OPTIMIZING T-LEARNING COURSE SCHEDULING BASED ON GENETIC ALGORITHM IN BENEFIT-ORIENTED DATA BROADCAST ENVIRONMENTS

Yong-Ming Huang
Department of Computer Science and Information Engineering
Institute of Manufacturing Information and Systems
National Cheng Kung University

Associate Professor Dr. Chao-Chun Chen
Department of Computer Science and Information Engineering
Institute of Manufacturing Information and Systems
National Cheng Kung University
chaochun@mail.ncku.edu.tw

Ding-Chau Wang
Department of Computer Science and Information Engineering
Institute of Manufacturing Information and Systems
National Cheng Kung University

ABSTRACT
Ubiquitous learning receives much attention in these few years due to its wide spectrum of applications, such as the T-learning application. The learner can use mobile devices to watch the digital TV based course content, and thus, the T-learning provides the ubiquitous learning environment. However, in real-world data broadcast environments, the mobile learners are unable to continuously watch a digital course for a long time, because the power of devices and the user patient constrain available learning time. In this paper, we design an optimal watching mode for data broadcast T-learning environment, such that the learner can retrieve as many distinct courses as possible within given time. We then optimize the watching mode by using the genetic algorithm in order to reduce the computation cost of the optimization. Our experimental results show that genetic optimization process indeed reduces the computation cost, and still lead to a near optimal watching mode.

1. INTRODUCTION

Recent advance in digital communication technologies and wireless communication devices have led to a vast and expeditious development in e-learning applications, such as M-learning (mobile learning) (Massey, 1999), T-learning (TV-based interactive learning) (Aarreniemi-Jokipelto, 2004; Aarreniemi-Jokipelto, 2005). The learner can use mobile devices (e.g., 3G mobile phones) to watch the digital TV based course content, and thus, the T-learning is extended to be an efficient ways to solve the learning system scalability and the bandwidth limitations problem in ubiquitous learning environment. Digital TV channels are now based on the wireless data broadcast technique (Zheng, 2005), whose advantage is that the server can broadcast information to numerous clients (Atzori, 1997). This way of exploiting broadcast to disseminate information is particularly applicable to the T-learning system, since in many teaching experiences, many learners need same course content. Thus, the T-learning with data broadcast techniques would greatly reduce the bandwidth cost of the e-learning applications. In the rest of this paper, we study T-learning under data broadcast environments.

In the T-learning system, the system broadcasts the course content according to the broadcast program. In the broadcast program generation, *Broadcast Disks* algorithm is a well-known and widely discussed approach (Acharya, 1995) in the last decade. The main idea is to broadcast the data of the high access probability more frequently than those of the low access probability so that users would spend less expected time on average to access their required data.

Study on the design of the broadcast program, and its variants are numerous and successful (Hwang, 2001; Triantafillou, 2001; Vaidya, 1995). However, they simply assume that users would continuously listen to the broadcast for a long enough period to receive the needed information. But, the learner is only allowed to listen to the broadcast for limited period due to following two reasons. First, the mobile user is unable to for obtaining a period of learning in a real environment. For example, a learner watches the teaching curriculum to learn English for a short period before work. Second, a mobile device could be unable to continuously retrieving the digital TV for a long time due to the limited battery life of mobile devices (Chen, 2002). Hence, a learner cannot continuously watch the digital TV in a T-learning environment.

In the T-learning environment, an efficient watching mode should "wait" for some time slots (i.e., go into doze mode) after receiving different hot and distinct courses. Hence, the watching mode of the learner includes two phases: (i) watching time phase, and (ii) rest time phase. The watching time phase uses the active mode for a period to watch digital courses, the rest time phase uses the doze mode for a period to stop watched digital courses. The watching mode allows learners switch between the doze mode and the active mode to intelligently retrieve courses in order to save learning time or avoid increasing the battery consumption in watching digital TV. However, determining how long a learner should wait if without the knowledge of the broadcast program structure is difficult and complex. On one hand, a learner does not wish to miss a hot course while he/she is in the doze mode. On the other hand, the learner is not happy to receive the same hot course in the active periods. Therefore, scheduling the precious watching time to receive the hot courses is a critical issue to be resolved.

In this paper, our first contribution is to provide a data retrieval method for broadcast-based T-learning environment by employing our recently designed technique, *benefit-oriented data broadcasting technique* (Lin, 2008). In such environments, learners can obtain the watching mode with greatest benefit from a broadcast index, even when the learner is unaware of the broadcast program structure. In our design, the T-learning system would inform the learner the best combination of watching mode (W_t, R_t) according to learner's the start time (S_t) and the total learning time (T), where W_t is watching time and R_t is rest time. In the case that a learner follow the best watching rule to retrieve the broadcast program, the learner can watch the T-learning courses with the most beneficial manner. To properly resolve this problem, we will first define the concept of benefit from the learner's viewpoint so as to measure how good a combination (W_t, R_t) is to the learner. The combination (W_t, R_t) with greatest benefit is the one that the broadcast server would send to learners.

An intuitive approach to find the optimal (W_t, R_t) pair is to test all possible (W_t, R_t) combinations, where $1 \le W_t, R_t \le T$. However, the computation cost for achieving the optimal benefit could be enormously high. Our second contribution in this paper is to optimize (W_t, R_t) by using a genetic algorithm (GA) process (Holland, 1975). The genetic algorithm has been proven as one of the most applicable solution to the problem of permutations and combinations (Davis, 1985; Ozdamar, 1997; Hou, 1994). In this paper, we design a highly efficient genetic algorithm that dramatically reduce the computation cost and still lead to a near optimal (W_t, R_t) combination. We then conduct a set of experiments to show the performance of the genetic algorithm. Our experimental results show that the GA-based optimization indeed obtains a near optimal benefit with low computation cost.

The remaining sections of this paper are organized as follows. Section 2 introduces the system model on the T-learning technologies. In Section 3, we describe the parameter and formulate our problem. Section 4 presents our proposed genetic algorithm. Then, Section 5 shows the experiment results. Finally, we conclude the paper in Section 6.

2. BROADCAST-BASED LEARNING ARCHITECTURE

Figure 1 depicts the Digital-TV broadcast-based learning architecture for T-learning applications. The Digital-TV broadcast system includes five components: (1) broadcast content generator, (2) broadcast program generator, (3) watching mode indicator generator, (4) course broadcast, (5) course retrieval, which are presented in details as follows.

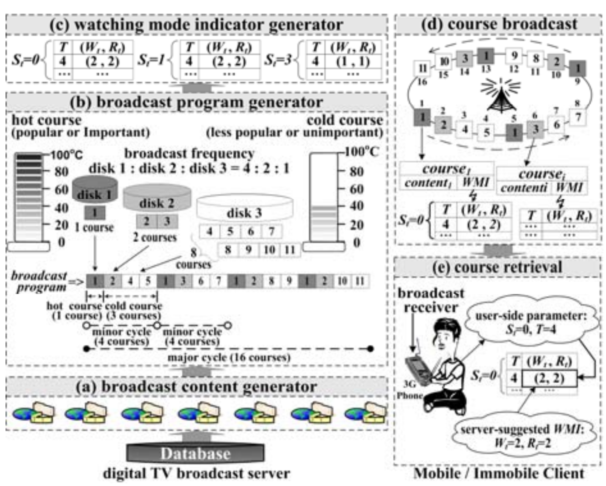


Figure 1. Broadcast-based learning architecture.

The *broadcast content generator* is used to generate broadcast contents to satisfy user demands, as shown in Figure 1(a). In real-world, a server is impossible to broadcasts all contents to users, due to too many different demands from users. Thus, the broadcast contents that satisfy each user only include the most applicable data (e.g., hot courses) to the broadcast applications (Imielinski, 1994). The *broadcast program generator*, shown in Figure 1(b), is used to generate broadcast program for the broadcast data (Acharya, 1995). The broadcast program determines the broadcast sequence and the frequency of the broadcast content. The *watching mode indicator generator* is designed to generate the best combination of watching mode (W_t, R_t) based on the start time (S_t) of the learner and the total learning time (T) of the learner. Figure 1(c) illustrates an example of the watching mode indicate. If the S_t=1 and the T=4, the learner uses the watching mode (W_t=2, R_t=2) to watch the broadcasting courses.

When the above-mentioned procedures have been completed, the digital-TV broadcast system broadcasts the course information, as shown in Figure 1(d), via wireless networks or wired networks, such as satellite, cable, or terrestrial transmitters, etc (Atzori, 1997). While course information is broadcast, the user can retrieve the course information through mobile devices (e.g., 3G phone, PDA), as shown in Figure 1(e). In order to increase receive information efficiency, many proposed methods achieve the high-performance data retrieval based on cache or prefetch techniques (Zheng, 2002).

3. PROBLEM STATEMENT

3.1. Terminologies

The parameters of the digital-TV broadcast system include *NumDisks*, $DiskSize_i$, and $DiskFreq_i$. We define the *NumDisks* as the number of disks in the broadcast model. The $DiskSize_i$ is defined as the number of courses in a disk. The $DiskFreq_i$ is defined as broadcast frequency ratio of each broadcast disk in a broadcast disk program. For example, in Figure 1(b), *NumDisks*=3, $DiskSize_2$=2, and $DiskFreq_2$=2.

The parameters of the T-learning learner include S_t, W_t, R_t, and T. S_t is defined as the learner starts to watch the digital course at time stamp S_t. W_t is defined as the time interval that the learner watched the digital-TV, and R_t is defined as the time interval that the learner stopped watching digital-TV. The overall watch and rest time of all iterations from start time S_t to the total learning time, is denoted as T (i.e., $T = \sum W_t + \sum R_t$). $(W_t, R_t)^{opt}$, named an *optimal watching mode*, has the valuable information that suggests the learner an optimal watching mode to watch courses. Using the optimal watching mode, a learner is able to retrieve the hot courses with the least time consumption.

In our design, each broadcast course has two kinds of information: one is the content of the course, the other is a small table, *watching mode indicator (WMI)*, as shown in Figure 1(d). The *WMI* is composed of two attributes: one is the total learning time T; the other is the corresponding optimal access mode expressed in the form of a pair $(W_t, R_t)^{opt}$ according to different S_t. In this example, a learner has 4 units of time (i.e., T = 4) to receive the broadcast data and now receive the first course is the $course_1$ (i.e., the S_t =0) in the program. From the downloaded course, the *WMI* suggests that (2, 2) is the best watching mode for T=4. So, for the rest of the 4 time slots the learner will achieve the greatest benefit if it receives data for the next 2 time slots and then sleep for the next 2 time slots.

3.2. Terminologies

From the learner's view point, the greatest benefit is to acquire the most valuable course from a T-learning system with the least amount of time, no matter at what time instant the learner starts to listen and the total learning time. Hence, we formally define our problem as follows.

Problem formulation:

Given a frequency-based broadcast program B, learner's total learning time T and start time S_t.

Find the $(W_t, R_t)^{opt}$ pair such that the *benefit* is maximal

---

In the formulated the problem, we observed two phenomenons. First, we observed the watch time affecting the *benefit*. Basically, the less watch time has the larger the *benefit*, because spending less time on watching course content means more energy is saved for future use. Second, we also observed broadcast frequency affects the *benefit*. This is because that the higher broadcast frequency represents the more popular course, and the more popular course contains the most applicable data (e.g., hot courses) to satisfy learners. As mentioned above, two observations are adopted to express the *benefit*.

Observation 1: The greater the total rest time, the greater the benefit, i.e., *benefit* is proportional to total rest time. This is because the learner spends less watch time on watching the needed course.

Observation 2: The greater the importance of the retrieved distinct courses, the greater the *benefit*, i.e., *benefit* is proportional to the sum of the broadcast frequency of retrieved distinct courses.

Notice, we only consider the distinct courses in Observation 2 because a received duplicate course does not offer any new information. Hence, the importance of a retrieved course can be properly represented by its access frequency. Therefore, we formally define the assessment formula of the *benefit* as

benefit = (total rest time) × (sum of the frequency of watched distinct courses)

Figure 2 shows an example to illustrate the assessment formula of the *benefit*. In this example, we studied two cases. In Case 1, the learners (i.e., learner A and learner B in Figure 2(a)) watch Digital-TV at difference S_t, and they use the same watching mode to watch courses. In Case 2, the learners (i.e., learner B and learner C in Figure 2(a)) watch Digital-TV at the same S_t, and they use the different watch mode to watch courses.

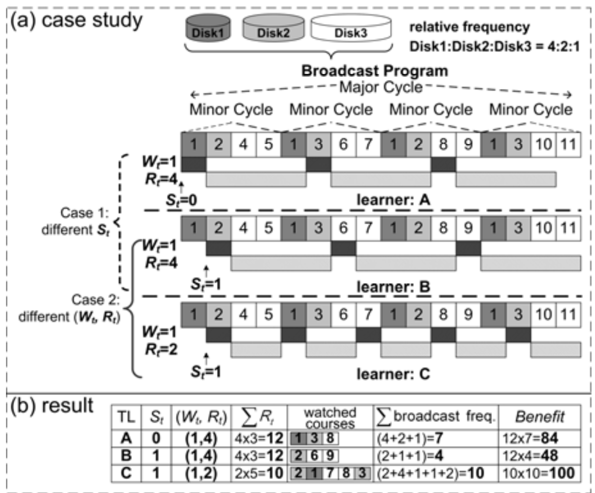


Figure 2. An example for the assessment formula of the benefit.

By Case 1, two learners A and B have received the same *WMI* (W_t, R_t)=(1,4). Assume that S_t of A is 0 and S_t of B is 1, as shown in Figure 2(a). From the Figure 2(b), their benefits obtained under these two S_t's are 84 and 48, respectively, which quite a dramatic difference is caused by only one unit of difference in their S_t. By Case 2, two learners B and C start learning at the same S_t=1. Assume that *WMI* of B is (1,4) and *WMI* of C is (1,2), as shown in Figure 2(a). From the Figure 2(b), their benefits obtained under these two *WMI*'s are 48 and 100, respectively, which quite a dramatic difference is caused by only one unit of difference in their *WMI*. From the case study of the above, the *WMI* (1,4) of the learner B is changed to the *WMI* (1,2) of learner C in Case 2, then the benefit (48) of learner B would become the benefit (100) of learner C, a very significant increase. Hence, the *WMI* should be estimated by taking S_t into consideration.

An intuitive approach to find the *WMI* is to test all possible (W_t, R_t) combinations, where $1 \leq W_t, R_t \leq T$. However, the complexity of executing such algorithm is $O(T^2 \times NumDisks)$, too high a complexity to meet our need. In this paper, we will design a highly efficient genetic optimization process to resolve this problem, and the details are presented in Section 4.

4. GENETIC OPTIMIZATION PROCESS

Genetic Algorithms (GAs) (Srinivas, 1994) are heuristic search algorithms based on the natural selection and evolution. GAs is used to simulate processes in natural system necessary for evolution. They represent an intelligent exploitation of a random search within a defined search space to an optimization problem that evolves toward better solutions. The Figure 3 illustrates the GA flow chart for optimizing *WMI* in this paper. The flow chart includes eight steps, and is presented as follows.

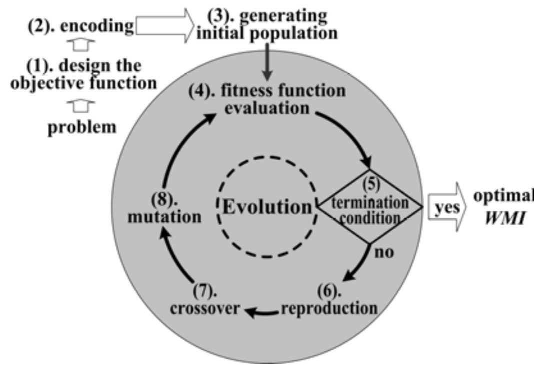


Figure 3. The GA flow chart for optimizing *WMI*.

Step1. Design the objective function
The purpose of the objective function is to determine the maximum/minimum optimization of problems under the specific conditions. Hence, we must analyze the problem and design the objective function based on the problem before implementation of GAs. In this paper, our objective function is to find out the maximum of *benefit* in Section 3.2.

Step2. Encoding
This step is to encode variables of the solution as a chromosome. The encoding methods can be classified as integer encoding, real number encoding, etc. In this paper, we use integer encoding to encode *WMI*. Due to the characteristics of the broadcasting environment, the watch time and rest time muse be the integer type. The Figure 4 illustrates an example of integer encoding in this paper. It has two variables W_t and R_t, respectively, and $1 \le W_t$, $R_t \le T$.

W_t	R_t
1	2

Figure 4. An example of integer encoding.

Step3. Generating initial population
Once a suitable representation has been decided upon for the individuals (i.e., chromosomes), it is necessary to create an initial population to serve as the starting point for GAs. In this paper, the initial population adopts uniform to create a random initial population with a uniform distribution.

Step4. Fitness function evaluation
In GAs, the fitness function evaluation is defined measures the suitability of individuals for the environment under consideration. In this paper, we design the fitness function based on the formula of the *benefit* (Section 3.2), and the fitness function is represented as follows.

Fitness function= (total rest time) × (sum of the frequency of distinct courses)

Step5. Termination condition
The GA process is repeated until a termination condition has been reached. In this paper, the termination condition is fixed number of generations, and thus Step 5 is to determine whether a conditions for termination. If the condition is satisfied termination condition, then GA outputs the optimal solution. Otherwise, the individual with higher fitness value will enter the next procedure.

Step6. Reproduction
The reproduction determines how the GA process creates children at each new generation. In this step, we use the frequently-used roulette wheel selection. The idea of roulette wheel selection simulates a roulette wheel with the area of each segment proportional to its expectation. The step uses a random number to select one of the sections with a probability equal to its area, and the selected probability of the individual *k* is shown in Eq. (1), where f_k is the fitness of the individual *k*.

$$p_k = f_k / \sum_{i=1}^{n} f_i \qquad (1)$$

Step7. Crossover
The crossover is to create a new individual, which inherits features from both parents in certain way. The common crossover includes one-point crossover, two-point crossover, and uniform crossover, and in many

studies (Syswerda, 1989; Falkenauer, 1999), the uniform is the better way to crossover. Thus, we use the uniform to crossover in this paper. The implementation of uniform crossover can be divided into two substeps. First, the system uses the random binary to create a crossover vector. Second, the system selects the genes where the vector is a 1 from the first parent, and the genes where the vector is a 0 from the second parent, and combines the genes to form the child. Figure 5 shows an example for the uniform crossover. The first parent is (1,2), the second parent is (2,1), and the crossover vector is (1,0). After the uniform crossover, the child is (1,1), as show in the figure.

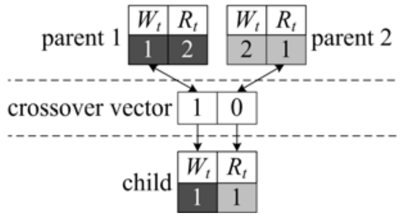


Figure 5. Illustration of uniform crossover.

Step8. Mutation
In the GA process, the mutation is to allow the algorithm to avoid local maximum/minimum by preventing the individuals become too similar to each other, thus evolution would not generate better descendants. The basic idea of mutation is to make small random changes in the individuals, which provide genetic diversity and enable GA to enlarge search space. Figure 6 illustrates an example of mutation. Assume that the second gene is the mutation point in this example. The mutation operator would set the gene to a random number with a predefined probability. After mutation, the second gene from 2 to became 5, as show in Figure 6.

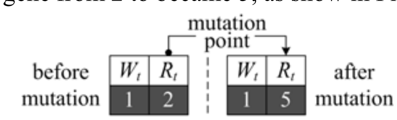


Figure 6. Illustration of mutation.

When the mutation has accomplished its process, which represents has accomplished an evolution process of a generation. Then, GA continues repeating Step 4~Step 8 for evolution of each generation until the termination condition is met. Finally, the system outputs the most fitness of the individual, and the result is the optimal *WMI*.

5. EXPERIMENT STUDY

After presenting the GA design, this section provides a detailed quantitative analysis to verify our proposed method. The system prototype designed for the experiment study is implemented by the genetic algorithm tool (GATOOL) of MATLAB (Genetic Algorithm and Direct Search Toolbox, 2008).

5.1. Experiment setup

Section 5.1 introduces parameter settings of the digital-TV broadcast system for our experiment study. The experimental digital-TV broadcast system has three broadcast disks, Disk 1, Disk 2, and Disk 3. The sizes of them are 1, 2, and 8 respectively, and the relative frequency is 4: 2: 1, as show in Table 1.

Table 1. Parameters and setting of the digital-TV broadcast system.

Parameter	Setting
NumDisks	3 (Disk 1,Disk 2, Disk 3)
$DiskSize_i$	Disk 1 = 1, Disk 2 = 2, Disk 3 = 8
$DiskFreq_i$	Disk 1 : Disk 2 : Disk 3 = 4 : 2 : 1

Table 2 depicts the parameters and setting of the learner for our experiment study. The parameters of the total learning time (*T*) are set to 4, 8, 16, and 32. The start time (S_t) are set based on the various popular degree of courses, it includes hot, medium, cold. These parameter settings are to simulate 12 scenarios for the behavior of learners, as show in Figure 7. Finally, the parameters and setting of the GA is shown in Table 3.

Table 2. Parameters and setting of the learner

Parameter	Setting
T	T=4,8,16,32
S_t	popular degree=Hot, medium, cold

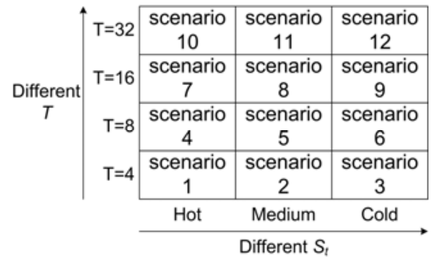


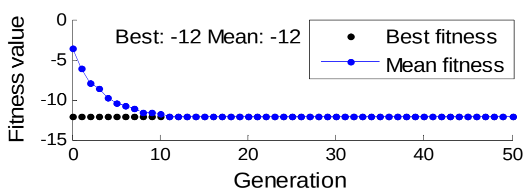
Figure 7. Categories of experiment scenarios.

Table 3. Parameters and setting of the GA (GATOOL).

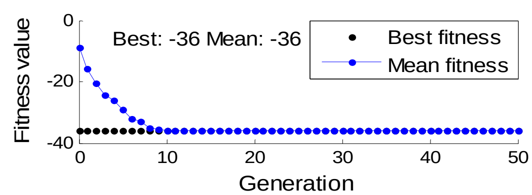
Parameter	Setting
Population size	50
Generation	50
Selection function	Roulette
Crossover function	Scattered
Mutation function	Gaussian, Scale: 0, Shrink: 1.0

5.2. Scenario-based testing

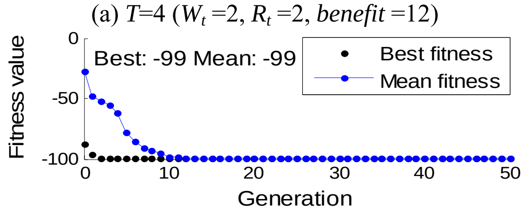
In order to evaluate the validity of GA for optimizing WMI, we design a set of scenario-based tests to observation the performance of the GA solution. Figure 8 illustrates the transformation of the individual's fitness on the evolution of each generation, and includes the WMI and benefit of each scenario. Form the result, GA obtains the WMI and benefit for each scenario after 10 iterations. Hence, GA can compute the WMI with high performance in the broadcast-based learning environment.



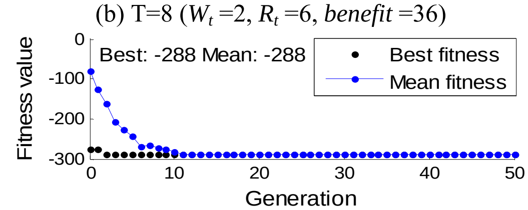
(a) T=4 (W_t =2, R_t =2, *benefit* =12)



(b) T=8 (W_t =2, R_t =6, *benefit* =36)

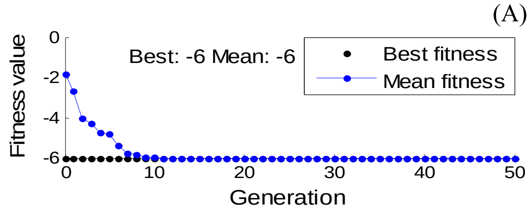


(c) T=16 (W_t =7, R_t =9, *benefit* =99)

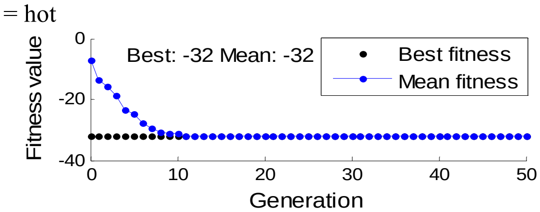


(d) T=32 (W_t =8, R_t =24, *benefit* =288)

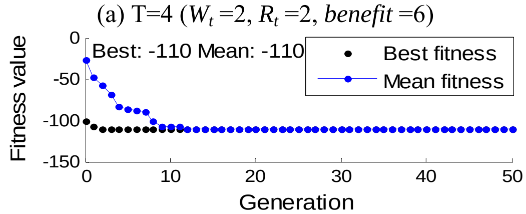
(A) S_t = hot



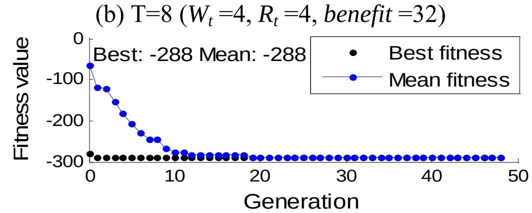
(a) T=4 (W_t =2, R_t =2, *benefit* =6)



(b) T=8 (W_t =4, R_t =4, *benefit* =32)

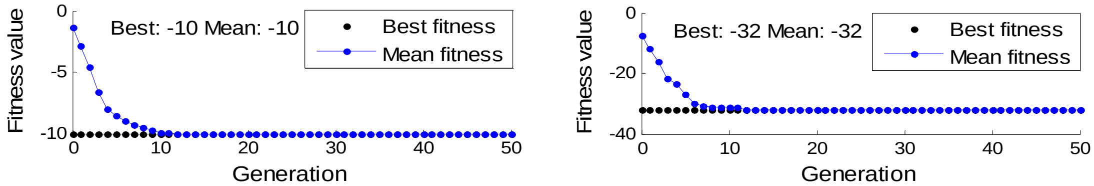


(c) T=16 (W_t =6, R_t =10, *benefit* =110)

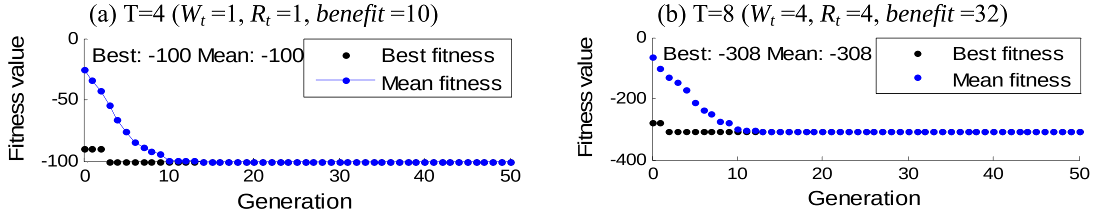


(d) T=32 (W_t =8, R_t =24, *benefit* =288)

(B) S_t = medium



(a) T=4 (W_t =1, R_t =1, *benefit* =10) (b) T=8 (W_t =4, R_t =4, *benefit* =32)



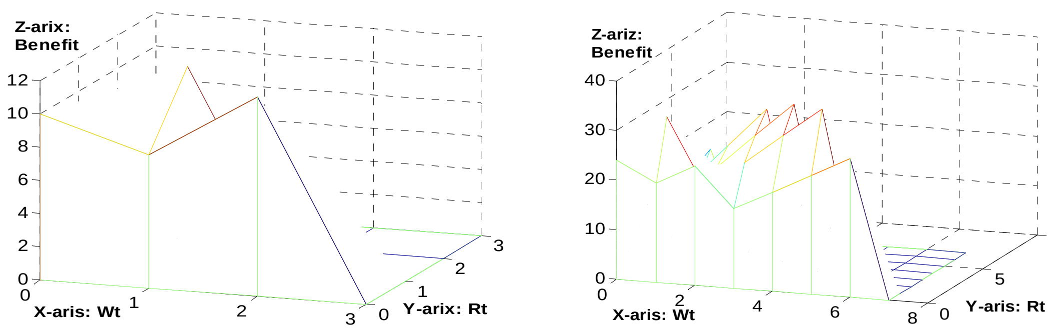
(c) T=16 (W_t =6, R_t =10, *benefit* =100) (d) T=32 (W_t =10, R_t =22, *benefit* =308)

(C) S_t = cold

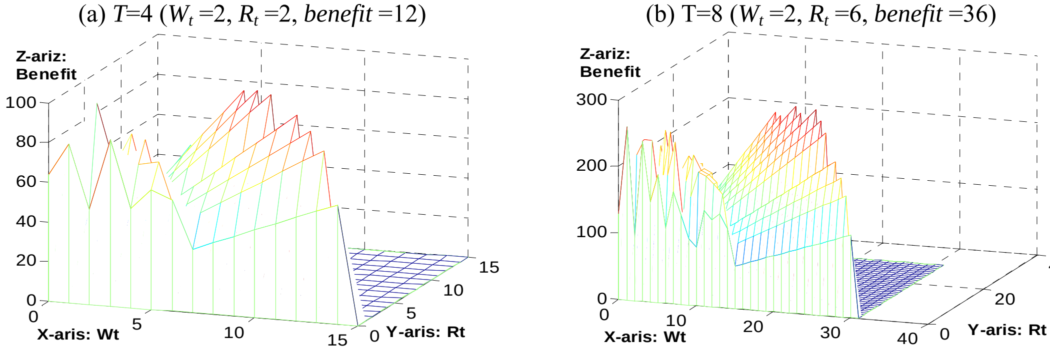
Figure 8. *WMI* and *benefit* by using GA.

5.3. Scenario-based testing

In order to evaluate the accuracy of GAs on optimize *WMI*, our second experiment verifies the *WMI* generated from GA. The steps of this experiment can be divided into two substeps. First, we implement the enumeration approach which tests all possible (W_t, R_t) combinations to find the optimal (W_t, R_t) pair. Second, we compare the *benefit* of GA and the enumeration approach to verify the performance of GA. The Figure 9 shows the *WMI* and the *benefit* by using the enumeration approach. The results show that the computation cost of the enumeration approach increases as the increasing total learning time (T), but the enumeration approach ensure to find the optimal *WMI*.

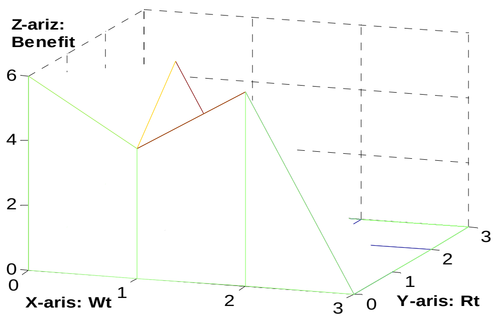


(a) T=4 (W_t =2, R_t =2, *benefit* =12) (b) T=8 (W_t =2, R_t =6, *benefit* =36)

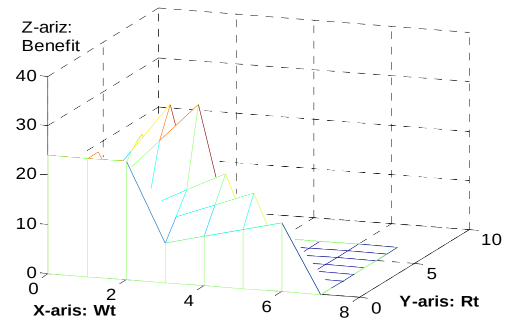


(c) T=16 (W_t =6, R_t =10, *benefit* =100) (d) T=32 (W_t =8, R_t =24, *benefit* =288)

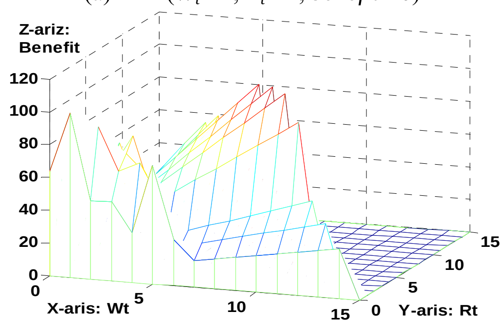
(A) S_t = hot



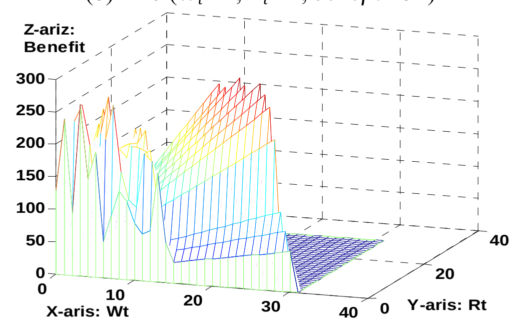
(a) T=4 (W_t =1, R_t =1, *benefit* =6)



(b) T=8 (W_t =4, R_t =4, *benefit* =32)

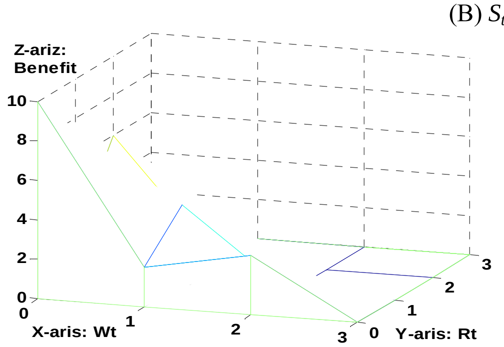


(c) T=16 (W_t =6, R_t =10, *benefit* =110)

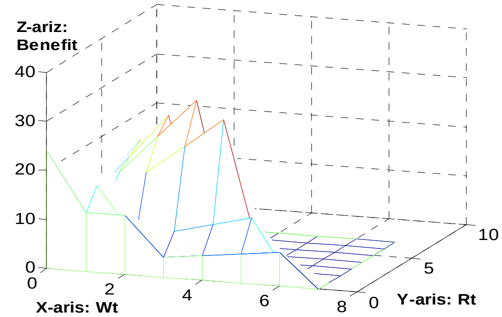


(d) T=32 (W_t =7, R_t =25, *benefit* =300)

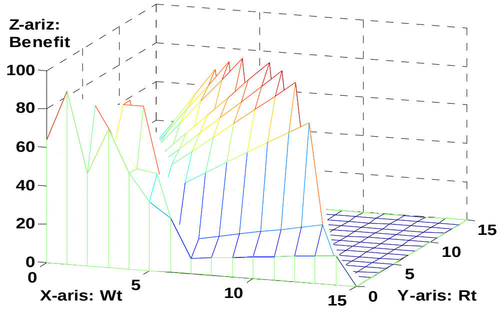
(B) S_t = medium



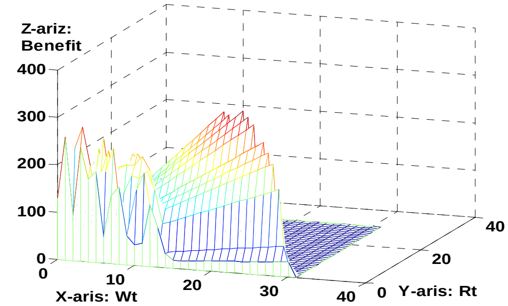
(a) T=4 (W_t =1, R_t =1, *benefit* =10)



(b) T=8 (W_t =4, R_t =4, *benefit* =32)



(c) T=16 (W_t =6, R_t =10, *benefit* =100)



(d) T=32 (W_t =10, R_t =22, *benefit* =308)

(C) S_t = cold

Figure 9. *WMI* and *benefit* by using the enumeration approach.

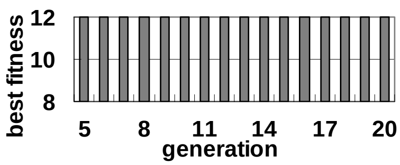
Table 4 shows the comparisons of the benefit on different methods. From the result, GA obtains almost the same results with the enumeration approach, the only two cases of GA ((S_t = hot, T=16), (S_t = medium, T=32)) did not achieve the optimal WMI. In other word, the correct rate of GA is 83%. In addition, we also observe that the difference of benefit between the GA approach and the enumeration approach for two cases ((S_t = hot, T=16), (S_t = medium, T=32)) is only less than 5%. Thus, the GA approach is quite efficient for computing WMI.

Table 4. Comparisons of the benefit on different methods

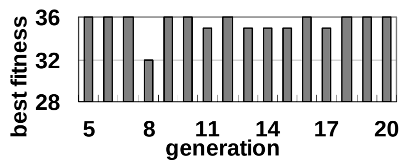
		GA	Enumeration
S_t = hot	*T*=4	*benefit* =12	*benefit* =12
	T=8	*benefit* =36	*benefit* =36
	***T*=16**	*benefit* **=99**	*benefit* **=100**
	T=32	*benefit* =280	*benefit* =288
S_t = medium	*T*=4	*benefit* =6	*benefit* =6
	T=8	*benefit* =32	*benefit* =32
	T=16	*benefit* =110	*benefit* =110
	***T*=32**	*benefit* **=288**	*benefit* **=300**
S_t = cold	*T*=4	*benefit* =10	*benefit* =10
	T=8	*benefit* =32	*benefit* =32
	T=16	*benefit* =100	*benefit* =100
	T=32	*benefit* =308	*benefit* =308

5.4. Performance tuning of GA

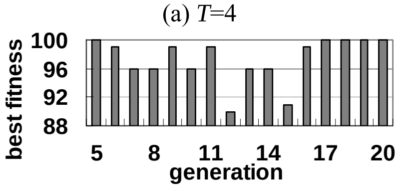
In order to obtain lower computation cost on complicate optimization computing of GA, we study the evolutionary generation that can strike a balance between significantly reducing the computation cost and obtaining a near optimal *WMI* in this experiment. From the result of the experiment 1 (Figure 8), we observe that GA can be generate the optimal *WMI* about tenth generation in the evolutionary process. Thus, the evolutionary scope is studied from 5 generations to 20 generations to find the optimal evolutionary generation. Figure 10 shows the best fitness of GA on different evolutionary generations. From the result, we observe that GA can be generate the optimal *benefit* about eighteenth generation in the evolutionary process for each scenario. Therefore, setting 20 generation for the GA can be applied to most cases to find optimal *WMI*.



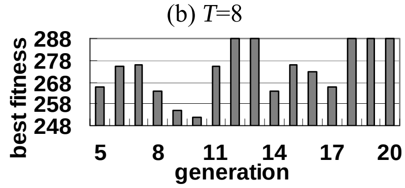
(a) *T*=4



(b) *T*=8

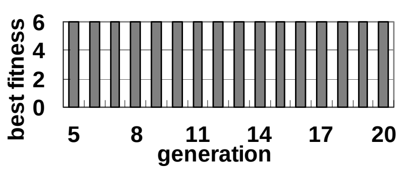


(c) *T*=16

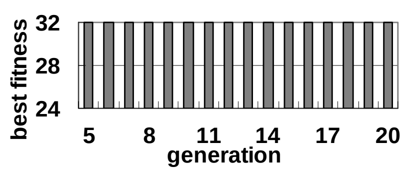


(d) *T*=32

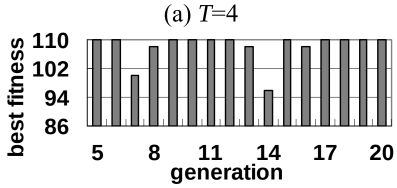
(A) S_t = hot



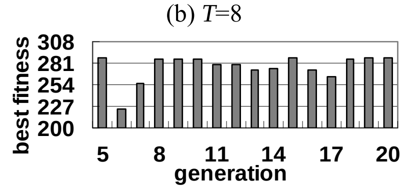
(a) *T*=4



(b) *T*=8

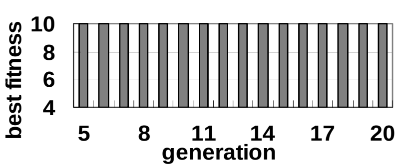


(c) *T*=16

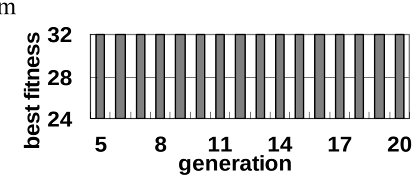


(d) *T*=32

(B) S_t = medium



(a) *T*=4



(b) *T*=8

(c) T=16 (d) T=32

(C) S_t = cold

Figure 10. The best *fitness* of the GA on different generation.

5.5. Performance tuning of GA

In the last experiment, we compare the computation cost for the two optimization methods, i.e., GA and the enumeration approach. The comparison results for these two methods are shown in Figure 11. From the result, GA obtains lower computation cost than enumeration approach. This is because that GA uses fixed number of evolutionary generation to find the optimal *WMI*. However, the computation cost of the enumeration approach will be increases as the increasing total learning time (T). Compared with the enumeration approach, GA spends quite low computation and is able to obtain optimal *WMI*.

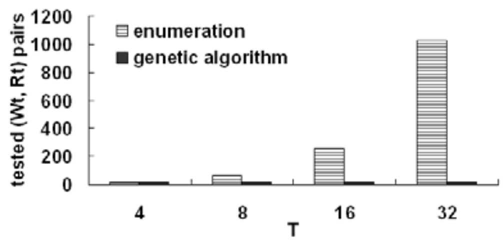


Figure 11. Comparisons of computation cost.

6. CONCLUSION

T-learning is an emerging and popular type of e-learning applications. However, previous T-learning researches do not consider the learning efficiency in limited available time. To resolve this problem, we provide an optimal watching mode $(W_t, R_t)^{opt}$ to retrieve as many distinct courses as possible in limited time. Due to the high computation complexity on computing the optimal watching mode, we optimize the watching mode by using the genetic optimization process to reduce the computation cost. Our experiment results show that the genetic algorithm can achieve very near optimal benefit, and only need quite low computation cost.

ACKNOWLEDGEMENTS

This work is supported by National Science Council of Taiwan (R.O.C.) under Grants 100-2221-E-006-267-MY2.

REFERENCES

Aarreniemi-Jokipelto, P. (2005). T-learning model for learning via digital TV. In *Proc. of the 16th Ann. Conf. on Innovation in Education for Electrical and Information Engineering (EIE'05)*, Lappeenranta, Finland.

Aarreniemi-Jokipelto, P., & Tuominen, J. (2004). Experiences with an interactive learning environment in digital TV. In *Proc. of the 4th IEEE Intl. Conf. on Advanced Learning Technologies (ICALT'04)*, Finland, 296-300.

Acharya, S., Alonso, R., Franklin, M., & Zdonik, S. (1995). Broadcast disks: data management for asymmetric communication environments. In *Proc. ACM SIGMOD Intl. Conf. on Management of Data (COMAD'95)*, San Jose, CA, 199-210.

Atzori, L., De Natale, F. G. B., Gregorio, M. D., & Giusto, D. D. (1997). Multimedia information broadcasting using digital TV channels. *IEEE Trans. on Broadcasting, 43*(3), 242-251.

Chen, I.-R., Phan, N. A., & Yen, I-L. (2002). Algorithms for supporting disconnected write operations for wireless web access in mobile client-server environments. *IEEE Trans. on Mobile Computing, 1*(1), 46-58.

Davis, L. (1985). Job shop scheduling with genetic algorithms. In *Proc. of the 1st Intl. Conf. on Genetic Algorithms (ICGA'85)*, Mahwah, NJ, USA, 136-140.

Falkenauer, E. (1999). The worth of the uniform. In *Proc. of the congress on evolutionary computation (CEC'99)*, Washington, DC, USA, 776-782.

Genetic Algorithm and Direct Search Toolbox (2008). http://www.mathworks.com/access/helpdesk/help/toolbox/gads/gads_product_page.html, Retrieved June 2011.

Holland, J. H. (1975). *Adaptation in natural and artificial systems*, Ann Arbor, MI: The University of Michigan Press.
Hou, E. S. H., Ansari, N., & Ren, H. (1994). A genetic algorithm for multiprocessor scheduling. *IEEE Trans. on Parallel and Distributed Systems, 5*(2), 113-120.
Hwang, J.-H., Cho, S., & Hwang, C.-S. (2001). Optimized scheduling on broadcast disks. In *Proc. of the 2nd Intl. Conf. on Mobile Data Management (MDM'01)*, Hong Kong, China, 91-104.
Imielinski, T., & Viswanathan, S. (1994). Adaptive wireless information systems. In *Proc. Special Interest Group on Database Systems (SIGDBS) Conf.*, Tokyo, Japan, 19-41.
Lin, L.-F., Chen, C.-C., & Lee, C. (2008). Benefit-oriented data retrieval in data broadcast environments. *Wireless Networks, 16*(1), 1-15.
Massey, A. P., Ramesh, V., & Khatri, V. (2006). Design, development, and assessment of mobile applications: the case for problem-based learning. *IEEE Trans. on Education, 49*(2), 183-192.
Ozdamar, L. (1997). A genetic algorithm approach to a general category project scheduling problem. *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 29*(1), 44-59.
Srinivas, M., & Patnaik, L. M. (1994). Genetic algorithms: a survey. *IEEE Computer, 27*(6), 17-26.
Syswerda, G. (1989). Uniform crossover in genetic algorithms. In *Proc. of the 3rd Intl. Conf. on Genetic Algorithms (ICGA'89)*, Fairfax, Virginia, USA, 2-9.
Triantafillou, P., Harpantidou, R., & Paterakis, M. (2001). High performance data broadcasting: a comprehensive system's perspective. In *Proc. of the 2nd Intl. Conf. on Mobile Data Management (MDM'01)*, Hong Kong, Chian, 79-90.
Vaidya, N. H., & Hameed, S. (1995). Scheduling data broadcast in asymmetric communication environments. *ACM Wireless Networks, 5*(3), 171-182.
Zheng, B., & Lee, D. L. (2005). Information dissemination via wireless broadcast. *Communications of the ACM, 48*(5), 105-110.
Zheng, B., Xu, J., & Lee, D. L. (2002). Cache invalidation and replacement strategies for location-dependent data in mobile environments. *IEEE Trans. on Computers, 51*(10), 1141-1153.