THE USV ANNALS OF ECONOMICS AND PUBLIC ADMINISTRATION | VOLUME 22, ISSUE 1(35), 2022

USING GENETIC ALGORITHMS TO UNIVERSITY TIMETABLING PROBLEM

Ionuţ BĂLAN
"Ştefan cel Mare" University of Suceava, Romania
ionut.balan@usm.ro

Received 23 March 2022; Accepted 13 June 2022

Abstract:

The problem of timetabling in universities is an NP-hard problem, which involves a task scheduling activity for certain people based on multiple constraints and a limited number of resources. This paper addresses a complicated issue, that of scheduling academic classes (timetabling). The aim is to find a feasible schedule for the Faculty of Economics, Administration and Business at the University "Stefan cel Mare" in Suceava. The approach focuses on the use of genetic algorithms and has, as its main objective, the minimization of idle time for students who participate in teaching activities.

Key words: genetic algorithm, chromosomes, mutations, cross-overs, optimization, timetabling

JEL classification: I14, I15, R58, L86

1. INTRODUCTION

UCTTP (University Course TimeTabling Problem) is an optimization problem in the NP-hard problem class. This activity appears at the beginning of each semester in universities and includes the allocation of events (courses, professors and students) to a number of time intervals and fixed rooms. This problem must satisfy both hard and soft constraints when allocating events to resources. As for the hardware constraints, they must be fully met, and in order to increase the quality of the schedules, it is desired to fulfill as many software constraints listed in the problem (Asmuni, 2008, Obit, 2010).

The hard constraints that may arise in an optimization problem, from the category covered in this paper, could be the following (Balan, 2021):

- The time interval 8-20 cannot be exceeded;
- It is not possible to set classes on days other than the 5 intended for teaching activities (from Monday to Friday);
- A teacher cannot take two different courses in the same time frame;
- A student cannot participate in two different activities in the same time frame;
- The number of students present at an activity must not exceed the number of seats in the hall;
- Activities can be placed only in rooms intended for those categories of activities for which they were intended;
- The maximum number of course hours for one student per day will be 6, while the maximum total number of teaching activities will be 8.

The soft constraints that could be encountered in such a problem are: the teacher could opt for some intervals in which to carry out his activity, the teacher could specify the room in which to carry out his activity, also it can be given a break (lunch) for people who start activities at 8 am, etc.

Among the approaches used so far to solve the problem of UCTTP we list the following: techniques from operational research like graph coloring (Budiono et al.,2012) and liniar programming (Prabodanie, 2017), genetic algorithms (Alsmadi et al., 2011, Balan, 2021), ant colony optimization (Mazlan et al., 2019), memetic algorithm (Jat et al., 2008), harmony search algorithm (Al-Betar, 2012), particle swarm optimization (Fen et al., 2009), tabu search (Islam et al., 2016), but also multicriteria optimization variants (Darmawan et al., 2016, Mahbub et al., 2020).

In this paper we used the description of the results obtained in UCTTP, a problem solved using genetic algorithms. The described results were obtained by running a program written in Java, using object-oriented technology, with data from the Faculty of Economics, Administration and Business at the University "Stefan cel Mare" Suceava.

2. PROBLEM DESCRIPTION

The problems of educational timetabling can be of several types: scheduling university exams, scheduling university courses, scheduling in various projects and others. The difficulties that arise in the development of such programming are increasing. Higher education institutions are introducing more and more new specializations, creating new faculties, which leads to an increase in the number of students. In most cases, the growth rate of these elements is clearly higher than the growth of material resources (mainly educational spaces). This leads to an increase in the number of restrictions that must be observed when trying to create a university schedule. For example, within the “Ştefan cel Mare” University of Suceava, there are 11 faculties, with different numbers of specializations in the undergraduate programs. Thus, from faculties that have two specializations, with a total of 12 groups, for 4 years of study, we reach the Faculty of Economics, Administration and Business to a total of 53 groups, for 3 years of study. This increase in the number of groups leads to a considerable effort, when it is desired to create a timetable for students. Being a difficult combinatorial optimization problem, solving it using manual processing is quite difficult, the time resource that is needed being wasted, regardless of the size of the human resource involved in this programming activity. This consideration obliges us to resort to finding IT solutions in solving this problem.

Among the methods used to solve the optimization problem described, we resorted to population-based techniques, namely genetic algorithms. Genetic algorithms are used in various fields to solve optimization problems. They are inspired by the process of natural evolution, which is reinforced by their name. Algorithms work with a population of possible solutions and include three key steps: selection, regeneration, and replacement. In the selection phase, individuals with high fitness are selected to be parents for the next generation. In the regeneration phase, two operators are used: crossing and mutation on the parents who were selected in the first phase and in the replacement phase: the individuals from the initial population are replaced by the newly created individuals.

Genetic algorithms, which are a subset of EA (Evolutionary Algorithms), are based on the following steps: (1) initial population generation, (2) evaluation of the population generated using the evaluation function, (3) selection of parents to cross based on the information obtained from the evaluation functions, (4) the cross-breeding operator applies to produce children, (5) the mutation operator applies to produce children, (6) parents and children are selected to form the new population for the next generation, and (7) if the completion condition is met, the algorithm stops, otherwise moves to step two (next generation) and continues (Obit, 2010).

The population used in the tests whose results are presented in this article consists of 100 chromosomes, which will evolve over 200 generations. In each generation we will apply 2 different mutation operators (the first 20 times and the second 10 times) and one crossover (20 times).

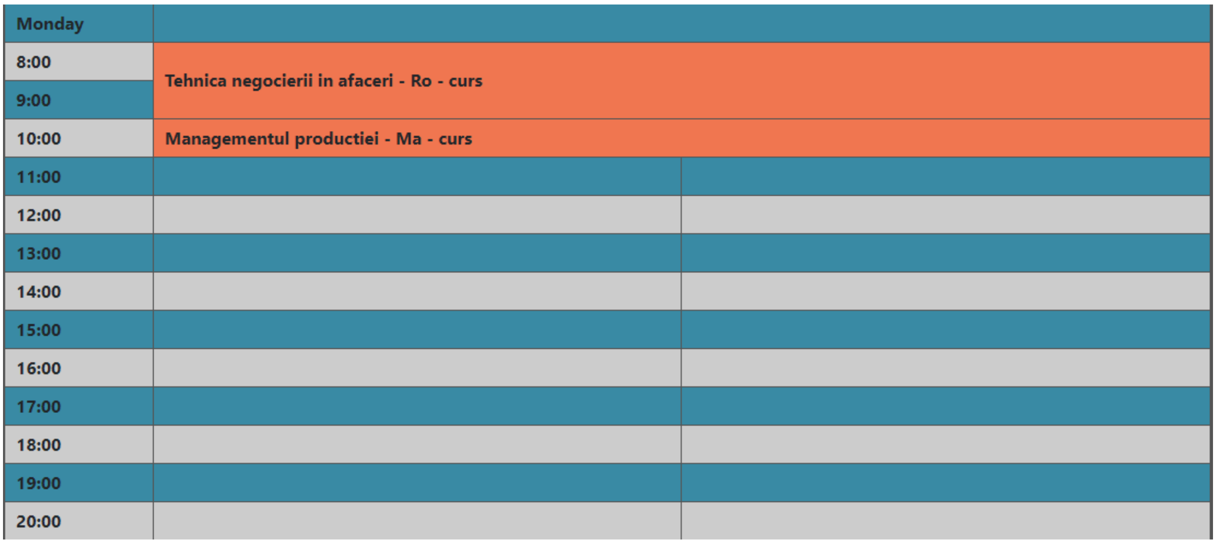
A chromosome has the size equal to the number of activities that will be entered in the timetable (we will have 523 didactic activities that will have to be entered in the timetable), to which is added an element that will memorize the value of the objective function. A gene will provide us with a series of information, such as: the name of the specialization, the name of the discipline, the person who teaches that discipline, the year of study, the group, the number of students, whether it is a course or not, number of weeks in which the activity is held , number of hours per week, number of groups and semi-groups of that specialization.

3. COMPUTATIONAL RESULTS

The problem that wants to be optimized has as input data the courses and practical activities (seminars and laboratories) within the Faculty of Economics, Administration and Business within the “Ştefan cel Mare” University of Suceava. The data taken into account are related to the second semester of the academic year 2019-2020. The number of specializations for which the optimization will be performed is 9, and for each specialization we have 3 years of study. The institution employs 63 teachers, who can have from one activity per week, to over 25 activities per week. The activities related to the studied problem take place from Monday to Friday, from 8:00 to 20:00. The course activities are held for one year related to a certain specialization, the seminars are held with a group belonging to a specialization, and the laboratories are held at the level of a semi-group. From the previous statement it results that at the moment when in a certain specialization course activities are carried out, the activity of seminars or laboratories cannot be carried out.

In our case, I want to optimize the idle time for the "student" resource. This human resource is of great importance for the institution for which the schedule is generated, so in this approach we have focused on this category. The objective function depends on the number of free hours that students have between two activities that take place on the same day. Students can be divided on activities into 3 main categories of groups: the whole year for courses, the group for seminars and the semi-group for laboratories. The calculation of the idle time value for each specialization was done by summing the free hours, between teaching activities, at the semigroup level.

Figure 1 shows the schedule for a specialization (marked AF3, which consists of 2 groups and 2 semi-groups). It is noticed that the idle time is small, being given by the 3 hours, from Tuesday, in the first group, between 12:00 and 15:00. Of course, the specialization chosen to be represented is not so complex. In other specializations that consist of several groups (and semi-groups) the number of break hours will increase. In the paper "A new genetic approach for Course Timetabling Problem", we presented both the composition of each specialization and the evolution of the function to be optimized (in the 200 generations), but also the schedule of a specialization consisting of 2 groups and 3 semigroups (where several hours of pause are observed). Although, in the more numerous study formations, several unwanted breaks appear, with the help of the developed algorithm we reached the performance, to obtain after the 200 generations, a total number of hours representing idle time, values below 200.



a.Moday

Tuesday		
8:00		
9:00		
10:00		
11:00	Fiscalitate - Lu - sem	
12:00		
13:00		
14:00		
15:00	Tehnica negocierii in afaceri - Ro - sem	
16:00	Contabilitatea grupurilor de societati - Ma - sem	
17:00		Baze de date - Io - lab
18:00	Contabilitate manageriala - Lu - sem	
19:00		
20:00		

b.Tuesday

Wednesday		
8:00		
9:00		
10:00		
11:00		
12:00		
13:00	Managementul productiei - Ma - sem	
14:00		
15:00		
16:00		
17:00		
18:00		
19:00		
20:00		

c.Wednesday

Thursday		
8:00		
9:00		
10:00		
11:00	Baze de date - Io - lab	
12:00		Fiscalitate - Lu - sem
13:00		Contabilitate manageriala - Lu - sem
14:00		
15:00		Tehnica negocierii in afaceri - Ro - sem
16:00		Managementul productiei - Ma - sem
17:00		
18:00		Contabilitatea grupurilor de societati - Ma - sem
19:00		
20:00		

d.**Thursday**

Friday		
8:00		
9:00		
10:00		
11:00		
12:00		
13:00		
14:00	Contabilitate manageriala - Ca - curs	
15:00		
16:00	Fiscalitate - Mi - curs	
17:00		
18:00	Baze de date - Io - curs	
19:00	Contabilitatea grupurilor de societati - Ma - curs	
20:00		

e.Friday

Figure 1. Weekly schedule for a specialization

Through the results we have achieved the main objective proposed, that of minimizing idle time for students. However, in addition to this human resource, within the institution, we also have teacher resources, for which no attempt has been made to optimize this idle time. The distribution of classes to full-time teachers is not uniform. Thus, some teachers have several activities to support, while others support two, or perhaps only one, activity per week. Given that in this paper the focus was on students, the allocation of teachers in the schedule may create dissatisfaction for the latter category. In figure 2 we presented the connection between the number of hours that each teacher has and the number of days in a week, in which they support their activities.

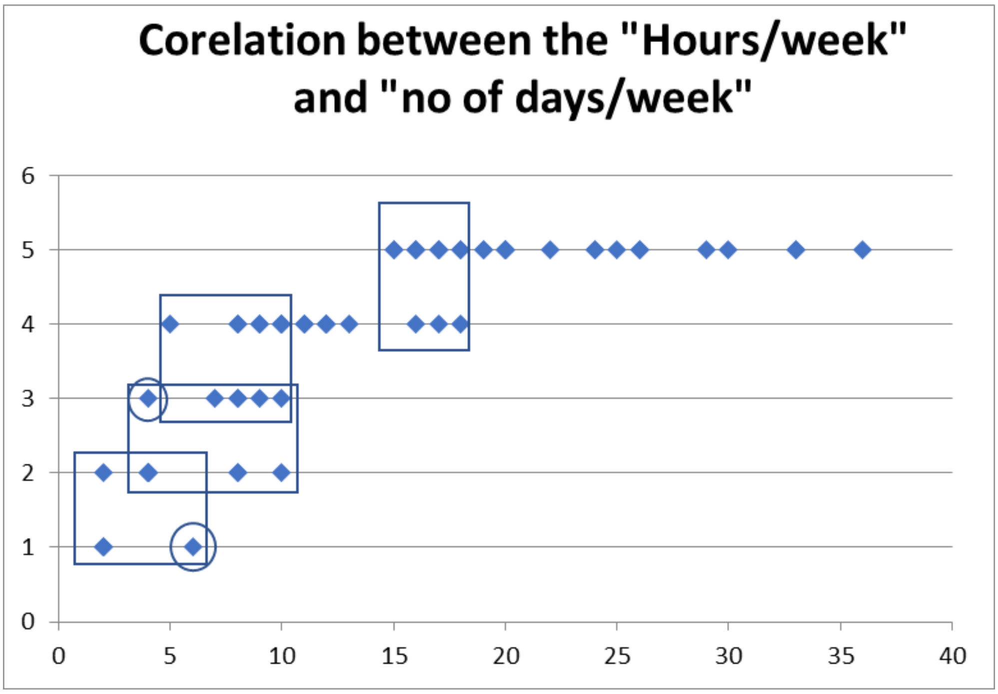


Figure 2. Correlation between the number of hours per week and the number of days per week required to support the activity

From this figure we can see that in the vast majority of cases the distribution of hours for each teacher is fair. Small exceptions are noticed in some cases (exceptions highlighted in rectangles). Thus, a teacher who has to work 6 hours a week will need only one day a week to complete his / her tasks, while another person who has to do 4 hours of teaching activities will be forced to comes to work in 3 days (examples highlighted in the figure by a circle).

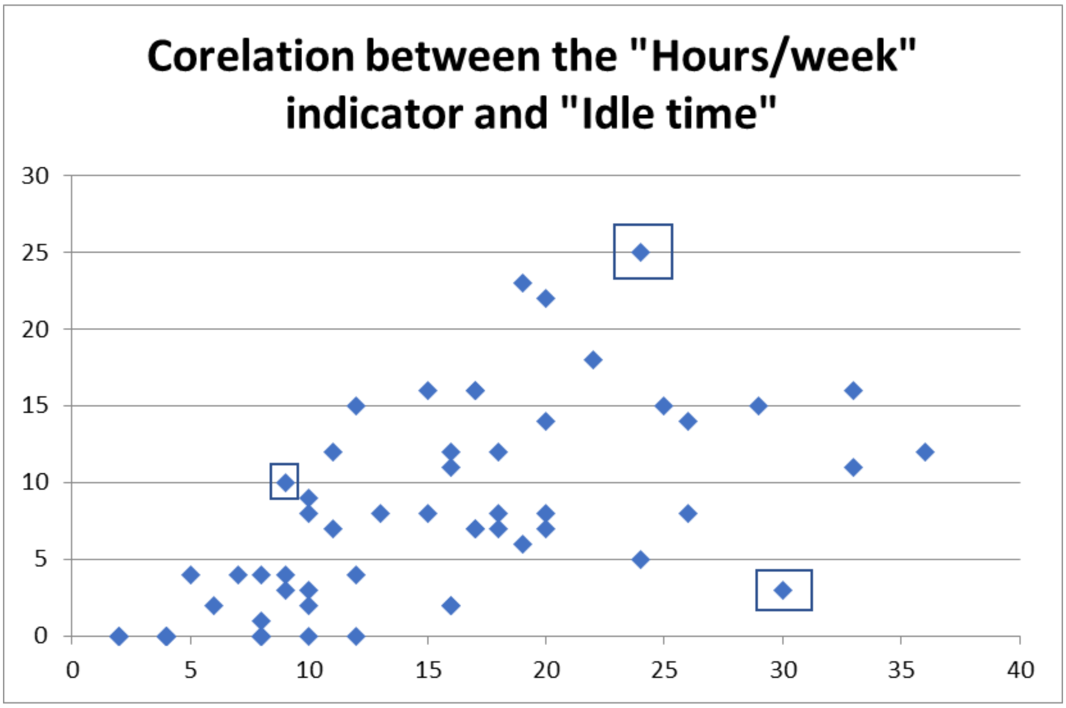


Figure 3. The correlation between the number of hours per week and the idle time related to each teacher

In figure 3 we have represented the connection between the number of weekly hours related to each teacher and idle time for each one. There is also a small problem in this case. Thus, it is reached that a teacher who has to take 30 hours of teaching activities to have only 3 hours of break, another who has 24 hours of activity to have 25 hours of break, and the third with 9 hours of activity to have a 10-hour break. All three cases are highlighted in the figure in rectangles.

In order to improve these last results, without affecting the results obtained for the student, a multi-criteria approach is recommended for the studied problem. Genetic algorithms can also be used in this approach. The optimization of idle time for teachers can be done at the same time as that for students, a new thing that could appear, can be considered as a soft constraint, would be taking into account the schedule options of teachers.

CONCLUSIONS

In this paper we present the results obtained using genetic algorithms, for a university timetabling problem. The case study was a faculty from the "Ştefan cel Mare" University of Suceava. The focus was on student human resources. Given the complexity of such a problem, the results obtained were quite good, the running times were satisfactory and the computing resources used were not with a high degree of performance (Intel core i7 processor, 12 GB RAM). This computational approach brings a real benefit to the organization, compared to the classic, manual approach, which involved allocating a certain number of people for this activity, which should, depending on the complexity, allocate many hours to perform tasks. The results obtained using the described algorithms are clearly better to the classical variant. However, the student is not the only human resource involved in teaching activities. Even if a student-centered education is desired, we must also consider the didactic human resource (teachers). One proposal, to satisfy both categories of people involved in the educational process, would be to use multi-objective optimization, an approach that will lead to a balancing of idle time for both students and teachers. This new approach will be debated taking into account the schedule preferences of teachers.

REFERENCES

1. Al-Betar, M.A., Khader, A.T., (2012), *A harmony search algorithm for university course timetabling*, Annals of Operations Research 194(1):3-31
2. Alsmadi, P.-O., Abo-Hammour, Z., Abu-Al-Nadi, D.I., Algsoon, A., (2011), *A novel genetic algorithm technique for solving university course timetabling problems*, 7th International Workshop on Systems, Signal Processing and their Applications (WOSSPA)
3. Asmuni, H., (2008) *Fuzzy methodologies for automated university timetabling solution construction and evaluation*, Ph.D. Thesis, School of Computer Science University of Nottingham
4. Balan, I., (2021) *A new genetic approach for Course Timetabling Problem*, Journal of Applied Computer Science &Mathematics, Vol.15, No.1
5. Budiono, T., Wong, K.W., (2012), *A pure graph coloring constructive heuristic in timetabling*, Conference: International Conference on Computer & Information Science (ICCIS)
6. Darmawan, V.E.B, Chen, Y.-W., Larasati, A., Dwiastuti, A., (2019), *Multi-objective Modeling for a Course Timetabling Problem*, International Conference on Creative Economics, Tourism & Information Management
7. Fen, H.S, Deris, S., Hashim, S.Z.M., (2009), *University course timetable planning using hybrid particle swarm optimization*, GEC '09: Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation
8. Islam, T., Shahriar, Z., Perves, M.A., Hasan, M., (2016), *University Timetable Generator Using Tabu Search*, Journal of Computer and Communications > Vol.4 No.16
9. Jat, S.N., Yang, S., (2008), *A Memetic Algorithm for the University Course Timetabling Problem*, 20th IEEE International Conference on Tools with Artificial Intelligence
10. Mahbub, M.S., Ahmed, S.S., Ali, K.I., Imam, M.T., (2020), *A Multi-Objective Optimization Approach for Solving AUST Classtimetable Problem Considering Hard and Soft Constraints*, I. J. Mathematical Sciences and Computing, 5, 1-14
11. Mazlan, M., Makhtar, M., Khairi, A.F., Mohamed, M.A, (2019(), *University course timetabling model using ant colony optimization algorithm approach*, Indonesian Journal of Electrical Engineering and Computer Science 13(1):72-76
12. Obit, J. H., (2010), *Developing novel meta-heuristic, hyper-heuristic and cooperative search for course timetabling problems*. Ph.D. Thesis, School of Computer Science University of Nottingham
13. Prabodanie, R., (2017), *An Integer Programming Model for a Complex University Timetabling Problem: A Case Study*, Industrial Engineering & Management Systems 16(1):141-153