

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSI*net*

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Assignment of External Off-the-Job Training Courses to Employees Using Genetic Algorithm

Rong-Chang Chen, Ting-Tsuen Chen and Wei-Luen Fang
Department of Logistics Engineering and Management, National Taichung Institute of Technology,
No. 129, Sec. 3, Sanmin Road, Taichung, Taiwan, Republic of China

Abstract: The purpose of this study is to employ a genetic algorithm to solve the assignment problem of external off-the-job training courses. External off-the-job training offers many benefits to enterprises and thus is considered as a competitive weapon for many companies. With such understanding, planning and offering suitable training programs to employees is crucial. In this study, GA is employed as an analytical tool to allocate training courses to employees. The allocation is decided by a system which takes the employees' preferences as well as the fairness of the allocation into consideration. The use of GA in solving the problem shows that the complex problem can be well solved and suitable allocations can be made. In addition, the system constructed by our approach is also easy to use and can facilitate the allocation under many different kinds of scenarios of the company.

Key words: Genetic algorithm, off-the-job training, assignment problem, allocation, decision making

INTRODUCTION

Though enterprises are facing more and more competition today than ever before, an employee is still considered as a valuable asset and an important competitive weapon. To ensure enterprise's long-lasting competitiveness, many enterprises plan training programs and train their employees even at high costs. If companies choose not to provide suitable training, they will find it difficult to stay ahead of their competitors. Given such importance, some companies provide external off-the-job training program which is one of the most important activities in a company to enhance employees' ability and concept. Take a medium company in Taiwan for example. The company subsidizes 10-20 thousands NT dollars per month to each employee for their training. External off-the-job training program could provide at least several benefits to the company. First, off-the-job training is performed away from the workplace and thus employees can learn new concepts and new skills from employees of other companies. Second, external off-the-job training paid by the company is usually considered as welfare for employees to develop a variety of skills or to foster personal development and thus it can increase the employees' group adhesion. In addition to the benefits mentioned above, for some companies the off-the-job training can save cost when compared to internal

training. Consequently, offering employees with suitable off-the-job training is crucial for many competitive companies.

In planning off-the-job training, Human Resource (HR) Departments usually face several problems. For one thing, the company's budget of training is limited. This causes that the quantity of employees to be trained is also limited. Second, employees usually consider external off-the-job training which they like as welfare and hope that they can be approved to attend such training courses. Thus, the allocation of training courses should consider employees' preferences or needs. Another thing to be noticed is that the allocation should be fair. If the system of allocation fails to be fair, some employees will no longer consider it as welfare but a negative activity instead. To solve the problems mentioned above, this study develops a system which takes the employees' preferences and the fairness into account to allocate off-the-job courses. To consider the employees' preferences, employees are asked to identify their preferences, in an employee-course table, which takes the form of a scoring rule where a positive integer represents their priority, such as one standing for a first choice, two a second, so on up until a specified maximum number of preferences allowed, as shown in Table 1. As for the fairness of the allocation, some approaches, such as Genetic Algorithm (Coley, 1999; Gen and Cheng, 1996, 1999; Goldberg, 1989;

Holland, 1975; Mitchell, 1996; Winter *et al.*, 1995) which produces a random group of initial solutions, have provided equality by their nature. Thus, Genetic Algorithm (GA) is employed as an analytical tool to solve the aforementioned allocation problem.

Allocating limited company's resources to suit all employees' demand has long been a difficult problem, which turns into harder as the number of employees and resources provided by the company becomes large and conflicts on resource choices occur. For instance, if too many employees have chosen the same course as their first choice, as only limited employees can be allocated this course, the other employees will have to be allocated another one, say its second or third choice. Likewise, this second or third choice might be another employee's first choice and thus the HR Department again must either refuse some employees their first choices or further demote some employees to lower preferences. The problem mentioned above usually occurs in medium or large-sized companies since in these companies the assignment problem becomes more complex. To find the optimal solution of the problem as described above, different schemes (Cattrysse and Van Wassenhove, 1992; Chu and Beasley, 1997; Harper *et al.*, 2005; Irving and Manlove, 2002; Lorena and Narciso, 1996; Martello and Toth, 1990; Narciso *et al.*, 1999; Ross and Soland, 1975; Tsai *et al.*, 2008) were used in past years to solve the generalized assignment problem. Among them, GA was shown to be an effective scheme to deal with the assignment problem and it has been successfully applied to a wide of fields (Harper *et al.*, 2005; Juang *et al.*, 2007; Tsai *et al.*, 2008). In addition, matching the courses with employees is usually a difficult and time-consuming task. The company will prefer a quick and convenient tool to generating solutions. GA shows its potential to satisfy this expectation since it can produce feasible solutions within short time. Furthermore, GA is featured of supporting multi-objective decision making, thus facilitating the choice of the most suitable solution. Therefore, we will employ GA to solve the training course assignment problem in this study.

Table 1: An employee-course table with N_e employees and N_c courses

Employee	Course						N_c
	1	2	3	4	5	...	
1		1			2		
2	1	2					3
3			1				2
4	2			1			
5	1	3		5	2		4
⋮							1
N_e		1	3				2

MATERIALS AND METHODS

problem description: A typical process of assignment of external off-the-job training courses to employees is briefly shown in Fig. 1. The company usually plans to provide a series of external off-the-job training courses in the end of the year and the training is implemented in next year. The company's HR department first identifies employees' training demand and then plans the training programs. The planned training programs are mainly based on three main factors: the demand of employees for working, the demand of the company for future development and the employees' willing, needs and/or preferences. The course demand of each employee is usually different and depends on the work position or individual's needs. After identifying employees' demand, the company will review, evaluate and approve their requirements according to the company's annual budget. In this stage, the company will decide an allowed maximum number of employees being trained in a course. Once the number is decided, the employees are asked to fill in a preference table or list to show their desires. The preference is expressed with a numbering system with one indicating the first choice, two the second choice and so on. The tables are then collected and the preferences of employees are aggregated. Finally, the allocation is done according to the company's system. Since, the employees tend to resign their job after the year-end bonus is received, it is thus reasonable to decide the allocation in the beginning of next year.

The assignment problem of off-the-job training courses is complicated. The allocation system of the company is expected to satisfy the following:

- The allocation satisfies employees' preferences as much as possible

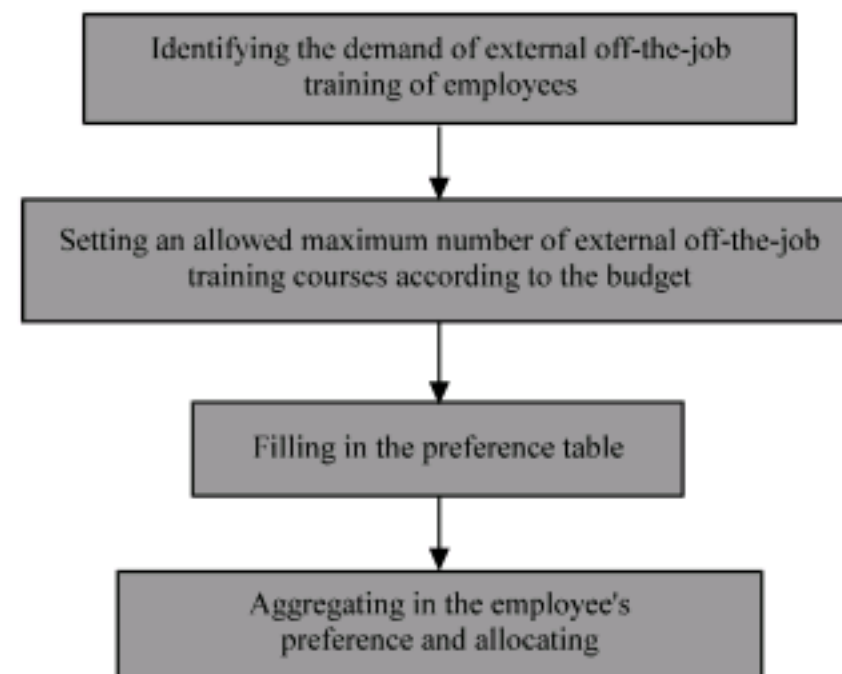


Fig. 1: A typical process of assignment of external off-the-job training courses to employees.

- The allocation is as fair as possible
- The system is flexible. It allows different employees have different weights so that excellent employees or managers can have higher priority to be assigned their top choices
- The system is easy to use

Formulation: To solve the problem mentioned earlier, some variables must first be defined. Let $D = \{1, 2, \dots, N_d\}$ be a set of employees' demand of training courses and let $S = \{1, 2, \dots, N_s\}$ be a set of courses supplied by the company. The present course assignment problem is a problem of allocating N_s courses to N_d employees' demand. Suppose that each course can allow $N_{s,max}$ employees to attend and each employee can choose at most $N_{d,max}$ courses. Thus, $N_s \times N_{s,max}$ should not less than $N_d \times N_{d,max}$ to ensure solutions exist.

For $i \in D, j \in S$, we designate c_{ij} as the preference given by demand i to being assigned training course j . This demand-supply preference matrix ($N_d \times N_s$) includes the employees' preferences indicated by a positive integer of one for a first choice, two for a second choice and the like. A demand can set at most $N_{p,max}$ choices. In general, $N_{p,max}$ is set to be 5, i.e., each employee can at most set top five priorities for each demanded course. If an employee is assigned a course outside his/her preference list, c_{ij} is assigned a suitably large penalty value P . Alternatively, we may give priority weights w_i to each employ so that we can provide some employees' demand a higher probability of being allocated their higher priority courses. In this study, constraints that needed to be satisfied consist of,

- Each demand of an employee is assigned exactly one course
- Each course can be at most assigned to $N_{s,max}$ employees

The mathematical formulation of this problem can therefore be given by:

$$\text{Minimize } F = \sum_{i=1}^{N_d} \sum_{j=1}^{N_s} w_i c_{ij} x_{ij} \quad (1)$$

subject to

$$\sum_{j \in S} x_{ij} = 1 \quad \forall i \in D \quad (2)$$

$$\sum_{i \in D} x_{ij} \leq N_{s,max} \quad \forall j \in S \quad (3)$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in D, j \in S \quad (4)$$

Where:

$$x_{ij} = \begin{cases} 1 & \text{if course } j \text{ supplied by the company is assigned to} \\ & \text{employee's demand } i, \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

and

$$c_{ij} = P \quad \text{if } c_{ij} \notin \{1, 2, \dots, N_{p,max}\} \quad \forall i \in D, j \in S \quad (6)$$

A lower value of demand's preference means that the demand has greater priority with it. For example, 1 stands for the first choice and thus it has greater priority than a value of 2 which represents the second choice. On the other hand, a higher value of priority weights, w_i , indicates that the demand has greater priority to be allocated their higher preferred course. The objective function is to be minimized.

The GA approach: A GA is a random search mechanism that attempts to simulate the biological processes of natural selection and natural evolution (Coley, 1999; Gen and Cheng, 1996, 1999; Goldberg, 1989; Holland, 1975; Mitchell, 1996). GA has been developed for solving all kinds of problems nowadays and has been widely applied. It provides a collection of feasible solutions, facilitating the decision under multiple-objective environments and allowing the decision maker to select the best alternative.

GA starts with an initial population with string structures. Each individual string in the population is called a chromosome that represents a solution to the problem. The chromosomes of GA are composed of some genes which are similar to the human beings'. The chromosomes are evaluated and iterated over many generations until the fittest one is found.

Since, GA is powerful in solving the assignment problem, we apply it to solve the present off-the-job training course assignment problem. The eight basic steps of GA approach in this study are as follows:

- Step 1:** Encoding and generating the initial population
- Step 2:** Evaluating the fitness of each chromosome
- Step 3:** Using binary tournament selection method to select two parent chromosomes with the fitness value
- Step 4:** Using a crossover operation to generate a child chromosome as an offspring
- Step 5:** Mutating to explore more solution space
- Step 6:** Evaluating the individual fitness of the offspring
- Step 7:** Comparing the fitness value of population with offspring and replace the least fit offspring
- Step 8:** Stop, if the end condition is satisfied. If not, go to step 3. The termination condition is the predefined generation number

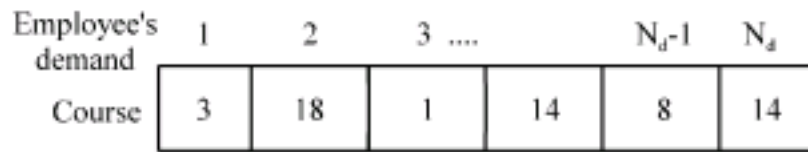


Fig. 2: Encoding of a chromosome

The detail of each step in the algorithm is explained in the following.

Step 1: Encoding and initial population: A chromosome which is composed of some genes is coded in an ordered structure. The encoding of a chromosome is shown in Fig. 2. There are N_s genes. Each gene contains a positive integer number which indicates employees that has been allocated to the course. For example, the value of the first gene is 3, indicating that the first course is assigned to the 3rd choice.

There are two popular approaches to generating the initial population: (1) the heuristic approach and (2) the random approach. In this study, we employ the random approach, in which the GA program randomly generates all the initial gene values. Due to the randomness, each employee's demand has the same probability to be the first choice.

Step 2: Evaluating the fitness function: In this step, the fitness value of the chromosome is calculated. A lower fitness value represents a more satisfying allocation and the optimal fitness value is retained. After generating the new chromosomes, we can measure the performance of the chromosomes according to the employees' preferences. The function we used to measure the chromosomes' fitness is shown as below:

$$\text{Minimize } F = \sum_{i \in D} p(c_{ij}) w_i \quad (7)$$

The fitness function contains the function $p(\cdot)$, c_{ij} and w_i , where, w_i represents the priority weights of employee's demand i and c_{ij} presents the preference indicated by demand i to be assigned course j . If demand is assigned a course which is not in the employee's preference list, the GA program will give a sufficiently large value P to c_{ij} . A common function is the square function (Harper *et al.*, 2005). In fact, the function $p(\cdot)$ can be given arbitrarily. In this study, different penalty functions are employed and their influences on the performance are investigated.

Step 3: Selection: We adopt the binary tournament selection method (Mitchell, 1996; Gen and Cheng, 1999) to select the parents from the population of chromosomes. The process of selection is explained as follows. First, two

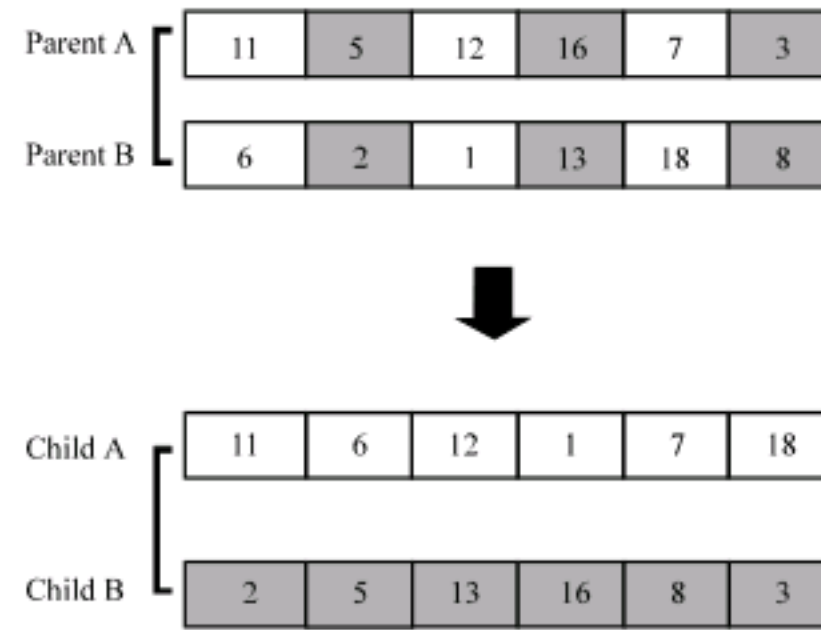


Fig. 3: Crossover is operated by separating the good and bad genes

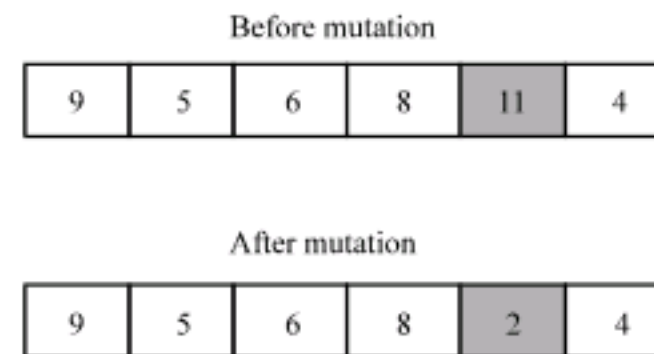


Fig. 4: Mutation operation is performed by the swap method

chromosomes are chosen randomly from the population. GA chooses the fittest one to become a parent and then it repeats the above process to select another parent. After that, the GA's combines parents A and B to generate the offspring.

Step 4: Crossover: We produce two children from two parents by separating good and bad genes. The GA program evaluates the genes in the parents' chromosome and then divides the good and bad genes according to the fitness. The crossover operation is shown in Fig. 3. The genes with the shades are the bad genes while the genes with the white color represent the good genes. After evaluating the genes, the GA produces two children, one with all the good genes and the other with the entire set of bad genes.

Step 5: Mutation: The mutation method we adopt is the change method which is shown in the Fig. 4. The gene with dark color represents the bad gene. The GA program chooses the worst gene in the chromosome to mutate. Mutation occurs when there is a change in the gene's value towards to a better gene. For example, if the demand is assigned a course which is not in the employee's preference list, the mutation operator will change the gene's value by assigning a course which is in the employee's preference list.

An unreasonable chromosome may be produced using the above change method. Therefore, the GA examines the chromosome after the mutation. For the unreasonable chromosome, the GA program corrects the gene value by assigning a new value at random. After the correction, the GA program will check the chromosome again. If the chromosome is still unreasonable, the GA program will repeat the correction until the chromosome is reasonable.

Step 6: Evaluating the individual fitness of the offspring: The step is quite easy to complete. The fitness value of each offspring is calculated by Eq. 7. Note that in Eq. 7 the penalty function can be arbitrary and the weight of employee's demand can also be given based on the company's policy. A higher priority weight can ensure the assignment is in the preference list.

Step 7: Substitution: After the crossover and mutation operations, the child chromosome is ready to replace an existing member of the population. We adopt a steady-state model (Mitchell, 1996; Coley, 1999) for the substitution. The advantage of using a steady-state model is to make sure that chromosomes can only become fitter and at the same time, keeps the diversity of GA. In this model, before replacing the existing chromosomes, the GA examines the chromosomes. The child chromosomes replace only when they are fitter and also when they are different from any other chromosomes within the population. If a child chromosome is substituted into the population, the child chromosome becomes qualified to be the parents of the next generation.

Step 8: The termination criterion: In this study, the termination criterion is the generation number defined by the user. The procedure described earlier will repeat until the number of cycles of evolution reaches the pre-assigned generation number. Once the termination criterion is satisfied, the solution is displayed in a chart. We can clearly know the course assignment distribution from the chart.

RESULTS AND DISCUSSION

To facilitate the discussion of the influences of genetic parameters, penalty function, penalty value and different priority weights on the solutions, a base case, which is based on the scenario of a famous company in Taiwan, is built up with the following settings: the quantity of employees N_e at a department is 61, the

Table 2: The influence of variation of the generation number N_G on the fitness values for the base case

N_G	OPT	C_v	Average runtime (sec)
100	1560	0.0863	13.0
200	1520	0.0666	24.9
300	1316	0.7715	36.5

$N_{pop} = 100, R_c = 0.5, R_m = 0.05$

number of courses provided by the company is 20, $N_{s,max}$ is 4 and the maximum number of courses that employees can choose, $N_{d,max}$, is 3. The maximum number of preference that an employee can set his/her choices $N_{p,max}$ is 5 and the c_{ij} penalty scores are {1, 4, 9, 16, 25} to the preferences {1, 2, 3, 4, 5}.

For brevity, we designate the generation number as N_G , the crossover rate as R_c , the mutation rate as R_m , the population size as N_{pop} and the minimum fitness value calculated from Eq. 7 as F_{min} . Since, a GA with a random initial population will produce a different result each time, we designate the average minimum fitness value F_{min} as AVG, the best minimum fitness value as OPT and the coefficient of variation of minimum fitness value F_{min} as C_v . Each case was run at least ten times to ensure that a better solution is obtained.

In this study, we used Microsoft.NET to develop the program, which can conveniently be executed on Windows platform. The system is shown in Fig. 5. In the main menu, users can easily set genetic parameters, select input and output paths, set the times of execution and preview the results. These functions are designed to facilitate the operation of the program. The input data can also be entered one after another in the dataInput menu.

A good system should be able to see the result conveniently. Two output file types can be produced in this program: .html and .txt, facilitating the result's observation. A typical output file with .html is shown in Fig. 6. In this output file, we can see the optimal fitness value, the distribution of choices, the run time and the allocation result, where, a "x" stands for a choice outside the preference list.

Effects of genetic parameters: The influence of variation of the generation number N_G on the fitness value was first tested. The generation number, N_G , was varied with 100, 200 and 300. A suitable generation number was chosen based on the optimal F_{min} , C_v and average runtime. A higher N_G can give a lower OPT, as shown in Table 2. Though a higher N_G give a better solution, it takes more time to run the program. When the generation number N_G is doubled, the execution time is nearly doubled too. Table 2 shows this trend. Thus, a compromise should be taken when both the run time and the accuracy of the solution are important.

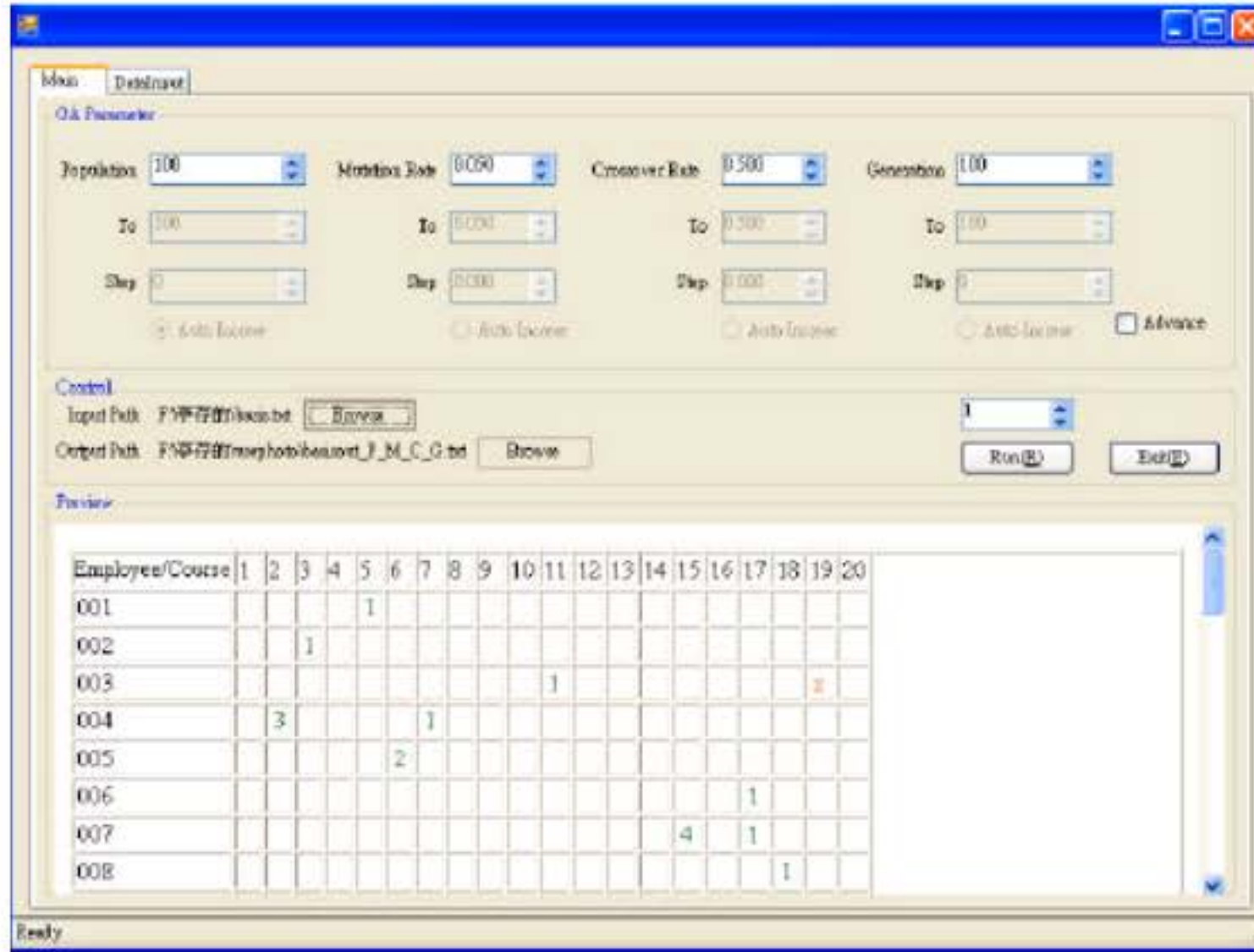


Fig. 5: The GA system developed in this study

Employee /Course	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
001				1																											
002			1				2			5																					
003											1																				x
003											1																				x
004										2	1																				
005					1	2																									
006																	1														
007							2										1														
008		1																													
009						1																									x
011						2																			1						
012					1																										
013			1																												
014			1	2																											
015																		2													
016							1																								
017																		1													
018											1														2						

Fitness = 327
 共有 17 次被安排在第1志願
 共有 8 次被安排在第2志願
 共有 1 次被安排在第5志願
 共有 3 次被安排在志願之外
 Population: 100 MutationRate: 0.05 CrossRate: 0.5 Generation: 100
 Runtime: 1.484375sec.

Fig. 6: The output .html file

Table 3: The influence of variation of the generation number N_G on the fitness values for the base case

N_G		1st	2nd	3rd	4th	5th	Out	F_{min}
100	Opt	24.0	23.0	10.0	7.0	4.0	7.0	2080.0
	Avg	21.8	20.2	5.8	4.7	2.2	6.3	1860.0
200	Opt	25.0	31.0	12.0	7.0	3.0	6.0	1780.0
	Avg	20.0	21.3	7.4	5.0	1.9	5.4	1649.3
300	Opt	26.0	29.0	10.0	7.0	4.0	6.0	1752.0
	Avg	21.1	21.3	6.8	4.8	2.1	4.9	1521.8

$N_{pop} = 100, R_c = 0.5, R_m = 0.05$

Table 4: The influence of variation of population size N_{pop} on the minimum fitness value

N_{pop}	AVG	C_v	Average runtime (sec)
100	1860.0	0.086	13.2
200	1627.5	0.094	25.1
300	1568.6	0.091	38.1

$N_G = 100, R_c = 0.5, R_m = 0.05$

The influence of variation of the generation number N_G on the choices for the base case is shown in Table 3. 1st represents a course is arranged in his/her first choice. Similarly, 2nd means the second choice and so on. If the demand of an employee is assigned a course which is not in their preference list, we will use out to designate this situation. From Table 3, we can that an increase in N_G can ensure more first choices and fewer choices outside the preference list.

The influence of population size was tested with $N_{pop} = 100, 200$ and 300 . The experimental results show that an increase in population size can give better solutions but increase the run time (Table 4). The run time is nearly doubled when the population size is doubled. There should, therefore, have a compromise between the run time and solution quality.

The influence of crossover rate on the assignment was tested with $R_c = 0.5, 0.7$ and 0.9 , respectively. The experimental results in Table 5 show that an increase in crossover rate can obtain more first choices and fewer choices outside the preference list. Table 5 also shows that most of demand can be assigned courses in their preference list, indicating the system can produce good solutions.

The effect of mutation rate on the assignment was tested with $R_m = 0.005, 0.02, 0.05$ and 0.1 , respectively. The experimental results in Table 6 show that a better mutation rate might be 0.02 , since it has a lower runtime, lower coefficient of variation and a good average minimum fitness value.

Effects of penalty value and penalty function: An employee might be very disappointed when he/her is assigned a course which is not in his/her preference list. To reduce the employees' dissatisfaction, the penalty value P can be assigned a properly large number. The influence of different penalty values on the

Table 5: The influence of crossover rate on the assignment

R_c	1st	2nd	3rd	4th	5th	Out	AVG	C_v
0.5	21.8	20.2	5.8	4.7	2.2	6.3	1860.0	0.086
0.7	21.6	19.9	6.6	5.0	2.4	5.5	1675.6	0.113
0.9	22.1	19.3	7.2	5.1	1.9	5.4	1643.2	0.072

$N_G = 100, N_{pop} = 100, R_m = 0.05, P = 250$

Table 6: The influence of mutation rate on the assignment

R_m	AVG	C_v	Average runtime (sec)
0.1	1767.5	0.083	14.2
0.02	1799.8	0.059	12.2
0.05	1860.0	0.086	13.2
0.005	1851.3	0.159	12.5

$N_G = 100, N_{pop} = 100, R_c = 0.5, P = 250$

Table 7: The influence of different penalty values on the employees which are assigned the courses out of their preference list

Penalty function	Penalty value (P)	Out		
		Optimal	Average	C_v
{1, 2, 3, 4, 5}	10	4	6.2	0.030
	25	5	5.9	0.124
	50	4	5.9	0.159
	250	5	5.8	0.125
{1 ² , 2 ² , 3 ² , 4 ² , 5 ² }	10	11	14.2	0.108
	25	6	8.3	0.156
	50	5	6.5	0.138
	250	5	6.3	0.073

$N_G = 100, N_{pop} = 100, R_c = 0.1, R_m = 0.05$

Employee / Course	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
040			1																	
041							1													
044							3													
045			1																	
046								1												
047								2												
048					2															
049				2																

Fig. 7: The output result after increasing the priority weights

number of employee's demand which is assigned a course outside their preference list is shown in Table 7.

When P is increased, the number of assigned courses outside the list is decreased. The difference is more apparent with the penalty function of $\{1^2, 2^2, 3^2, 4^2, 5^2\}$ than with the penalty function of $\{1, 2, 3, 4, 5\}$.

Influences of different weights: The GA program also can set different priority weights for employees. When the company wants a more important employee like a department director to have higher priority to be assigned their preferred courses, a higher weight can be set. As shown in Fig. 7, employees 040, 041, 044, 045, 046, 047, 048 and 049 are originally assigned their 1st, 1st, 3rd, 1st, 1st,

Employee /Course	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
040			1																	
041							1													
044					1															
045			1																	
046								1												
047								1												
048								1												
049			1																	

Fig. 8: The output result after increasing the priority weights

2nd, 2nd and 2nd choices, respectively. After increasing their weights, the output result shows that all eight employees are assigned their first choice, as shown in Fig. 8.

CONCLUSION

In this study, we have employed a genetic algorithm as an analytical tool to allocate external off-the-job training courses to employees. External off-the-job training offers many benefits to enterprises and is thus considered as a competitive weapon for many companies. We have allocated external off-the-job training courses to employees in an efficient way by using GA. The fitness function of GA in this study takes into consideration of the employees' preferences.

Experimental results show that the GA works quite well in dealing with the off-the-job training course assignment problem. In addition, the influences of variation of population size, the generation number, crossover rate and mutation rate on the solutions are tested. Different penalty values and different penalty functions are also tested and the results show that the combination of penalty function and penalty value can increase top choices and reduce choices outside the preference list.

Future works are recommended to attain the multiple objectives for the off-the-job training assignment problem. In addition, we encourage more future studies to apply the proposed approach to solve other similar assignment problems.

ACKNOWLEDGMENTS

The authors wish to express their appreciation to Mr. Jheng-Bang Chen for his help during the course of this study. We also want to send our appreciation to Miss

Chao-Hua Chen and Mr. Hsin-Yo Tsai for their assistance. This study was supported by the National Science Council under Grant No. 96-2622-E-025-003-CC3.

REFERENCES

Cattrysse, D. and L.N. Van Wassenhove, 1992. A survey of algorithms for the generalized assignment problem. *Eur. J. Operat. Res.*, 60: 260-272.

Chu, P.C. and J.E. Beasley, 1997. A genetic algorithm for the generalized assignment problem. *Comput. Operat. Res.*, 24: 17-23.

Coley, D.A., 1999. *An Introduction to Genetic Algorithms for Scientists and Engineers*. 1st Edn., World Scientific Press, Singapore.

Gen, M. and R. Cheng, 1996. *Genetic Algorithms and Engineering Design*. 1st Edn., Wiley, New York.

Gen, M. and R. Cheng, 1999. *Genetic Algorithms and Engineering Optimization*. 1st Edn., Wiley, New York.

Goldberg, D.E., 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. 1st Edn., Addison-Wesley, New York, USA., ISBN: 0201157675.

Harper, P.R., V. De Senna, I.T. Vieira and A.K. Shahani, 2005. A genetic algorithm for the project assignment problem. *Comput. Operat. Res.*, 32: 1255-1265.

Holland, J.H., 1975. *Adaptation in Natural and Artificial Systems*. 1st Edn., University of Michigan Press, Michigan, Ann, Arbor, ISBN: 0262580969.

Irving, R.W. and D.F. Manlove, 2002. The stable roommates problem with ties. *J. Algorithms*, 43: 85-105.

Juang, Y.S., S.S. Lin and H.P. Kao, 2007. An adaptive scheduling system with genetic algorithms for arranging employee training programs. *Expert. Syst. Appl.*, 33: 642-651.

Lorena, L.A.N. and M.G. Narciso, 1996. Relaxation heuristics for a generalized assignment problem. *Eur. J. Operat. Res.*, 91: 600-610.

Martello, S. and P. Toth, 1990. *Knapsack Problems: Algorithms and Computer Implementations*. 1st Edn., Wiley, New York.

Mitchell, M., 1996. *An Introduction to Genetic Algorithms*. 1st Edn., Massachusetts Institute of Technology, A Bradford Book, Cambridge, Boston, ISBN: 0262133164.

Narciso, M.G., L. Antonio and N. Lorena, 1999. Lagrangean/surrogate relaxation for generalized assignment problems. *Eur. J. Operat. Res.*, 114: 165-177.

Ross, G.T. and R.M. Soland, 1975. A branch and bound algorithm for the generalized assignment problem. *Math. Programm.*, 8: 91-103.

Tsai, H.F., T.S., Chen, R.C. Chen and J.C. Chen, 2008. Genetic algorithm for the training time assignment problem of core laboratories. *Proceedings of the 9th WSEAS International Conference on International Conference on Automation and Information*, June 24-26, World Scientific and Engineering Academy and Society Wisconsin, USA., pp: 474-479.

Winter, G., J. Periaux and M. Galan, 1995. *Genetic Algorithms in Engineering and Computer Science*. 1st Edn., Wiley, New York.