

Research on Solving Timetabling Problem Based on Improved Retrospective Algorithm

Jie Song

School of Information Technology, Kunming University, Kunming, China

1353257038@qq.com

Keywords: Timetabling; Retrospective algorithm; Fitness function; Hard constraint; Soft constraint.

Abstract. The algorithms for timetabling have been widely studied but there are no perfect solutions for it yet. Through investigating and comparing the existing algorithms, it has been found that these algorithms can be classified into two categories designated as constructed algorithms and modern algorithms, which show the individual advantages and shortages. In this improved algorithm, the genetic algorithm and some useful rules have been introduced into the retrospective algorithm. The practical application proves that the improved retrospective algorithm is especially suitable as a timetabling system.

Introduction

Timetable is to solve the problem caused by the resources competition from time and space. In the 20th century, mid-70s, an Americans named S.VEN demonstrated that timetable is a kind of NP complete classification problem[1]. At the same time, the timetable problem had its own inherent mathematical model, namely that timetable problem exists solution, and the solution can be found. Until today, the research activities on timetable are still very active at home and abroad[2]. This paper draws on the advantages of genetic algorithm, the anthropomorphic strategy and the evaluation, punishment mechanism in genetic algorithm are introduced in the retrospective algorithm. The fitness obtained by using the proposed method is ideal, and the timetable can be obtained quickly, which is very suitable for the requirements of practical applications.

The description of timetable

The main task of timetable is to arrange the classes, teachers, curriculum and classroom time within a week, which can not exist conflicts. With the viewpoint of the set, the timetable problem includes the requirement set E , the resource set S , the constraint set C . The ultimate goal is find the mapping from S to C for all elements in E .

Combined with the actual demand, the requirement set refers to the curriculum set $L = \{1, 2, 3, \dots\}$, where the element are the number of the curriculum. We can require the concrete curriculum information from the number, such as total hours of week and teaching methods.

Resource set refers to the time set T and classroom set R . $R = \{1, 2, 3, \dots\}$, where the element represents the classroom number. We can requirement the concrete information about classroom from the number, such as the number of the classroom can be accommodate and the teaching methods (multimedia classrooms, gymnasiums, etc.). $T = \{1, 2, \dots, 20\}$, it represents that one day

contains four large classes (8 hours), and one week has 20 time sheet.

The constraint set contains two types of regulation: one is hard constraint, the schedule is effective only it satisfies the hard constraint; The other is soft constraint, the schedule adjusted by soft constraint is more scientific and reasonable[3].

The hard constraints include the following two:

- (1) Some special courses should be in particular classrooms. For example, the mechanical drawing course should be taught in drawing room.
- (2) Some courses should be taught at the specified time. For example, some courses are taught by part-time teachers, thus , the course should be arranged in the “part time”.

The soft constraints include the following three:

- (1) For the courses taught more than one time in a week, it should be arranged in the next day. For example, the higher mathematics course are taught two times in a week. It is more scientific to arrange it on Monday and Wednesday.
- (2) It should try to average course load of the week.
- (3) The different sections of the same course should be arranged in the classroom.

The application of improved retrospective algorithm in timetabling

Retrospective algorithm is the process of searching a state tree. Specific to the timetable problem, it means that all the courses in a timetable are arranged with a certain order. When there exists a course that has no solution, namely the time sheet and classroom resources allocates failure. The allocation of previous course is modified. Then re-allocate resources to the curriculum. The specific operation are shown as follows:

Coding.

The coding problem should be solved at first. The coding matrix is used in this paper. The matrix X represents the possible timetable of a class. The value of row in X refers to the course number and the value of column in X refers to the time sheet number. If a particular course of i course is scheduled on j time slice, the value of x_{ij} is 1, otherwise the value is 0. For example, a certain schedule has four courses, which contains 8 time slices. Each of the course have two lessons (one large lesson). The lesson 1 is arranged on time slice 3, the lesson 2 is arranged on time slice 4 and so on. The concrete coding is shown in Fig. 1. It can be seen that, this schedule is not valid, because two courses are arranged in time slice 3, which is inn violation of a hard constraint.

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Fig. 1 Code example

A auxiliary matrix Y is introduced in the algorithm, the value of the row refers the the number of available classrooms. The first column records the capacity of the classroom, the second column records classroom teaching methods. Starting from the third column, each column represents the available time slice in a week.

Fitness evaluation function.

The fitness is the goal of schedule in line with the degree of each soft constraint. This paper numbered all the soft constraints as $1, 2, 3, \dots, m$. Set d as a feasible solution, $l \in L$ as a course.

Define the following function:

$$g(l, d) = \sum_{i=1}^m p_i w_i \quad (1)$$

Where $g(l, d)$ represents the dissatisfaction of solution d from course l ; $w_i \in [1, 1000]$ represents the weight corresponding to the soft constraint i . If the course l in solution d violates the constraint i , $p_i = 1$, otherwise $p_i = 0$ ($1 \leq i \leq m$).

The fitness function is defined as follows:

$$f(d) = 1/1 + \sum_{l \in L} f(l, d) \quad (2)$$

According to this fitness function, the optimal solution d_{\max} can be defined, which satisfies:

$$f(d_{\max}) = \max_{d \text{ is feasible solution}} f(d) \quad (3)$$

As for the value of w_i , it is determined by the importance in accordance with the constraints.

The important conditions is given greater weight. Optimal solution d_{\max} can not be obtained in practical applications. Thus, this method is to get the premise of guaranteeing speeds as large as possible so that obtaining the fitness function solution $f(d)$.

Optimization principle.

For the time slice allocated initialization, it can be obtained in accordance with a time of day slice occupancy rate of five classification queue (up to a maximum level at the day of week usage time slice are different, which is divided into 5) according to the matrix X , and in the process of timetable, the queue is updated according to the changes of X (shown in Fig. 2). When the courses are allocated the initial time slices, the time slice are selected from the low occupancy level. If the current level is no queue enough time slice, it is selected from the high level.

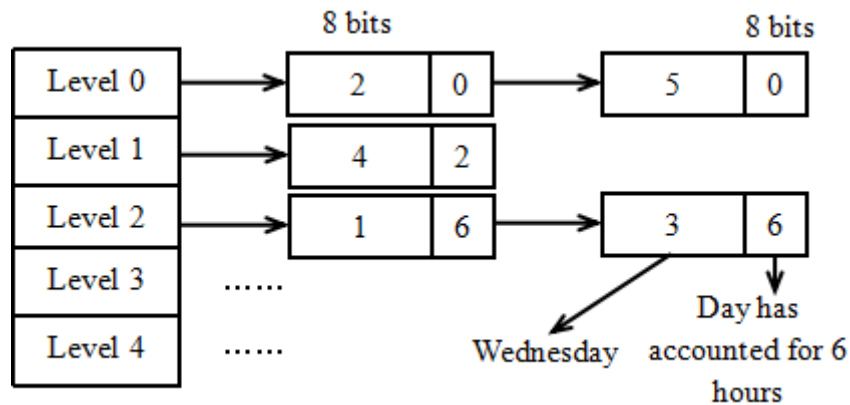


Fig. 2 Five class of time slots queue

The distribution according to need strategy.

In the allocation process, the current has successfully allocate course set L_{cur} . And it form 1 portion of the timetable, which is called the course $l \in L - L_{cur}$ considered to allocate in the current

curriculum d_{cur} . Set ds as the combination of the feasible time slices and classroom allocated to l . According to Eq. (1), the dissatisfaction under d_{cur} is defined as follows:

$$g(l, ds, d_{cur}) = \sum_{i=1}^m p_i w_i \quad (4)$$

The definition of w_i is the same with Eq. (1), p_i represents that in the current section schedule d_{cur} , if ds is allocated to l , whether the arrangement of l satisfy the constraint i . If it is satisfied, $p_i = 0$. Otherwise, $p_i = 1$.

The distribution according to need strategy refers to the pre-specified acceptable fitness. According to Eq. (2), the system can calculate the dissatisfaction g_{req} . Each step in the distribution should be guaranteed that the value of Eq. (4) is less than g_{req} . It can be seen that, the pre-specified acceptable fitness is important. And its reasonable range is $[0.015, 0.060]$.

The optimal allocation strategy.

The optimal allocation strategy is to satisfy:

$$g(l, ds, d_{cur}) = \min g(l, d_{match}, d_{cur}) \quad (5)$$

The advantage of this strategy is the ability to get the optimum solution, but the solution speed is significantly slower than the on-demand strategy. These two strategies can be selected by the user according to their own needs.

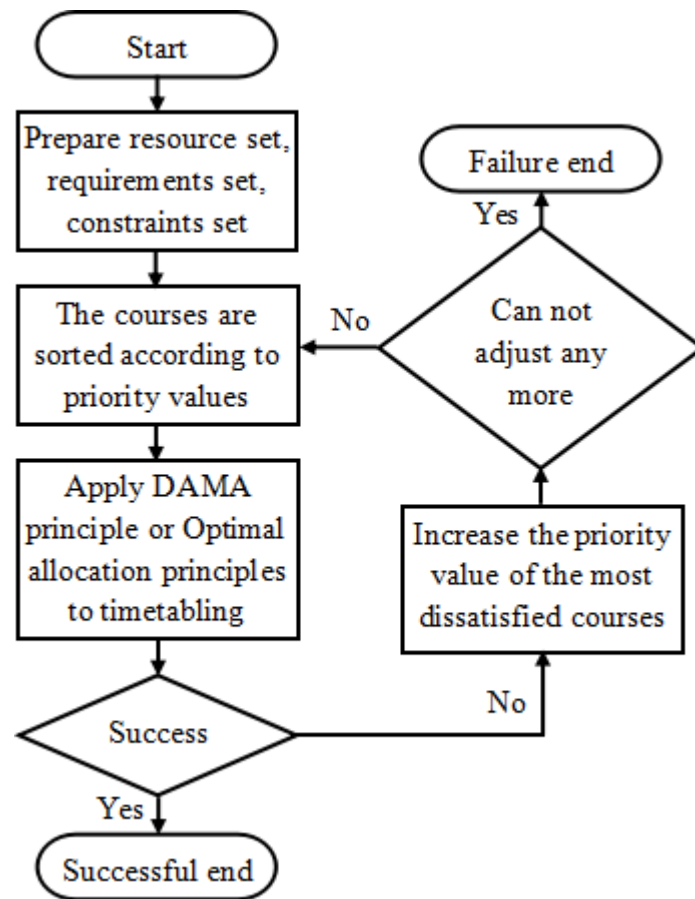


Fig. 3 Flow chart

The dynamic priority strategy.

Specific strategies is that, according to the distribution according to need strategy or The optimal allocation strategy, the system allocates schedule. When a certain course l can not get the solution. The priority of l will be improved. After sorting, the next round is processed by the two strategies. So that after a few rounds later, the system can find the order to get more optimal solution, which seems to introduce the learning function to the system, eliminating the algorithm curriculum priorities sensitivity.

The above strategies is integrated into the retrospective algorithm, and the flow chart of the whole algorithm is shown in Fig. 3.

Algorithm Evaluation.

In practical applications, the most impact on timetable results is the ratio $\alpha(\alpha \geq 1)$ between the available time slice and the total hours of a week. The most demanding condition is that α is equal to 1. All the available time slice will be occupied. If this timetable system can complete such task, the condition that α is more than 1 will be also solved. Therefore, in the practical application of this system, the usage of the optimal allocation strategy is analyzed. Fig . 4 shows the actual lesson task analysis.

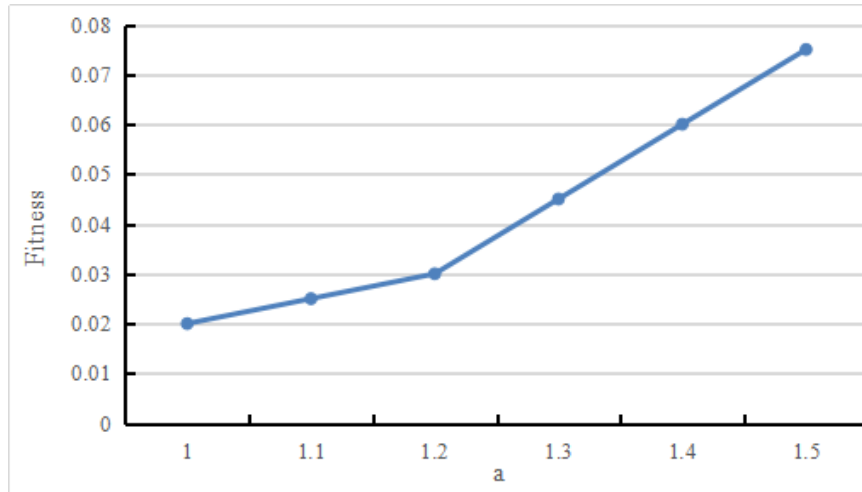


Fig. 4 Evaluation results

Conclusions

The proposed system has been tested and practically applied. Timetable effect is ideal and of high efficiency. Even with optimal allocation strategy, it can also obtain the results quickly. Processing time is generally less than 2 second, which can meet customer demand. In automatic timetable process, if the solution can not be found, the system will automatically reduce the parameters, and then solved, and that the system has a strong interactive, which can combine with artificial timetable.

References

- [1] Guo Fangming, Zhong Li. Arranging model of university timetable based on reinforcement learning. Computer Engineering and Design, Volume 11 (2003), P. 125-128.
- [2] Gao xima, Zhang Ping. Timetabling system based on kernel algorithm. Transaction of Nanyang Teacher training School, Volume 12 (2003), P. 55-58.
- [3] Tang Yong, Tang Xuefei, Wang Ling. A timetabling system based on genetic algorithms. Computer Applications, Volume 10 (2002), P. 93-94.