Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 103 (2017) 52 – 58

Procedia Computer Science

XIIth International Symposium «Intelligent Systems», INTELS'16, 5-7 October 2016, Moscow, Russia

Variational genetic algorithm for NP-hard scheduling problem solution

A.I. Diveev[a,b], O.V. Bobr[a,*]

[a]*Peoples' Friendship University of Russia, Miklukho-Maklaya str. 6, Moscow 117198, Russia*
[b]*Dorodnicyn Computing Centre of FRC "Computer Science and Control" of Russian Academy of Sciences, Vavilova str. 40, Moscow 119333, Russia*

Abstract

The article is devoted to the study of metaheuristic method for scheduling problems solution. The article describes genetic algorithm successfully applied to solve the problems. The article considers the model of a problem of an optimal timetable development, describes genetic algorithm applied to the problem. The results of the algorithm are provided.

Peer-review under responsibility of the scientific committee of the XIIth International Symposium "Intelligent Systems"

Keywords: timetable; genetic algorithm; small variations of the basic solutions.

1. Introduction

Scheduling theory problems usually represent a wide range of NP-hard combinatorial optimization problems being rich in applications. Most optimization problems do not require an accurate answer. It is sufficient to find a reasonable solution meeting the conditions within a short space of time. Approximate algorithms deal successfully with this task. As far as finding accurate solutions is concerned almost all the NP-hard problems are equivalent to each other. Approximate algorithm development, or the proof of its development impossibility given that P and NP classes mismatch, allows to compare the NP-hard problems based on their approximation.

* Corresponding author. Tel.: +74959550734
E-mail address: bobr_ov@pfur.ru

Peer-review under responsibility of the scientific committee of the XIIth International Symposium "Intelligent Systems"
doi:10.1016/j.procs.2017.01.010

The NP-hard combinatorial optimization problems can be solved using genetic algorithms when the required amount of input data is large, and the rate of exponential data growth for discrete optimization problems can only grow linearly.

Genetic algorithms do not assure finding global optimum in polynomial time, however, as opposed to well-known heuristic and deterministic algorithms of search optimization, they are able to provide a reasonably good solution within a shorter space of time.

The task of scheduling training sessions belongs to the class of combinatorial type problems, characterized by huge dimensions and huge number of limitations of complex shape. In fact, there are no universal methods of solving such problems at the moment. Direct application of mathematical (classical) theory of the schedule to the task of preparation of training sessions is not possible[1]. However, there is a number of heuristic and exhaustive searching methods that are quite amenable to programming.

The approaches to the formalization and solution to the classes scheduling problem in higher educational institutions have been changing with the development of the General theory of schedules. Methods of linear programming[2,3] (1970-1980), the network model[4] (1980), the logical programming with constraints[3,5] (1990s) were applied to formulate the tasks of scheduling.

In the last decade, the direction had a development of searching efficient heuristic methods solution to the problem of classes scheduling of the University, dynamic programming method, branch and bounds method, metaheuristic algorithms[6]. The given problem is solved basically using metaheuristic algorithms such as genetic algorithm[7,8], annealing method[9], ant colony algorithms[6].

The goal of the research is to prove that using genetic algorithm with basic solution variations can be applied to develop the optimal timetable. Metaheuristic algorithms are the best iteration algorithms in terms of operation speed and finding solutions to complex optimization problems.

Genetic algorithm with small variations in basic solution is a particular case of genetic algorithm, and can be applied to find solutions to complex optimization problems. The goal of solving the similar problems is the search and definition of the most relevant optimization solution of the discrete set of possible solutions. Scheduling problem, temporary tables development problem are the typical examples of similar problem solution. These problems are solved in business, academic activities and other areas.

2. The problem model

The problem considered involves developing the optimal timetable for the higher education institution. There is an arbitrary poor timetable from the point of view of hour intervals. Sets of learning groups, course units and professors provide continuous learning process.

It is necessary to develop the timetable meeting the requirements of group curriculum maintenance, using classrooms and professors to the full in the learning process. The optimal timetable must contain the minimum number of hour intervals for students and professors. The peculiarity and the difficulty of the problem is that the possibility of large number of timetable variants iteration sorting is limited, with the number of timetable variants rising dramatically as the number of course units, learning groups, and professors increases. For s course units, t professors, r classrooms, the number of possible timetable variants in an academic day is $(s!)^{tr}$. Given $s = 5$, $t = 3$, and $r = 2$, the number of possible timetable variants is about 3×10^{12}.

For given limited sets of course units, audiences, professors, and learning groups, the training timetable is to be developed meeting the following criteria:

- group curriculum congruence;
- course units and audiences congruence;
- consideration of professors area of specialization

while minimizing total hour intervals for learning groups and professors. Formally, the schedule table can be represented in the matrix form

$$\mathbf{X} = \left[\mathbf{x}^{i,j}\right], \quad i = \overline{1,N}, \quad j = \overline{1,M} \quad , \tag{1}$$

Where N is the number of lessons per week, M is the number of learning groups, $\mathbf{x}^{i,j}$ is an integer vector, consisting of three elements

$$\mathbf{x}^{i,j} = \left[x_1^{i,j} \quad x_2^{i,j} \quad x_3^{i,j}\right]^T \quad , \tag{2}$$

$x_1^{i,j} \in X_1$, X_1 being the variety of course units, $X_1 = \{0,1,...,n_1\}$, $x_2^{i,j} \in X_2$, X_2 being the variety of audiences, $X_2 = \{0,1,...,n_2\}$, $x_3^{i,j} \in X_3$, X_3 being the variety of professors, $X_3 = \{0,1,...,n_3\}$. x_k, $i = \overline{1,3}$ elements being zero-equal simultaneously means a professor does not give lessons in an audience to j group. The optimal timetable with minimum number of hour intervals is to be developed. Criteria of optimality for the scheduling problem are:

$$\begin{bmatrix} F_1 = \sum_{j=1}^{M} X_1^j \to min, \\ F_2 = \sum_{j=1}^{T} Z^j \to min. \end{bmatrix} \tag{3}$$

where Z^j is the matrix of professors engagement in learning process, $Z = \left[z^{i,j}\right]$, $i = \overline{1,N}$, $j = \overline{1,T}$.

There is a number of rigid restrictions not to be broken:

- impossibility of simultaneous giving different lessons in an audience;
- discipline and audience areas of specialization congruence;
- impossibility of a professor being present in different audiences simultaneously;
- audience and professor areas of specialization congruence;
- impossibility of two different professors giving two different lessons in an audience simultaneously;
- course unit and professor areas of specialization congruence.

Let $Q(i,k)$ be the number of academic hours per week meeting the curriculum for all i lessons of k group. Then the requirement of group curriculum congruence for k learning group can be written in the following form:

$$\sum_{i=1}^{N} \theta(x_1^{i,k}) = Q(i,k), \tag{4}$$

where

$$\theta(x_1^{i,k}) = \begin{cases} 1, & \text{if } x_1^{i,k} > 0 \\ 0, & \text{else} \end{cases}.$$

If $x_1^{i,j} = 0$, then there are no i lesson hours for j group. It is reasonable to check the expression when $Q(i,k) > 0$, and $x_1^{i,j} > 0$. The requirement of impossibility of simultaneous giving different lessons in an audience is written as:

$$\sum_{j=1}^{M-1} \sum_{k=j+1}^{M} \theta((x_1^{i,j} - x_1^{i,k}) + (x_2^{i,j} - x_2^{i,k})) = 0 \quad , \tag{5}$$

where $i = \overline{1,N}$. Set of audiences X_2 is divided into subsets after their areas of specialization:

$$X_2 = \bigcup_{k=0}^{L} \beta(x_2^{i,j}, k) \quad , \tag{6}$$

where $\beta(x_2^{i,j}, k)$ is the subset of $x_2^{i,j}$ audiences with k area of specialization. Let $\alpha(x_1^{i,j}, k)$ be the k area of specialization for $x_1^{i,j}$ subject, then the requirement of discipline and audience areas of specialization congruence can be written in the following form:

$$\sum_{j=1}^{M} (\alpha(x_1^{i,j}, k) - \beta(x_2^{i,j}, k)) = 0, \tag{7}$$

where $i = \overline{1, N}$. The requirement of impossibility of a professor being present in different audiences simultaneously is written as:

$$\sum_{j=1}^{M-1} \sum_{k=j+1}^{M} \theta((x_2^{i,j} - x_2^{i,k}) + (x_3^{i,j} - x_3^{i,k})) = 0, \tag{8}$$

where $i = \overline{1, N}$. Let $\gamma(x_3^{i,j}, x_1^{i,j}, k)$ be k area of specialization of $x_3^{i,j}$ professor giving $x_1^{i,j}$ lesson, then the requirement of audience and professor areas of specialization congruence can be written in the following form:

$$\sum_{j=1}^{M} (\beta(x_2^{i,j}, k) - \gamma(x_3^{i,j}, x_1^{i,j}, k)) = 0, \tag{9}$$

where $i = \overline{1, N}$. The requirement of impossibility of two different professors giving lesson in an audience simultaneously is written as:

$$\sum_{j=1}^{M-1} \sum_{k=j+1}^{M} \theta((x_3^{i,j} - x_3^{i,k}) + (x_2^{i,j} - x_2^{i,k})) = 0, \tag{10}$$

where $i = \overline{1, N}$. The requirement of course unit and professor areas of specialization congruence can be written as:

$$\sum_{j=0}^{M} (\alpha(x_1^{i,j}, k) - \gamma(x_3^{i,j}, x_1^{i,j}, k)) = 0, \tag{11}$$

where $i = \overline{1, N}$. Let D be the number of academic days. Then N/D is the number of lessons per one day. d day of i lesson is determined from the formula $d = \left\lceil \frac{i-1}{N/D} \right\rceil + 1$. The number of r first lesson in d day is determined from the formula $r = (d-1)\frac{N}{D} + 1$. The number of p last lesson in d day is determined from the formula $p = d\frac{N}{D}$. The criteria of optimality are the following: total hour intervals for learning groups and total hour intervals for professors:

$$F_1 = \sum_{j=1}^{M} \sum_{d=1}^{D} \sum_{r=1}^{N/D} (q_{r+1}^{j,d} - q_r^{j,d} - 1) \to min, \tag{12}$$

where $q_r^{j,d}$ being an ordered set of $x_1^{i,j}$ non-empty elements indices

$$F_2 = \sum_{t=1}^{T} \sum_{d=1}^{D} \sum_{r=1}^{N/D} (q_{r+1}^{t,d} - q_r^{t,d} - 1) \to min, \tag{13}$$

where $q_r^{t,d}$ being an ordered set of non-empty elements indexes of matrix of professors' engagement in learning process, T being the total amount of professors. Apart from rigid restrictions, there can be a number of soft constraints that may be broken without loss of timetable correctness. Those may include professors' preferences concerning the day of giving lessons. Therefore, the process of optimal timetable development involves course units sorting in lessons grid with regard to constraints and criteria of optimality.

3. Proposed method

There are no efficient exact methods for solving scheduling theory problems. There is a small number of scheduling theory problems with complexity increasing polynomially as the amount of input data grows[6]. For example, these are product parallel one-stage or multistage machining problems[6]. But the most existing scheduling theory problems do not have polynomial solution methods. Due to large amount of input data it can be believed that in most cases it is impossible to find a method providing accurate solution. To find an approximate solution for the problem given, it was decided to conduct a research based on a metaheuristic method of genetic algorithm[10].

Genetic algorithm simulates evolutionary processes in generations. In the same manner as it is in the nature, these algorithms perform the search of "good" chromosomes in order to develop an optimal timetable using selection and reproduction. A certain evaluation of every chromosome is required in order to reflect its fitness. Selection allows to pick out the chromosome with the most fitness while reproduction allows to produce new chromosomes[11].

Such algorithm operation allows to identify the best solution in a large population size with higher probability.

4. Genetic algorithm

Genetic algorithm based on the principle of basic solution is used to find the solution. This approach allows to apply genetic algorithm operations to basic solution small variations sets, and not to problem encoded solutions, chromosomes.

Let us define the basic solution small variations vector as:

$$\mathrm{w} = [w_1, w_2, w_3, w_4]^T, \tag{14}$$

w_1 being variation code. Let us introduce the following variations:

- $w_1 = 0$: any two rows exchange at w_2 and w_3 lessons in X_1, X_2, X_3 tables simultaneously;
- $w_1 = 1$: audiences exchange at w_2 lesson for w_3 and w_4 groups;
- $w_1 = 2$: course units (audiences, professors) exchange for w_2 group at w_3 and w_4 lesson simultaneously;
- $w_1 = 3$: audiences exchange for w_2 group at w_3 and w_4 lesson.

Therefore, genetic algorithm based on basic solution variations can be applied to the optimal timetable development problem.

Chromosome Pareto front rank is determined using (12), (13). Fitness function for the genetic algorithm is determined using Pareto front rank:

$$F(\mathrm{W}_i) = \frac{1}{r} \tag{15}$$

5. Experimental setup

The basic timetable with $M = 4$ learning groups, $t = 24$ professors, and $r = 9$ audiences was selected to conduct the experiment.

Let us limit the number of lessons per week to be $N = 30$, workweek to be $D = 6$ academic days, then the number of lessons per day is 5.

For processing convenience, let us assign numbers to the input data. The schedule table will be represented in the form of three tables: course units distribution table, professors engagement distribution table, audiences distribution table.

Genetic algorithm based on basic solution variations has been applied to develop the optimal timetable. Table 1 sets the timetable variations.

Table 1. Timetable variations.

Variation number	Variation description	w_1	w_2	w_3
0	Rows exchange at random spaces of time	Row number	Row number	-
1	Audiences exchange at the same space of time	Row number	Column number	Column number
2	Audiences exchange for groups	Column number	Row number	Row number
3	Audiences exchange without subject and professor for a group	Column number	Row number	Row number

Checking for collisions covers the three distribution tables. The experiment had been executed with the following parameters: population size is 1024, number of generations is 512, number of crossovers is 256, chromosome length is 12.

Functional values have been improved in the course of algorithm operation: initial timetable had $F_1 = 16, F_2 = 12$. After one iteration of optimization we achieved functional values $F_1 = 8, F_2 = 13$.

6. Conclusion

Genetic algorithm has been successfully applied to develop the optimal timetable. The basic idea of the algorithm is to use basic solution small variations that aid to find the best solution for hard optimization problems. The approach appears advanced due to its efficiency as far as complicated problems solution is concerned.

Altering algorithm parameters may pick out the input parameters values for the optimal algorithm operation. These recommendations are appropriate for the whole class of problems.

Singularity-free problem has been considered in the experiment presented. The further research are to include analysis and undefined requirements. Adding various local search strategies to the given program in order to obtain more precise results is another important line of research.

References

1. Finkelstein Yu Yu. *Applied methods and applied problems of discrete programming*. Moscow: Science; 1976.
2. de Werra D. An introduction to timetabling. *Eur J Oper Res* 1985;**19**:151-62.
3. Lach G, Lübbecke ME. Curriculum based course timetabling: new solutions to Udine benchmark instances. *Ann Oper Res* 2012;**194(1)**:255-72.
4. Wren A. Scheduling, Timetabling and rostering – a special relationship? In: Burke E, Ross P, editors. *Practice and Theory of Automated Timetabling*. Berlin: Springer; 1995. p. 46-75.
5. Rudova H, Vlk M. Multi-criteria soft constraints in timetabling. In: Kendall GLL, Pinedo M, editors. *Proc. of the 2nd Multidisciplinary International Conference on Scheduling*. New York: Stern School of Business; 2005. p 11-5
6. Lazarev AA, Gafarov ER. *Scheduling Theory. Problems and algorithms*. Moscow: MSU; 2011.
7. Yandibaeva NV. Geneticheskiy algoritm v zadache optimizatsii uchebnogo raspisaniya (Genetic algorithm in the university timetable optimization problem). *Sovremennye naukoemkie tekhnologii (Modern science-intensive technologies)* 2009;**11**:97-8.

8. Astakhova IF, Firas AM. Sostavlenie raspisaniya uchebnykh zanyatii na osnove geneticheskogo algoritma (Genetic algorithm based preparation of the academic timetable). *Vestnik VGU: Sys An and Inf Tech* 2013;**2**:93-9.
9. Klimenko AB, Klimenko VV, Taranov A Yu. Methods of parallelization of simulated annealing at a decentralized solution to the problem of scheduling with minimum number of resources. *Estestvennye i matematicheskie nauki v sovremennom mire (Modern Nat Sci and Math)* 2014;**22**.
10. Gladkov LA, Kureichik VV, Kureichik VM. *Genetic algorithms*. Moscow: Fizmatlit; 2010.
11. Cormen T. *Introduction to Algorithms*. MIT Press; 2009.