World Transactions on Engineering and Technology Education
Vol.13, No.3, 2015

Solving a timetabling problem with an artificial bee colony algorithm

Tinggui Chen & Chonghuan Xu

Zhejiang Gongshang University
Hangzhou, Zhejiang, People's Republic of China

ABSTRACT: An efficient arrangement of teaching resources, including the allocation of times and classrooms for university courses, through the process of timetabling, is a common problem faced by many universities in China. In order to allocate the resources in a timetabling problem to seek a reasonable solution, a method based on a discrete artificial bee colony algorithm has been devised and is presented in this article. It improves, upon the basic artificial bee colony algorithm, and was tested to verify that it can arrange all the resources reasonably in the process of timetabling. In addition, an adaptive local optimisation strategy improves the convergence and efficiency of the timetabling.

INTRODUCTION

The timetabling problem is a typical combinatorial optimisation problem, which was proved to be an NP (nondeterministic polynomial) complete problem in the early 1970s [1]. Research on this problem has been highly active since the 1960s and many algorithms and techniques have been introduced. The early research was mainly based on heuristic algorithms and manual simulations. These algorithms lack intelligence, have limited performance, and it is difficult to handle a large number of scheduling tasks. The timetabling problem, being NP-complete, means that algorithms used to solve NP-complete problems can be applied. These include the greedy algorithm, the backtracking method and the graph algorithm [2-5]. Indeed, these have been used successfully.

With the development of computers, a new type of algorithm has been developed in recent years, viz. intelligent optimisation algorithms [6]. These have been used to simulate human intelligence, biological evolution, physical and chemical phenomena, and animal swarm behaviour. The algorithms include the genetic algorithm, neural network algorithm, immune algorithm, simulation annealing algorithm, tabu search and the swarm intelligence algorithm. Intelligent optimisation has been applied to solving the timetabling problem.

An artificial bee colony (ABC) algorithm inspired by the intelligent foraging behaviour of a honey bee swarm was proposed by Karaboga [7]. It has been used widely in complex optimisation problems [8][9], such as teaching evaluation [10] and flow shop scheduling [11]. Its principle and mechanism are simple, with few adjustment parameters. The algorithm is easy to implement. The traditional ABC algorithm can be applied only to continuous problems; for discrete problems, a discrete ABC algorithm can be used.

Using the discrete ABC algorithm, a new method by which to solve the timetabling problem is presented in this article. This improves efficiency and overcomes problems existing in the traditional method.

PROBLEM DESCRIPTION AND MATHEMATICAL MODEL

The timetabling problem involves determining the times and classrooms for all courses, including teacher assignments, so as to avoid conflict among the courses. There are some hard constraint conditions and there are also some soft constraint conditions, which could improve teaching and should be considered as far as possible.

Problem Description

Timetabling involves five elements, viz. teacher, classroom, course, class and time. The goal is the optimal allocation of the above five elements; namely, the reasonable arrangement and use of teaching resources. This is a non-linear multi-objective combinatorial optimisation problem. Generally, there does not exist a unique optimal solution to this problem and so the aim is to find a reasonable approximate optimal solution.

A reasonable and feasible timetable must meet some hard constraint conditions, and also should meet soft constraint conditions, as far as possible. More specifically, hard constraint conditions must be met; if not met, there will be conflict. While soft constraint conditions should be met, as far as possible, to make the timetable more reasonable. Hard constraint conditions include the following:

- A teacher cannot teach more than two courses at the same time.
- A class cannot be on two courses at the same time.
- A course cannot have the same class more than once.
- A classroom cannot be scheduled for two courses at the same time.
- The classroom capacity must be adequate for any course that uses it.

Soft constraint conditions vary according to department, but generally contain the following:

- The total capacity of all the classrooms should be as small as possible.
- The number of students in a class is as close as possible, but not greater than, the classroom capacity, so as to make good use of resources.
- The same course should be distributed evenly within one week.
- Special conditions on teachers, such as working on different campuses, should be minimised.
- Students' courses should be uniformly distributed within one week.
- Classroom adjacent courses should not be nearby.

The Mathematical Model

The five elements in the timetabling problem can be described by the following sets:

The set of teachers $P = \{p_i\}$, $i = 1, 2, \ldots, n_p$;

The set of courses $L = \{l_i\}$, $i = 1, 2, \ldots, n_l$;

The set of classes $C = \{c_i\}$, $i = 1, 2, \ldots, n_c$;

The set of classrooms $R = \{r_i\}$, $i = 1, 2, \ldots, n_r$;

The set of times $T = \{t_i\}$, $i = 1, 2, \cdots, n_t$;

Where n_p, n_l, n_c, n_r, n_t respectively, are the total number of teachers, courses, classes, classrooms and times.

The sets are all ordered, with elements sorted according to an appropriate code. The classrooms include those that are ordinary, multimedia, computer rooms and laboratories; and are sorted according to the different categories, so as to form a one-dimensional linear structure.

The teaching time is from Monday to Friday each week; each lesson is in a basic teaching period, with two classes. The lessons every day cover no more than five basic teaching periods, for a total of 10 classes. The set of times contains five lessons (10 classes), from Monday to Friday. Each lesson can be seen as an element, so there are 25 elements, that is, $n_t = 25$.

The input of the timetabling problem can be described by the set $I = \{i_k\}$, $k = 1, 2\ldots, n$. Each element $i_k = (p_k, l_k, c_k, w_k, h_k, r_k, o_k)$, $p_k \in P$; $l_k \in L$; $c_k \in C$; w_k is the week's classes for c_k; h_k are the whole classes of c_k; r_k is the requirement for the classroom of c_k; o_k are other requirements, including the teacher's request for teaching time, the beginning and end week of a course, and so on. The different courses and classes that each teacher teaches must occur as a single entry in the set I. That is, for each element in the set I, teacher, course and class cannot be the same.

ALGORITHM DESIGN

The Solution

The description of the solution of the artificial bee colony algorithm is most important. Binary code is used in the algorithm described in this article. Considering the characteristics of the timetabling problem and the mathematical model, two-dimensional co-ordinates are used to code. A cross value represents whether some course of a class is arranged at this time.

Group Initialisation

In initial population generation algorithms, the most commonly used method is to generate randomly the initial population, which can avoid a premature solution. However, this will generate too many conflicts and increase the extent of the evolution. Therefore, in this article, a compromise approach is used to generate a group of individuals, with no classroom conflicts as the initial group. The set of the number and capacity of the classrooms also has a great impact on the operation of the algorithm. The classroom is an important resource in the timetabling system. The classroom utilisation rate is defined as:

Classroom utilisation rate = the number of used classrooms/the whole number of classrooms + the total number of students attending class/the whole capacity of classrooms.

This is seen as an important indicator by which to test the effectiveness of the timetabling system. When multiplied by 10 it serves as a fitness value for the computation.

Food Source Evaluation

There is more than one optimisation objective in the timetabling problem, so a combination of multi-object and fitness function is adopted.

- Important courses are arranged at the best time for teaching, as far as possible. For α_i(i = 1,2,3,4,5), there are five teaching units each day. The 1st, 3rd, 5th teaching units are better for teaching; so α_i = 1(i = 1,3,5). The 2nd, 4th teaching units are worse for teaching; so α_i = 0 (I = 2,4). β_j(j = 1,2,3,4) represents the importance of the course as a weighting. The courses contain those that are elective, basic, professional and degree, with weights respectively of 1, 2, 3, 4. Therefore, the optimal objective is $\max(f_1) = \sum(\alpha_i \times \beta_j)$.

- Aim to meet the requirement of teachers' teaching classes at times according to their preference, as far as possible. Set the coefficients according to the title of the teacher χ_i(i = 1,2,3,4). The teaching assistant, lecturer, associate professor, professor are 1, 2, 3, 4, respectively. The willingness to teach in a given time is given by δ_j, δ_j = 0, 1, 2 meaning not willing, indifferent and willing, respectively. So, the optimal objective is $\max(f_2) = \sum(\chi_i \times \delta_j)$.

- Weekly multi-class courses should be arranged, so that classes are separated by at least one day, as far as possible, which can ensure better teaching. ε_j(j = 1,2,3,4) represents the teaching effect coefficient of a course arranged in an interval of j days. So, the optimal objective is $\max(f_3) = \sum(\beta_i \times \varepsilon_j)$.

- Aim to improve the utilisation ratio of resources, as far as possible. A good timetabling result can save a lot of resources. In a lesson, the greater the ratio of the number of students to the capacity of the classroom, the higher the utilisation of the resource. The maximum value is one, which represents no idle resource. So, the optimal objective is $\max(f_4) = \sum(\frac{k_c}{r_r})$.

The fitness function for the timetabling problem is the weighted sum of each target value, that is:

$$F = \sum_{i=1}^{4} \theta_1 \times f_i \tag{1}$$

The value of the weights, θ_i , can be defined by management. These represent the relative importance of each objective. For the four objectives above, the values chosen were 3, 1, 2, 4, respectively.

Worker Bee Stage

In the basic ABC algorithm, the worker bees produce food sources in the neighborhood of their current position. The specific process may include two types of operation. One is to encode a group, and see if the code belongs to the same *teacher-class-course* as applied to the timetabling problem. Then, back to bees, choose randomly a pair of individuals and a group of genes, and exchange the pair of genes. In terms of the timetabling problem, it is possible to avoid errors such as a change in the number of arranged courses, and it is only necessary to detect whether there is conflict between two individuals, including the conflict that may arise among teachers, classes and classrooms at the same time. If there is a conflict, the selection can be cancelled or the individual exchanged. The other operation is to select a set of genes randomly and exchange randomly a non-null alleles with a non-conflict zero gene in the same group.

Observation Bee Stage

In the basic ABC algorithm, observation bees choose the food source according to the probability value q_i. The method is similar to the roulette selection method in the genetic algorithm. By using this, the nectar source with more nectar will attract more observation bees to do a search of the neighborhood. This is an advantage in getting close to the optimal solution. Thus, the formula for choosing the nectar source is as follows:

$$q_i = \frac{fit(x_i)}{\sum_{m=1}^{SN} fit(x_m)} \tag{2}$$

Investigation Bee Stage

In the basic ABC algorithm, the investigation bees randomly generated a food source in the pre-defined search space. However, in the process of evolution, the optimal food source usually is better than that of other food sources. Thus, the operation of the investigation bees will reduce the efficiency of the algorithm. Here, the operation of the investigation bees was improved. The food source is generated by the insertion, exchange and inversion of the optimal food source. In addition, in order to avoid the algorithm identifying a local optimal solution, multiple operations are performed.

SIMULATION EXPERIMENT

The algorithm above was applied to an actual timetabling problem. Data from a computer college in the first semester of 2014-2015 were used as the sample data to test the algorithm's validity. The relevant data are shown in Table 1. The ABC algorithm parameters are: SN = 10, limit = 20, MEN = 500.

Table 1: The sample data.

Number of teachers	Number of courses	Number of classes	Number of classrooms	Numbers in classes
110	105	50	60	18-24

A manual method of timetabling and the ABC algorithm were used to find the optimal timetabling. The results are shown in Table 2.

Table 2: The timetabling scheme.

Method	Classes per day for a class	Interval of same course	Classroom utilisation (%)	Teachers' satisfaction (%)	Utilisation ratio of teaching time (%)
Manual	4-8	1-3	88	85	83
ABC	4-8	1-2	93	90	90

As can be seen from Table 2, the timetable generated by the ABC algorithm is very close to the results of the manual method. Compared to the manual schedule, the *interval of the same course* is improved to some extent. In addition, the *classroom utilisation*, *teachers' satisfaction* and *utilisation ratio of teaching time* are all improved.

There are many variable factors in an actual timetable, e.g. some classes attend the course together, graduating classes attend the course, a course is taught by different teachers. Thus, the timetable needs to be adjusted subtly.

CONCLUSIONS

The timetabling problem is a non-linear multi-objective optimisation problem. The application of the artificial bee colony algorithm reported in this article makes timetabling simple, and with few parameters to set. It can ensure a global optimal solution using the special *nectar search function* and the adjustment function to improve local search. However, only the basic method of solving the problem is presented in this article. In fact, analysis and summary of the timetabling is needed for better results.

ACKNOWLEDGMENTS

This research is supported by the National Natural Science Fund Project of China (Grant No. 71401156); the Ministry of Education Humanities and Social Science Research Base Project (13JJD630017); Zhejiang Provincial Natural Science Foundation of China (LZ13G010001, Z13G010002); Zhejiang Youth Social Science Scholars Action Plan of Zhejiang Province (G178, 13ZJQN059YB); as well as the Contemporary Business and Trade Research Center and Center for Collaborative Innovation Studies of Modern Business of Zhejiang Gongshang University of China (Grant No. 14SMXY05YB).

REFERENCES

1. Even, S., Itai, A. and Shamir, A., On the complexity of time table and multi-commodity flow problems. *SIAM J. on Computing*, 5, **4**, 691-703 (1976).
2. Tang, H.Y. and Zhou, M., Design and implementation of course arrangement for gym. *Microcomputer Infor.*, 22, **1**, 237-240 (2006).
3. Yuan, H.W., Xue, M.G. and Yao, L., An individualized curriculum schedule producing algorithm based on rules. *Computer Engng.*, 32, **4**, 194-197 (2006).
4. Tao, H.T. and Zhang, T.G., A Study on the utilization of automatic course arrangement model based on graph method. *Microcomputer Infor.*, 21, **9**, 129-132 (2005).
5. Li, Q.F., Application of the maximal independent sets in college time-table system. *J. of Guangxi Academy of Sciences*, 22, **4**, 339-341 (2006).
6. Zhou, M., Yang, P., Zhang, Q. and Zhao, Y., Teaching methods for the course *Artificial Neural Networks and Applications*. *World Trans. on Engng. and Technol. Educ.*, 12, **3**, 507-512 (2014).
7. Karaboga, D., An Idea Based on Honeybee Swarm for Numerical Optimization. Technical Report TR06, Erciyes University, Engineering Faculty Computer Engineering Department (2005).
8. Chen, T. and Ju, C., A novel artificial bee colony algorithm for solving the supply chain network design under disruption scenarios. *Inter. J. of Computer Applications in Technol.*, 47, **2-3**, 289-296 (2013).
9. Chen, T. and Xiao, R., Modeling design iteration in product design and development and its solution by a novel artificial bee colony algorithm. *Computational Intelligence and Neuroscience*, 242481 (2015).
10. Liu, Y., Tian, L. and Hu, K., An improved clustering method to evaluate teaching based on ABC-FCM. *World Trans. on Engng. and Technol. Educ.*, 11, **2**, 64-69 (2013).
11. Pan, Q.K., Tasgetiren, M.F., Suganthan, P.N. and Chua, T.J., A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem. *Infor. Sciences*, 181, **12**, 2455-2468 (2011).