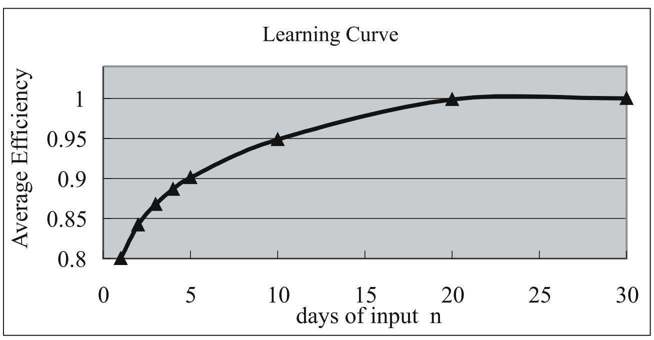

Fig. 1. Learning curve modelled by cumulative average efficiency

3 Model for project scheduling and staff allocation

The decision problem of interest is to schedule project tasks and allocate staff in a multi-project environment. Assumptions of the decision problem are described below.

In a time horizon involving n periods, k projects are to be implemented. Project x involves $p(x)$ number of tasks $(1 \leq x \leq k)$. Let $m = \sum_{x=1}^{k} p(x)$ denote the total number of tasks to be scheduled, S represent the total number of staff. The staff allocation decision is to assign staff i to task j $(1 \leq i \leq S,\ 1 \leq j \leq m)$ for each period t $(1 \leq t \leq n)$. The task scheduling decision is to schedule task j in the time horizon; that is, the workload of each task at each period t has to be determined.

A mixed nonlinear integer program is proposed to simultaneously model the two decisions. The model assumes that an outsourcing strategy will be applied whenever any task cannot be completed before the due date. The objective is to minimize the outsourcing costs.

Notation of the model is given below, in which the standard time for completing a task implies that its associated efficiency is 1.0.

(A) Notation

Parameters

- d_j standard time for completing task j, $1 \leq j \leq m$
- s_j allowable starting period of task j, $1 \leq j \leq m$
- e_j allowable ending period of task j, $1 \leq j \leq m$
- T_j allowable working interval of task j, $T_j = (s_j, e_j)$, $1 \leq j \leq m$
- C_j outsourcing cost per staff for performing task j, $1 \leq j \leq m$
- $H_{i,t}$ maximum allowable working days of staff i in period t, $1 \leq i \leq S$, $1 \leq t \leq n$
- M a large positive integer
- h maximum allowable working days of an outsourcing staff per period
- z efficiency of outsourcing staff, a constant in the time horizon
- b learning factor

Variables

- $X_{i,j,t}$ binary variable $X_{i,j,t} = 1$ denotes that staff i is assigned to task j in period t, and $X_{i,j,t} = 0$ implies no such assignment
- $Y_{j,t}$ number of outsourcing staff hired in period t to perform task j
- $\bar{E}_{i,j,t}$ cumulative average efficiency of staff i working task j up to period t
- $E_{i,j,t}$ efficiency of staff i working task j at period t
- $F_{j,t}$ completion percentage of task j in period t

(B) Mathematical model

The proposed mathematical model is formulated as follows.

$$\min \mathrm{TC} = \sum_{t=1}^{n} \sum_{j=1}^{m} Y_{j,t}\, C_j$$

s.t.

$$\sum_{i=1}^{S} \left(H_{i,t}\, X_{i,j,t}\, E_{i,j,t}\right) + Y_{j,t}\, h\, z \geq d_j\, F_{j,t}\,, \quad \text{for } 1 \leq j \leq m\,,\ 1 \leq t \leq n \tag{1}$$

$$\sum_{t \in T_j}^{F_{j,t}} = 1\,, \quad \text{for } 1 \leq j \leq m \tag{2}$$

$$Y_{j,t+1} \leq Y_{j,t}\,, \quad \text{for } 1 \leq j \leq m\,,\ 1 \leq t \leq n \tag{3}$$

$$\sum_{j=1}^{m} X_{i,j,t} = 1\,, \quad \text{for } 1 \leq i \leq S\,,\ 1 \leq t \leq n \tag{4}$$

$$\bar{E}_{i,j,t+1} \leq \bar{E}_{i,j,1} \left(\sum_{\tau=1}^{t} H_{i,\tau}\, X_{i,j,\tau}\right)^{b} + M\left(1 - X_{i,j,t}\right), \quad \text{for } 1 \leq i \leq S\,,\ 1 \leq j \leq m\,,\ 1 \leq t \leq n \tag{5}$$

$$\bar{E}_{i,j,t+1} \geq \bar{E}_{i,j,1} \left(\sum_{\tau=1}^{t} H_{i,\tau}\, X_{i,j,\tau}\right)^{b} - M\left(1 - X_{i,j,t}\right), \quad \text{for } 1 \leq i \leq S\,,\ 1 \leq j \leq m\,,\ 1 \leq t \leq n \tag{6}$$

$$\bar{E}_{i,j,t+1} \leq \bar{E}_{i,j,t} + M\, X_{i,j,t}\,, \quad \text{for } 1 \leq i \leq S\,,\ 1 \leq j \leq m\,,\ 1 \leq t \leq n \tag{7}$$

$$\bar{E}_{i,j,t+1} \geq \bar{E}_{i,j,t} - M\, X_{i,j,t}\,, \quad \text{for } 1 \leq i \leq S\,,\ 1 \leq j \leq m\,,\ 1 \leq t \leq n \tag{8}$$

$$E_{i,j,1} = \bar{E}_{i,j,1}\,, \quad \text{for } 1 \leq i \leq S\,,\ 1 \leq j \leq m \tag{9}$$

$$E_{i,j,t+1} = \max\left(\frac{\bar{E}_{i,j,t+1} \sum\limits_{\tau=1}^{t} H_{i,\tau}\, X_{i,j,\tau} - \bar{E}_{i,j,t} \sum\limits_{\tau=1}^{t-1} H_{i,\tau}\, X_{i,j,\tau}}{\max\left(\sum\limits_{\tau=1}^{t} H_{i,\tau}\, X_{i,j,\tau} - \sum\limits_{\tau=1}^{t-1} H_{i,\tau}\, X_{i,j,\tau},\, 1\right)},\, E_{i,j,t}\right), \quad \text{for } 1 \leq i \leq S\,,\ 1 \leq j \leq m\,,\ 1 \leq t \leq n \tag{10}$$

$$X_{i,j,t} = 0 \text{ or } 1\,, \quad \text{for } 1 \leq i \leq S\,,\ 1 \leq j \leq m\,,\ 1 \leq t \leq n \tag{11}$$

$$E_{i,j,t} \geq 0\,;\quad F_{j,t} \geq 0\,;\quad Y_{j,t} \in N\,, \quad \text{for } 1 \leq i \leq S\,,\ 1 \leq j \leq m\,,\ 1 \leq t \leq n \tag{12}$$

The objective function is to minimize the total outsourcing cost. Constraint 1 indicates that staff resources assigned to a task should be enough to complete the work load of the scheduled task in each period. Constraint 2 indicates that each task should be completed within the allowable working interval. Constraint 3 denotes that outsourcing should be performed as early as possible whenever a task demands, to ensure the on time completion of each task. Constraint 4 denotes that each staff should be allocated to only one task in each period.

Constraints 5–8, which should be considered as a set, ensure that the cumulative average efficiency of staff improves only when he/she works more time on the task. Of these four constraints, constraints 5–6 are for modeling the case $X_{i,j,t} = 1$ and constraints 7–8 are for the case $X_{i,j,t} = 0$. Constraint 9 sets the starting efficiency of each staff. Constraint 10 models the relationship between cumulative average efficiency for a time interval and the efficiency at a particular time. Constraints 11–12 are for limiting the values of variables.

The mathematical model is a mixed integer nonlinear program. The real number variables relate to the scheduling of tasks ($F_{j,t}$) and the efficiency of staff ($E_{i,j,t}$), the integer variables relate to the allocation of staff ($X_{i,j,t}$) and the outsourcing decision ($Y_{j,t}$). The nonlinear characteristic is due to the modelling of the learning effect. This model is complicated and cannot be easily solved by commercially available mathematical programming packages [12]. The genetic algorithm technique [13] is therefore applied to solve the problem.

4 Genetic algorithm

The technique of genetic algorithms (GA) aims to efficiently find a near optimum solution from an enormous solution space. Numerous applications of GAs have been published [13–15].

A genetic algorithm proceeds by progressively updating a population of candidate solutions, also called chromosomes. The population $P(t)$ is updated by creating new chromosomes by genetic operators and selecting good quality ones to form the next-generation population $P(t+1)$. A fitness function, which represents the solution quality of a chromosome, should be defined. The updating of $P(t)$ continues until predefined terminating conditions are met.

The proposed genetic algorithm for solving the formulated mathematical model is presented below.

(A) Representation of chromosomes

A chromosome is represented by a row vector $X = [X_1, X_2, \ldots, X_n]$, in which each element $X_i = [x_i^1, x_i^2, \ldots, x_i^k, \ldots, x_i^S]$ represents the staff allocation decision in period i ($1 \leq i \leq n$), and x_i^k (called gene) represents the task assignment of staff k in period i ($1 \leq x_i^k \leq m$). For example, $x_i^k = 3$ implies that staff k in period i is assigned to task 3.

Notice that staff cannot be assigned to a task j, in the periods outside its allowable working interval T_j. Therefore, the range of x_i^k, instead of $1 \leq x_i^k \leq m$, can be reduced to a set $J(x_i^k) = \{j | s_j \leq i \leq e_j\}$. Such a restriction effectively reduces the search space and in turn decreases the computation time of the proposed GA.

(B) Fitness function and linear program

The fitness function of the GA is defined by the objective function of the proposed mathematical model. To evaluate the fitness of a chromosome, we have to solve the nonlinear program by assuming that some variables have been determined, in their values, by the chromosome.

Notice that a chromosome denotes a staff allocation decision. The formulated mixed integer nonlinear program, in appearance, seems to involve three sets of decision variables: staff allocation ($X_{i,j,t}$), outsourcing decisions ($Y_{j,t}$), and project scheduling

($F_{j,t}$). A chromosome implies a staff allocation decision; that is, $X_{i,j,t}$ has been determined. Assigning $X_{i,j,t}$ by some integer values, the mathematical model becomes a linear program because constraints 4–11 can now be removed. The fitness of a chromosome, by solving a linear program, can therefore be easily evaluated.

(C) Genetic operators

Let N_P represent the number of chromosomes in $P(t)$. The initial population $P(0)$ is randomly created. Two genetic operators, crossover and mutation, are designed to create new chromosomes.

The crossover operator is introduced below. From the population $P(t)$, $N_p \times P_{cr}$ chromosomes are randomly selected and randomly paired. For each paired chromosomes, a break-point is randomly chosen to interchange some parts of the two chromosomes. For example, assume that $X_1 = [8, 4, 6, 5, : \underline{3, 7, 5}]$, $X_2 = [2, 6, 3, 1, : \underline{4, 6, 7}]$ are paired, with a chosen break-point (:). Then, the crossover operator will generate two new chromosomes $Y_1 = [8, 4, 6, 5, : \underline{4, 6, 7}]$, $Y_2 = [2, 6, 3, 1, : \underline{3, 7, 5}]$.

The mutation operator proceeds by randomly selecting $N_p \times P_{mu}$ chromosomes, where P_{mu} represent the mutation rate. For each selected chromosome, a gene x_i^k is randomly selected and randomly replaced by a new value in $J(x_i^k)$.

(D) Selection strategy

In each updating of $P(t)$, the total number of chromosomes including the new ones and existing ones is $f = N_p(1 + P_{cr} + P_{mu})$. Of these f chromosomes, N_p ones shall be selected to form $P(t+1)$. This research uses the rank-space method [14] to select chromosomes. The basic idea of the method is described below, while its detail algorithm is described in [14].

The rank space method selects chromosomes based on two criteria: solution quality and distance among chromosomes. First, chromosomes with good solution quality have higher priority to be selected. Second, chromosomes which are farther away from the selected ones have higher priority. The two priority criteria are integrated into one index to choose the chromosomes for $P(t+1)$. The purpose of applying two criteria is to avoid the GA being trapped by a local optimum solution.

5 Experiments

The proposed GA for solving the project-scheduling and staff-allocation problem has been implemented in C^{++} programming language, which calls a package ILOG CPLEX 7.5 [12] to solve the linear program. The linear program as stated is used to evaluate the fitness function of a chromosome. The computation times of the GA are also measured for different size problems.

(A) Data of experiment

In the first experiment, the decision problem involves two R&D projects, which include five tasks in total. The total work load and the allowable working interval of each task are shown in Table 1. There are ten staff with their starting efficiencies shown in Table 2, in which zero efficiency denotes that the staff cannot perform the task. The maximum allowable working days of each staff at each period are shown in Table 3. The time horizon for the decision problem is four months (four periods). Each outsourcing staff can work 25 days in each period, with a fixed efficiency (80%) in the time horizon. The outsourcing cost in each period is shown in Table 4.

In the GA, a chromosome involves 40 genes (variables), composed of four segments (periods) with ten genes (staff) in each segment. In each period, only a limited number of tasks can be performed. For example, in period 1, only tasks 1 and 4 can be performed. This constraint ensures that each task can only

Table 1. Load and allowable working interval of each task

Project ID	Task ID	Starting date	Ending date	Work load (person-day)
1	1	1/1/04	3/31/04	195
1	2	2/1/04	4/30/04	187
1	3	4/1/04	4/30/04	72
2	4	1/1/04	2/29/04	163
2	5	2/1/04	4/30/04	182

Table 2. Starting efficiency of each staff

Task ID \ Staff	1	2	3	4	5	6	7	8	9	10
1	0.5	0.4	0.9	0.7	0	0.5	0.4	0.6	0.8	0.3
2	0.3	0.7	0.7	0.6	0.7	0.8	0.4	0.8	0.4	0.5
3	0.5	0.4	0.6	0.8	0.3	0.2	0.8	0.5	0.5	0.8
4	0.2	0.8	0.5	0.5	0.8	0.5	0.4	0.9	0.7	0
5	0.5	0.4	0.8	0.4	0.5	0.7	0	0.6	0.8	0.3

Table 3. Maximum allowable working days of each staff in each period

Period \ Staff	1	2	3	4	5	6	7	8	9	10
1	30	20	30	25	28	30	23	30	25	30
2	22	30	25	30	28	22	28	26	30	28
3	20	25	20	25	23	25	24	28	25	23
4	25	28	20	25	24	25	30	26	25	25

Table 4. Outsourcing cost in each period

Task ID	Outsourcing cost ($1,000/staff-period)
1	12
2	15
3	10
4	13
5	16

be executed in its allowable working interval, and consequently reduces the search space and the computation time of the GA.

As stated, the formulated non-linear program, after assigning a particular chromosome (making a staff allocation decision), becomes a linear program. The outsourcing decision and the project scheduling decision for a particular chromosome can be easily obtained by solving the linear program. The fitness function (total outsourcing cost) of a chromosome can thus be obtained.

In the GA, parameters are set as follows: $N_P = 100$, $P_{cr} = 0.8$, $P_{mu} = 0.05$. The GA program terminates either when the best solution in $P(t)$ is sustained for over 500 generations or when $t = 100\,000$.

(B) Experiment results

The GA solves the program 20 times, each time with a different random seed. The 20 solutions are shown in Table 5. A chromosome, represented by a row in the table, denotes a staff allocation decision. For the 20 solutions, the mean outsourcing cost is $31.5K, which is equivalent to outsourcing 2.85 staff. The standard deviation of the outsourcing cost is $3.57K, which is equivalent to 0.366 staff. The 20 solutions are quite close, if measured by the number of outsourcing staff. They are different from each other by only one outsourcing staff at most.

The schedule of each task can be determined by the staff allocation decision and the outsourcing decision. For example, for the chromosome in the first row of Table 5, the schedule of each task, represented by work load percentage, is shown in Table 6.

(C) Comparison of computation time

The computation time of the GA will surely increase when the problem size increases. Table 7 shows the computation times for different sizes of problems. It takes about 1 hour and 45 min. to solve a problem with 30 staff, 15 tasks and five periods. Such a computation time seems acceptable for the project-scheduling and staff-allocation problem, which is typically planned seasonally in the real world.

Table 5. Twenty GA solutions

Chromosome (staff allocation decision)	Outsourcing cost ($1,000)
1411414411 | 1254452452 | 5211252152 | 3253252253	35
1411444441 | 1451221412 | 5251251552 | 5252223553	32
1411441411 | 5251454412 | 5252251211 | 5253252253	33
1411441411 | 5451451412 | 5252251252 | 3253253252	22
1411444441 | 5451451251 | 1252221152 | 3252253253	34
1411444441 | 1411451252 | 5211551252 | 5232552252	30
1411414441 | 5411454252 | 1251512252 | 5252253253	32
1411441441 | 5451424251 | 1251251251 | 3252253253	34
1411441441 | 5251454452 | 1211252152 | 2252253253	32
1411444411 | 1451422452 | 1251552251 | 5253522253	32
1411441441 | 1251454451 | 1211251255 | 3253252253	25
1411411441 | 1411452455 | 5251521252 | 5252523252	30
1411444441 | 1451251412 | 2251252251 | 3253252552	32
1411441411 | 5421422455 | 5251251555 | 3253222253	34
1411441411 | 5451444212 | 2251552252 | 5253253252	32
1411441411 | 5214451255 | 1211551252 | 5233253252	36
1411411411 | 5451452252 | 5511251252 | 5233223253	26
1441441411 | 1251451211 | 5215251252 | 5253253253	36
1411444441 | 2451221452 | 1251251512 | 3252253552	32
1411411441 | 1451452241 | 1251252255 | 5253253252	32

Table 6. Schedule of each task, represented by work load percentage

Task ID	1st period	2nd period	3rd period	4th period
1	56.8%	36.3%	6/9%	0%
2	0%	7.5%	45.8%	46.7%
3	0%	0%	0%	100%
4	40.3%	50.7%	0%	0%
5	25.5%	41.0%	33.5%	0%

Table 7. Computation times required to solve different sized problems

Number of tasks	Number of staff	Number of periods	Computation time
5	10	4	23 min 8 s
5	10	5	27 min 45 s
5	20	4	33 min 40 s
5	20	5	40 min 10 s
5	30	4	46 min 19 s
5	30	5	60 min 36 s
10	10	4	26 min 53 s
10	10	5	33 min 10 s
10	20	4	42 min 9 s
10	20	5	46 min 33 s
10	30	4	51 min 54 s
10	30	5	65 min 41 s
15	10	4	31 min 26 s
15	10	5	38 min 7 s
15	20	4	58 min 8 s
15	20	5	65 min 31 s
15	30	4	90 min
15	30	5	104 min 24 s

6 Concluding remarks

This paper investigates a project scheduling and staff allocation problem, including a model of the learning effect of staff, which has not been addressed in literature. The objective of this decision-making model is to minimize total outsourcing costs. A mixed integer nonlinear program is formulated to model the decision-making problem. A genetic algorithm (GA) is designed to solve the nonlinear program. In the GA, a chromosome represents a staff allocation decision. For a particular chromosome, the nonlinear program, by reducing some variables and constraints, becomes a linear program. The fitness of the chromosome is evaluated by solving the linear program.

The GA has been implemented and used to solve different sizes of problems to estimate their computation times. For a problem with 30 staff, 15 tasks and five periods in the time horizon, the GA takes about 104 min. Scheduling and staff allocation decisions are typically made only once in each season; therefore, such a computation time is acceptable to industry.

Future extension of this research would involve the consideration of project scatter effect. In a project, a task is usually performed by a team through the cooperation of team members. The scatter effect denotes that the team efficiency will decrease whenever the number of team members increases.

Acknowledgement The authors are grateful to the Intech Corp., a software customerization company in Taiwan, for providing this research problem.

References

1. Campbell GM, Diaby M (2002) Development and evaluation of an assignment heuristic for allocating cross-trained workers. Eur J Oper Res 138:9–20
2. Campbell GM (1999) Cross-utilization of workers whose capabilities differ. Manage Sci 45:722–732 Bassett M (2000) Assigning projects to optimize the utilization of employees' time and expertise. Comput Chemical Eng 24:1013–1021
3. Miller JL, Franz LS (1996) A binary-rounding heuristic for multi-period variable-task-duration assignment problems. Comput Oper Res 23(8):819–828
4. Urahama K (1994) Analog circuit for solving assignment problems. Trans IEEE on CAS-I 41(5):426–429
5. Xu HB, Wang HJ, Li CG (2002) A hybrid algorithm for the assignment problem. In: Proceedings 2002 International Conference on Machine Learning and Cybernetics, 2:881–884
6. Bassett M (2000) Assigning projects to optimize the utilization of employees' time and expertise. Comput Chemical Eng 24:1013–1021
7. Hendriks MHA, Voeten B, Kroep L (1999) Human resource allocation in a multi-project research and development environment. Int J Project Manage 17:181–188
8. Hanakawa N, Morisaki S, Matsumoto K (1998) A learning curve based simulation model for software development. In: Proceedings of the 1998 (20th) International Conference on Software Engineering, 350–359
9. Wright TP (1936) Factors affecting the cost of airplanes. J Aeronautical Sci 3:122–128
10. Yelle LE (1979) The learning curves: historical review and comprehensive survey. Decis Sci 10(2):302–328
11. Smith DB, Larsson JL (1989) The impact of learning on cost: the case of heart transplantation. Hospital Health Service Administration 34(1):85–97
12. CPLEX (2002) User's Manual, ILOG CPLEX 7.5CEnoteincomplete ref. info
13. Gen M, Cheng R (2000) Genetic algorithms and engineering optimization. Wiley, New York
14. Goldberg DE (1989) Genetic algorithm in search optimization and machine learning. Addison Wesley, New York
15. Man KF, Tang KS, Kwong S (1999) Genetic algorithms. Springer, New York
16. Winston PH (1992) Artificial intelligence. Addison-Wesley, New York