2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD 2017)

Mathematical Modeling and System Design of Timetabling Problem Based on Improved GA

Yanming Yang, Wanchun Gao and Yang Gao
Qingdao Campus
Naval Aeronautical and Astronautical University
Qingdao 266041, China

Abstract—**The course timetabling problem is a multi-objective combinatorial optimization problem and has been confirmed to be a multi-dimensional NP-hard problem. Timetabling process must be done for each semester frequently, which is an exhausting and time consuming task. The course timetabling problems are different for different universities. In this paper, we discuss the course timetabling problem in military academies. In general, ordinary university is to solve the problem of periodic scheduling, while the military academy is to solve the problem of irregular scheduling. The genetic algorithm (GA) is a promising scheme for solving NP-complete problems due to its high parallel, random and adaptive global searching characteristics. According to the formulate principles and characteristics of actual schedule in military academy, the course timetabling system is designed and implemented by using the improved GA. First, the characteristics of course arrangement in military academy are analyzed and the corresponding mathematical optimization model is constructed. And then, the course timetabling system is designed and implemented based on the improved GA, the main improvement of GA includes: three-dimensional code scheme, optimal preservation strategy, self-adaptive crossover probability and mutation probability design schemes. Finally, the course timetabling system is tested by the actual course data of a military academy, and the non-periodic irregular course is implemented. The feasibility and validity of the algorithm are verified by testing and analyzing the efficiency of the improved GA.**

Keywords- course timetabling problem; mathematical modeling; genetic algorithm (GA); optimal storage strategy

I. INTRODUCTION

Timetabling Problem is a special case of scheduling problems which is an NP-hard problem [1]. It is a widely studied area and many potentially useful algorithms have been offered for solving the university course timetabling problem. The goal of the university course timetabling problem (UCTP) is to find a method to allocate whole events to fix predefined timeslots and rooms, where all constraints within the problem must be satisfied [2]. Events include students, teachers and courses where resources encompass the facilities and equipment's of classrooms such as theoretical and practical rooms.

In universities, the main solution is the regular periodic timetabling problem, namely the timetable for a week time unit (week schedule), and in the military academies, to solve the non-periodic irregular timetabling problem, namely the schedule to date for the unit of time (the daily schedule). Therefore, the research and development has a strong advanced, practical and intelligent, and in line with the characteristics of military colleges and universities in the course timetabling system is imperative. The NP-hard combinatorial optimization problems can be solved using genetic algorithms when the required amount of input data is large [3]. Because of the characteristics of intelligence, parallelism and robustness in solving multi-objective combinatorial optimization problems, this paper chooses genetic algorithm to solve the problem of course arranging.

II. ANALYSIS ON THE COURSE TIMETABLING PROBLEM

A. *Influence Factors of Course Arrangement*

Course timetabling problems are mainly related to the five factors, such as classes, courses, teachers, time and classroom. These factors are not exist in isolation, they are interrelated and restrict each other.

1) *Time factors*

The concept of time involved in scheduling is: section, time, day, week, semester and school year. According to the actual situation of the university class, each class is generally 50 minutes. The duration of a lecture is usually 110 minutes (100 minutes for two lessons and 10 minutes for breaks), which is a period of 110 minutes. Due to the different circumstances of each university, the time period divided into one day is different.

2) *Course factors*

The course corresponds to the properties of the course number, course abbreviation, full course name, course belongs to the department, the class belongs to the class and classroom type requirements and so on. Each course has a teaching schedule, including the starting week, closing week and week hours and so on. In the military academies, the curriculum is divided into two basic classes and vocational courses. There are different requirements on the way of thinking and learning of the students, and if they can arrange their class time reasonably, they can acquire the knowledge and master the skills to the maximum extent under the appropriate learning intensity.

3) *Teacher* factors

In the process of arranging courses, in general, teachers and specific courses are one-to-many relationship, that is, a course must have a teacher, and a teacher can teach many courses. The corresponding attributes of teachers, number, name, teaching courses, class time and class location. Choreography curriculum should take into account the actual situation of each teacher, try to achieve personalized design.

4) Classroom factors

The classroom properties include classroom number, class name, classroom type, and number of occupants. Each classroom belongs to a teaching area. Classrooms should be of a type that meets the needs of the course, and the classroom capacity should be greater than or equal to the class size.

B. The Constraint Conditions of Course Arrangement

UCTP is about allocating lecturer to finite resources [4]. To obtain the optimal solution of the combinatorial optimization problem, we must set reasonable constraints. The constraints can be divided into hard constraints and soft constraints of these two categories. Among them, the hard constraint conditions is a measure of the feasibility of the curriculum plan is the standard, the soft constraint is a measure of the criteria for scheduling the pros and cons [5]. Based on the research and analysis of the characteristics of military academy curriculum, the constraint conditions that should be taken into account in the process of course arrangement are as follows [2,6].

1) Hard constraint conditions:

- H1: At the same time, a class cannot have more than one course.
- H2: At the same time, a teacher cannot have more than one course.
- H3: At the same time, a classroom cannot have more than one course.
- H4: The allocated classroom should accommodate more students than the number of classes.
- H5: Classroom types should match the course requirements.

2) Soft constraint conditions:

- S1: The most important courses are arranged in the most efficient way for students to learn.
- S2: In the same timetable, the distribution of the course should be basically uniform.
- S3: According to the number of students assigned to the classroom, maximize the use of resources.
- S4: Every morning the first and second classes to avoid as much as possible empty.
- S5: As far as possible to meet some of the teacher's class time and place reasonable requirements.
- S6: Theoretical and practical courses as far as possible alternating arrangements, in order to facilitate students to improve learning efficiency.
- S7: When the same teacher holds a number of courses, the course schedule should be kept at a reasonable interval so that the teacher can prepare lessons.
- S8: Some courses have a cohesive relationship between the courses, preferably when one of the courses started, another course to start.
- S9: Experimental, practical exercises, training and physical education classes are best arranged in the afternoon.

III. MATHEMATICAL MODELING OF TIMETABLING PROBLEM

A. Mathematical Description

Suppose the school has M classes, N teachers, H courses, P classrooms, Q time period, the model described in detail as follows:

- Class set is $C=\{c_1,c_2,\cdots,c_m,\cdots,c_M\}$, the number of classes is $\{x_1,x_2,\cdots,x_m,\cdots,x_M\}$.
- The teacher set is $G=\{g_1,g_2,\cdots,g_n,\cdots,g_N\}$. Each teacher corresponds to one or more courses, the corresponding number of teachers is $\{y_1,y_2,\cdots,y_n,\cdots,y_N\}$.
- The course set is $K=\{k_1,k_2,\cdots,k_h,\cdots,k_H\}$. Each course corresponds to one or more classes, one or more instructors. The number of classes for each course is $\{z_1,z_2,\cdots,z_h,\cdots,z_H\}$.
- The set of classrooms is $R=\{r_1,r_2,\cdots,r_p,\cdots,r_P\}$. The number of people in each classroom is $\{\omega_1,\omega_2,\cdots,\omega_p,\cdots,\omega_P\}$.
- The time set is $T=\{t_1,t_2,\cdots,t_q,\cdots,t_Q\}$. The Cartesian product of "time-classroom pair" is shown in equation (1):

$$\begin{aligned} S=T\cdot R=\{&(t_1,r_1),(t_1,r_2),\ldots,(t_1,r_P),(t_2,r_1),(t_2,r_2),\ldots,\\ &(t_2,r_P),\ldots,(t_Q,r_1),(t_Q,r_2),\ldots,(t_Q,r_P)\} \end{aligned} \tag{1}$$

Classes, courses, and teachers can be bundled together to form a teaching event (course tuple), since the appropriate course and teacher has already been assigned to each class in the start-up plan. The timetabling problem can be seen as finding a suitable "time-classroom pair" for each course.

B. Description of Hard Constraints.

1) In the same time section, one class can not have more than one course, as shown in equation (2):

$$\sum_{n=1}^{N}\sum_{h=1}^{H}\sum_{p=1}^{P} c_m g_n k_h r_p t_q \le 1 \tag{2}$$

where $m=1,2,\cdots,M;q=1,2,\cdots,Q$.

2) In the same time section, one teacher can not teach more than one course, as shown in equation (3):

$$\sum_{m=1}^{M}\sum_{h=1}^{H}\sum_{p=1}^{P}c_m g_n k_h r_p t_q \le 1 \tag{3}$$

where $n=1,2,\cdots,N;q=1,2,\cdots,Q$.

3) In the same time section, one classroom can not have more than one course, as shown in equation (4):

$$\sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{h=1}^{H}c_m g_n k_h r_p t_q \le 1 \tag{4}$$

where $p=1,2,\cdots,P;q=1,2,\cdots,Q$.

The total number of students (x_m) in class (c_m) can not exceed the number of students (ω_p) in classroom (r_p). That is, $x_m \le \omega_p$.

C. Description of Soft Constraints

According to the problem of course arrangement in military academies, this paper mainly considers the following 4 aspects:

1) The more important courses are arranged as far as in the time section of higher learning efficiency of students.

In this paper, a semester is set to 22 teaching weeks, divided into 6 working days per week, divided into 4 time sections. Using $\alpha_i\left(i=1,2,3,4\right)$ to show the quality of the class; using $\alpha_1=2.0$, $\alpha_2=1.5$, $\alpha_3=1.0$, $\alpha_4=0.5$ to show 4 hours a day of the quality of the class; using $\beta_j\left(j=1,2,3,4\right)$ to show the importance of the degree of course, that is, the weight. The course type includes lecture course, practice course, basic course and specialized course, whose weight is set to 1,2,3,4. As a result, the optimization objective is shown in the equation (5):

$$\max\left(f_1\right)=\sum\left(\alpha_i\times\beta_j\right) \tag{5}$$

2) As far as possible to meet the reasonable requirements of the class time and place, and so on.

According to the title of teachers is divided into assistant (Assistant Engineer), lecturer (Engineer), associate professor (Senior Engineer) and Professor (researcher) 4 grades. Coefficient X is used to show their grade, and the value is set to 1,2,3,4 in turn. The teacher satisfaction for a given teaching time is δ_j ; $\delta_j=0.0,0.5,1.0$ respectively expressed dissatisfaction, indifference and satisfaction The optimization objectives are shown in equation (6):

$$\max\left(f_2\right)=\sum\left(\chi_i\times\delta_j\right) \tag{6}$$

3) In the same course timetable, the arrangement of the course should be the basic average.

The daily course distribution uniformity of a class is shown in equation (7):

$$\mu=\frac{1}{\sqrt{\sum_{d=1}^{D}\left(e_d-e'\right)^2}} \tag{7}$$

$$e'=\frac{1}{D}\sum_{d=1}^{D}e_d \tag{8}$$

where D is the total number of days in a semester, y is the class hours in the d-th working day, and e' in equation (8) is the average class hours in class c_m per day. The optimization objective is shown in equation (9):

$$\max\left(f_3\right)=\frac{1}{M}\sqrt{\sum_{m=1}^{M}\mu_m} \tag{9}$$

where M is the total number of classes in the university.

4) According to the number of students assigned to the classroom, as far as possible to maximize the utilization of resources.

The higher the ratio of the total student number x_m in class c_m to the classroom capacity ω_p in a classroom r_p, the higher the resource utilization. The maximum value is 1. The optimization objective is shown in equation (10):

$$\max\left(f_4\right)=\sum\frac{x_m}{w_p} \tag{10}$$

IV. SOLVING FRAMEWORK OF THE COURSE TIMETABLING PROBLEMS

After establishing the goal of course arrangement and setting up the mathematical model, we will study the process of solving the course timetabling problem. The overall flow chart of the course timetabling system is shown in Figure 1. The process is divided into two parts. The first part: according to the Office of Academic Affairs issued the task of teaching the original data will be disorder (teaching tasks) to conduct a random row of class operations, generate orderly final data (course timetable); The second part: Genetic algorithm used to generate random The feasible course arranges the plan to carry on "the global optimization", algorithm process of timetabling problem based on improved GA is shown in Figure 2.

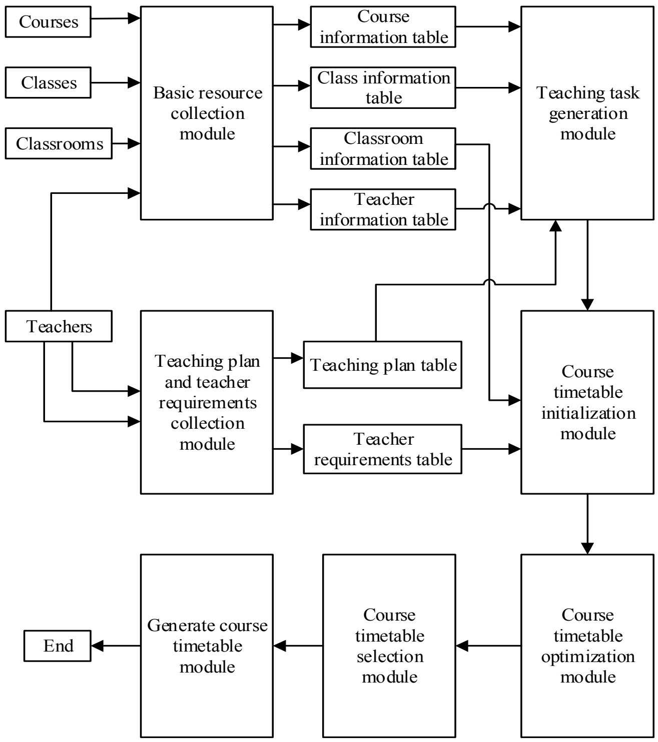

Figure 1. Flow chart of the course timetabling system.

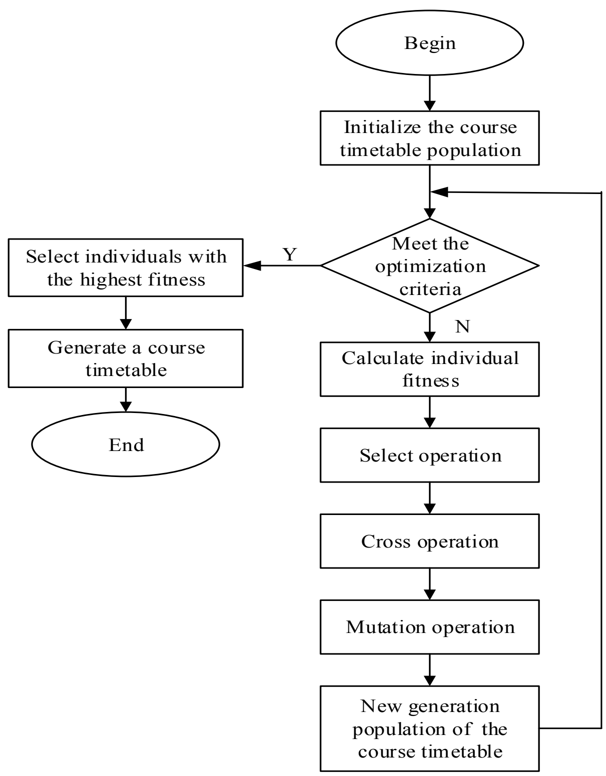

Figure 2. Algorithm process of timetabling problem based on improved GA.

V. DESIGN OF IMPROVED GENETIC ALGORITHM IN THE COURSE TIMETABLING SYSTEM

A. *Three-Dimensional Coding Scheme*

In the genetic algorithm, the encoding method directly affects the complexity of the algorithm and the efficiency of the implementation [7]. This paper adopts three-dimensional coding scheme, which is shown in Figure 3, in order to balance the various factors of timetabling problem.

In Figure 3, the X axis represents the time axis, and each time coordinate on the X axis corresponds to a time period. The Y-axis represents the classroom axis, and each coordinate interval on the Y-axis represents a classroom. There are P coordinate values (Y1~YP). Each coordinate interval on the Z-axis represents a teaching event (course tuple). In this way, three-dimensional coordinates can uniquely identify a small cube, known as the individual's gene block, also completed a class a course once the arrangement. When all of the chromosomes have been filled, it shows that a schedule has been created.

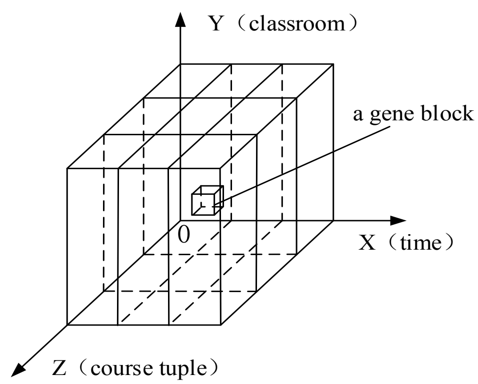

Figure 3. Three-dimensional coding of chromosome.

B. *Construct Feasible Initial Population*

According to the three-dimensional coding scheme, the initial population is generated as follows:

- Step1: first build an empty three-dimensional array, and expressed as a form of chromosome, the array of all the elements initialized to 0.
- Step2: Determine the number of times (M1) that the first one in a teaching event lists is taught in a semester, according to the size of M1 in the first layer of the three-dimensional array of chromosomes (the layer perpendicular to the Z axis) M1 position is randomly selected and assigned a value of 1.
- Step3: Continue to the next one in the teaching event lists until all the teaching events have been processed;
- Step4: Continue to build the next three-dimensional array of chromosomes, and repeat Step2 and Step3, until the number of three-dimensional array of chromosomes to reach the pre-set size of the population so far.

C. Construct the Fitness Function

According to the above course timetabling system soft constraints analysis of military academies, construct the fitness function as shown in equation (11) :

$$F = \sum_{i=1}^{4} \left(\theta_i \times f_i \right) \tag{11}$$

It is according to the target value weighted, where the value of $\theta_i \left(i = 1, 2, 3, 4 \right)$ can be defined by the management of their own, it represents the weight of course targets, respectively, the value is 4,1,2,3.

D. The Adaptive Crossover Probability and Mutation Probability

In the genetic algorithm parameters, the crossover probability and mutation probability selection is very important, they affect the performance of the algorithm. The specific design idea of this paper is to prevent the optimal individuals from crossing and mutating in the contemporary population, but to make more optimal individuals cross and mutation operation. The crossover probability and mutation probability are shown in equation (12) and (13) :

$$p_c = \begin{cases} a_1 \sin\left(\dfrac{\pi}{2} \times \dfrac{f_{\max} - f_c'}{f_{\max} - f_{avg}} \right), & f_c' \ge f_{avg} \\ a_2, & f_c' < f_{avg} \end{cases} \tag{12}$$

$$p_m = \begin{cases} a_3 \sin\left(\dfrac{\pi}{2} \times \dfrac{f_{\max} - f_m}{f_{\max} - f_{avg}} \right), & f_m \ge f_{avg} \\ a_4, & f_m < f_{avg} \end{cases} \tag{13}$$

where a_1, a_2, a_3, a_4 are constants in the interval (0,1), in general, let $a_1 = a_2, a_3 = a_4$, the specific value depends on the actual situation; $f_{\max}$ is the fitness value of the best individual in the contemporary population; f_{avg} is the fitness value of the two cross individuals, f_m is the fitness value of the individual.

E. The Optimal Storage Strategy

Roulette selection method is simple, but in the latter part of the algorithm may appear premature convergence or stagnation of the phenomenon. Therefore, this paper adopts the optimal storage strategy in the roulette selection method, the specific implementation process is as follows:

Suppose the population size is M, the parent population is $F = \left\{ x_1, x_2, \cdots, x_i, \cdots, x_M \right\}$, the initial state of the offspring population is $S = \{ \ \}$, and $f\left(x_i \right)$ is the fitness value of the individual x_i.

- Step1: calculate the adaptive degree value of each individual in a population F, and find the adaptive degree value of individual x_k, that is $f\left(x_k \right) = \max\left(f\left(x_1 \right), f\left(x_2 \right), \ldots, f\left(x_M \right) \right)$, and according to the size of the adaptive degree value to arrange order of all individuals in a population, get $F_1 = \left\{ y_1, y_2, \cdots, y_i, \cdots, y_M \right\}$;
- Step2: calculate the sum of all individuals in the population adaptive degree $\sum_{i=1}^{M} f\left(y_i \right)$;
- Step3: calculate the probability of y_k is selected individuals in the population $P_{y_k} = \dfrac{f\left(y_k \right)}{\sum_{i=1}^{M} f\left(y_i \right)}$, $k = 1, 2, \ldots, M$;
- Step4: calculation of individuals in a population is selected in the roulette wheel, the cumulative probability $PP_k = \sum_{j=1}^{k} P_{y_j}$, $k = 1, 2, \ldots, M$;
- Step5: rotation M time table:

  S1: generate M uniform pseudo-random numbers ξ in [0, 1] interval, if $\xi \le PP_1$, the individual y_1 is selected; if $PP_{k-1} < \xi \le PP_k \left(2 \le k \le M \right)$, the individual y_k is selected;

  S2: Statistics of the x value of each interval: $r_1, r_2, \ldots, r_i, \ldots, r_N$, r_i is the number of pseudo-random numbers corresponding to the region i, and then calculated the maximum value of r : $r_j = \max\left(r_1, r_2, \ldots, r_N \right)$, the corresponding individual y_j from the r_j, as the current round of individual X_i;

  S3: Add X_i to the population S;

  S4: Repeat s1, s2 and s3 until the number of individuals in the population S reaches the population size M.
- Step6: in the population S to find the lowest adaptive degree value of individual z_t, expressed as $f\left(z_t \right) = \max\left(f\left(z_1 \right), f\left(z_2 \right), \ldots, f\left(z_T \right) \right)$;
- Step7: the highest adaptive degree value of individual x_k in the population F instead of the individual z_t in the lowest fitness value in the population S;

- Step8: all individuals will be selected to save, and return.

VI. Testing of the Course Timetabling System

A. Population Scale Testing

Test groups 1 to 3 were tested for population sizes of 30, 50 and 100. In each group of tests were randomly selected 10 times the results of operations to record. The time of the approximate optimal solution obtained by the test can be used as the standard to measure the efficiency of the algorithm. The results of the three groups are shown in Table 1.

TABLE I. EFFECT OF POPULATION SIZE ON ALGORITHM EFFICIENCY (UNIT: SECOND)

Number of Tests	**Test Group 1 Population Size 30**		**Test Group 2 Population Size 50**		**Test Group 3 Population Size 100**	
	Time	***Fitness value of near-optimal solution***	***Time***	***Fitness value of near-optimal solution***	***Time***	***Fitness value of near-optimal solution***
1	271.6	1549	315.4	1610	721.6	1702
2	257.9	1491	334.2	1599	733.1	1598
3	302.8	1583	371.5	1583	763.0	1662
4	284.2	1568	352.1	1627	752.8	1678
5	265.3	1531	326.7	1593	708.4	1582
6	276.0	1596	340.1	1614	736.1	1636
7	263.8	1574	338.6	1589	751.4	1699
8	281.1	1469	357.4	1623	732.7	1684
9	271.4	1551	308.2	1634	741.4	1622
10	259.3	1528	319.3	1576	709.1	1649
average	273.3	1544	336.3	1604.8	734.1	1651.2

It can be seen from Table 1 that the larger the population size, the longer the running time of the algorithm, and the higher the fitness value of the near-optimal solution. But the population size cannot be increased unrestricted, otherwise it will lead to a sharp increase in algorithm running time, efficiency greatly reduced. The system is proved to be efficient when the population size is between 50 and 100 by several tests. Therefore, the population size is set to be 100.

B. Adaptive Parameter Testing

For adaptive parameters a_1、a_2、a_3、a_4 to take different values respectively, we test the fitness of the near optimal solution and the number of iterations of the algorithm. Some results are shown in Table 2.

It can be seen from Table 2 that the different values of the adaptive parameters have a certain degree of influence on the fitness value of the near-optimal solution and the number of iterations of the algorithm. However, no matter what the value of adaptive parameters, the population will have a significant evolutionary trend.

TABLE II. FITNESS VALUE OF NEAR-OPTIMAL SOLUTION AND ITERATION TIMES OF ALGORITHM UNDER DIFFERENT ADAPTIVE PARAMETERS

a_1	a_2	a_3	a_4	**Fitness Value of Near-optimal Solution**	**Iteration Times**
0.5	0.5	0.8	0.8	1609	311
0.9	0.9	0.6	0.6	1618	326
0.9	0.9	0.7	0.7	1568	289
1.0	1.0	0.6	0.6	1585	301
1.0	1.0	0.7	0.7	1663	324

C. Efficiency Testing

In this paper, the selection operation is to combine the optimal preservation strategy in the classic roulette selection method, and compare with the running efficiency of the references [8] is not integrated into the optimal preservation strategy. The results of 10 runs are randomly selected and the results are shown in Figure 4. According to Figure 4, the improved GA is more efficient than the reference [8].

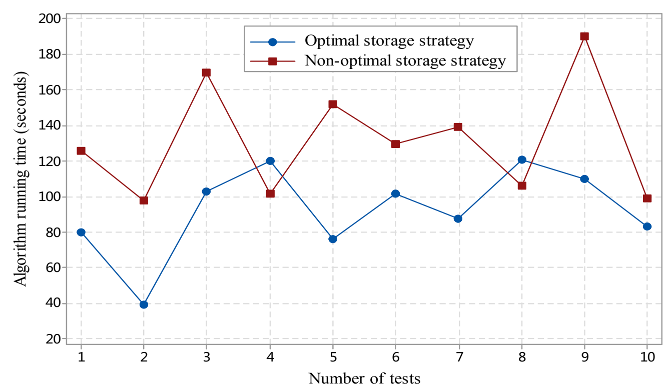

Figure 4. Comparison of algorithms without integration of optimal storage strategy and optimal storage strategy.

VII. Conclusions

Genetic algorithm has been successfully applied to develop the optimal timetable [9]. According to the characteristics of course timetabling system in military academy, this paper analyzes the influence factors and restraint conditions of course timetabling, and constructs the corresponding mathematical optimization model. And an intelligent course timetabling system of military academy based on the improved GA is designed and implemented. The test results show that the proposed improved GA can effectively solve the problem of arranging course in military academies. According to the result of the course arrangement, the course distribution is more uniform, and can realize the non-periodic irregular timetabling problem. The system has been applied for course arrangement in many military academies.

REFERENCES

[1] Kumar, Sanjeev. "Solving University Course Timetabling Problem Using AHP Method." Social Science Electronic Publishing (2016).

[2] Babaei, Hamed, J. Karimpour, and A. Hadidi. "A survey of approaches for university course timetabling problem." Computers & Industrial Engineering 86.C(2015):43-59.

[3] Diveev, A. I., and O. V. Bobr. "Variational Genetic Algorithm for NP-hard Scheduling Problem Solution." Procedia Computer Science 103(2017):52-58.

[4] Ilyas, Rafia, and Z. Iqbal. "Study of Hybrid Approaches used for University Course Timetable Problem (UCTP)." The, IEEE Conference on Industrial Electronics and Applications IEEE, 2015.

[5] Sun, Tong, and Q. Q. Guo. "Simulation on UTP Based on New Immune GA." Computer Simulation 29.2(2012):386-391.

[6] Yadollahi, Mehdi, et al. "Solving university course Timetabling problem using genetic algorithm." Awerprocedia Information Technology & Computer Science 2012.

[7] LI Ming, WANG Yan, NIAN Fuzhong. *Intelligent Information Processing and Application*. Beijing: Electronic Industry Press, 2010: 106-107.

[8] WANG Nuo. Optimization of Course Arrangement System Based on Genetic Algorithm[D]. Hebei University of Science and Technology, 2011.

[9] Kraleva, Radoslava Stankova, and V. Kralev. "A local search algorithm based on chromatic classes for university course timetabling problem." 7.28(2017):1-7.