



A Course Timetabling Problem for Classroom Usage Minimization

Selin Yavuz, Zülal Demirci, Begüm Güven, Zeynep Kuleli, Buğra Yılmaz, and Deniz Türsel Eliyi^(✉)

Department of Industrial Engineering, İzmir Bakırçay University, İzmir, Türkiye
deniz.tursel@bakircay.edu.tr

Abstract. University course timetabling problem is a problem that tries to optimally assign days, hours, and classrooms for courses in the curriculum, taking into account constraints such as availability of the lecturers, capacities of the lecture halls/classrooms and possible other special rules. In this study, the course timetabling problem of İzmir Bakırçay University Industrial Engineering Department is considered. Although it is a relatively new university in Turkey, it has become increasingly difficult to prepare course timetables due to the increasing number of departments, courses, and students, as well as the very limited current number of classrooms, lecture halls and laboratories within the campus. Currently, the course schedules are prepared with heuristic methods manually. This process is extremely challenging due to the mentioned scarce resources, the most critical being the number of amphitheaters dedicated to the department, as well as the limited number of computer labs. Even coming up with a feasible timetable for a semester can take weeks and several hundred valuable man-hours in the current situation. In order to solve the real-life problem on hand, an integer programming model was developed with an objective of minimizing the used number of classrooms. The model takes into account the unavailable hours of the lecturers. It also considers many constraints such as not overlapping the classes of consecutive years' lectures, as well as other traditional timetabling constraints. The model was solved using IBM ILOG CPLEX Optimization Studio, an optimization software package. In addition, an interactive and simple user interface was developed in MS Excel VBA programming language, which communicates with the optimization software automatically, thus allowing the decision makers to use the interface without the need for any technical knowledge. Optimal course timetables meeting the university rules and lecturer preferences can be created automatically and efficiently as a result.

Keywords: Operations research applications · Course timetabling · Integer programming · Optimization · User interface

1 Introduction

Scheduling is the decision-making process for allocating limited resources to jobs for specific purposes, within a specific time frame and under some constraints. It serves

purposes such as no delay in delivery dates, reduction of overtime, quick response to requests and the most effective use of opportunities. Time scheduling is a decision-making tool that plays a very important role in the production and service industry, and it is used in many areas. The time scheduling problem aims to use the time efficiently and ensures that the necessary assignments are made to the limited resources within a certain time interval while satisfying certain rules and constraints. It is quite common in all educational institutions including universities. It comes in various forms such as course scheduling and exam scheduling.

The university course scheduling problem is the problem of assigning days and hours to the courses, taking into account the constraints such as the availability of the lecturers, the capacities of the lecture venues and the rules of the educational institutions. There is no standard scheduling model, as each educational institution has different physical and administrative conditions. Constraints in this problem are generally divided into soft and hard. Hard constraints are the constraints that must be met, while soft constraints are the constraints that do not have to be met, however they adversely affect the solution if they are not satisfied. For this reason, course scheduling problems are usually modeled with the aim of minimizing soft constraint violations.

There are many methods used to solve university course scheduling problems in literature, such as mathematical programming, heuristics, metaheuristics, and constraint programming. This study involves the determination of the course schedules of Izmir Bakırçay University, Department of Industrial Engineering. The objective function of the course timetabling problem on hand is developed in line with the determined constraints and the optimal solution of the problem is obtained with real data. A user-interactive interface is also developed in order to increase the usability of the model and its the optimal solution.

A literature review of similar studies was conducted in order to determine the appropriate methods and techniques for the use of minimum number of physical rooms and to meet the predetermined constraints. As there is a vast amount of literature on course scheduling, only similar studies and especially those within similar institutions within the country are included in literature review.

While solving the course timetabling problem, strict and flexible constraints were used in [1], as in most studies in the literature. The preferences of the course instructors were included in their flexible constraints. Courses were classified as single session and double session courses. The optimal solution was obtained via developing a mathematical model and solving it on a commercial solver. Binary integer models were used in [1] and [2].

In another study, which aimed to create an optimal course schedule, a mixed integer mathematical model was used for the solution [3]. The importance level of each objective in the model was determined by using the Fuzzy Analytical Hierarchy Process method. On the other hand, in [4] and [5], the optimal result was achieved via 0–1 integer programming models. Reference [5] used only one decision variable in the model, which made the model leaner. A binary integer programming model was used together with goal programming in [6], as the problem had multiple objectives. A fuzzy logic-based method was utilized for the solution due to the uncertainty of the parameters in the model.

Reference [7] used a genetic algorithm to obtain a near-optimal timetable. A software with a user-interactive interface was developed using the C++ programming language in their study. In another study, the authors used C# programming language and artificial bee colony metaheuristic to solve the course scheduling problem [8]. Using this algorithm, a software with a interactive interface was developed. Reference [9], which employed a Mixed Integer Programming (MIP) model instead of heuristic methods, stated that optimal solutions were obtained quickly in their decision support system. Other authors also preferred to use a linear integer programming model for solving their problem, as in [10]. The model was solved using a small example; however, the optimal solution could not be reached. For this reason, it was suggested to use heuristic methods for the solution of the problem. Reference [11] also used a linear integer programming model, in which the courses were categorized as compulsory and elective. The number of restrictions and decision factors were kept to a minimal to ensure that the solution did not take too long. Heuristic approaches were adopted because the model was insufficient for solving the problem. For the first stage, a greedy algorithm was used, and an iterative forward search approach was used as an alternative. Tabu search and simulation annealing techniques were utilized in the second stage to find a workable solution.

Reference [12] included the development of a software program that eliminated the necessity for manual techniques. The content of their software included a user interface, a database and a heuristic analyzer. For obtaining the solution, the authors used a hybrid genetic algorithm, which is an artificial intelligence method. While creating solutions, many approaches were tried, and the most suitable approach was determined. In another study, a database for recording and querying the data was included [13]. The teaching plan was modeled on the ant colony algorithm by grading the requests of the faculty members in terms of time, and a weighted graph model was created. While scheduling the curriculum, pedagogical principles were taken into account. The application was developed with C programming language and then rewritten with the Python programming language due to the need for an interface for data entry. Reference [14] also includes the preparation of a curriculum based on pedagogical principles. According to the included pedagogical principles, it was deemed appropriate to come to school three days in the middle of the week. Because it was thought that the perception of students can be more productive these days. A genetic algorithm was used to solve the problem. In addition, different from standard genetic algorithms, problem-specific genetic algorithms were developed.

In [15], the course timetabling application was developed in Matlab and the solutions were obtained via a genetic algorithm. In the constraints determined for the problem, the satisfaction of the students, lecturers and faculty personnel were prioritized. Reference [16] also took into account the satisfaction of students, lecturers and faculty staff while solving their problem using the Hungarian Method. Reference [17], on the other hand, developed a new heuristic approach and achieved efficient solutions for their problem. Reference [18] designed a structure graph and pheromone model in accordance with the course schedules. Their ant colony optimization algorithm was able to solve problems with multiple heterogeneous constraints of ant systems. Finally, the study in [19] utilized artificial immunity algorithms for optimal solutions of the problem.

In our study, unlike the existing studies in the literature, we aim classroom usage minimization. An integer programming model was created for this purpose, and it was solved using IBM ILOG CPLEX Optimization Studio. A user interactive interface integrated with CPLEX was also developed using MS Excel VBA. This interface automatically presents the course schedules to the user. The following sections include our problem definition, mathematical model, our interface design and conclusions.

2 Problem Definition

The course scheduling or course timetabling problem is one of the most common scheduling problems encountered in all educational institutions, especially universities. In this problem, the goal is to determine the courses to be opened for each class and class times by meeting the desired constraints. The aim for course timetabling is to create an optimal course schedule that does not require long time and effort, in which both students and lecturers are productive.

İzmir Bakırçay University is a newly established and growing university in İzmir, Türkiye. The institution has to consider the increasing number of departments and students every year while solving the course scheduling problem, which became an increasingly difficult problem over the years. The course schedules that have been prepared within the university until now are created with traditional manual methods. These methods require a lot of effort and time, and are insufficient to meet the demands of the university in today's conditions. Considering all these factors, it is aimed to prepare an efficient course schedule in this study by using optimization, and to develop a user-interactive interface design for solving this important problem.

2.1 Assumptions

Currently, a hybrid education system including 40% online and 60% face-to-face classes has been adopted at İzmir Bakırçay University. For this reason, 2 separate course timetables are to be created for online and face-to-face classes. While creating the course schedules, the availability of the lecture halls, classrooms and laboratories are taken into account as well as the schedules of the lecturers and the department. The lectures in the relevant department, which is the Department of Industrial Engineering, are held on weekdays. The classes start at 9 am and end at 5 pm.

There are 3 physical spaces in total for planning the lectures: 1 computer laboratory, 1 classroom and 1 lecture hall, which are open to the use of the department. If the number of students enrolled in the course is more than 55, the lectures for this course should be held in the lecture hall. The courses in the curriculum of the department are divided into two categories as compulsory and elective.

2.2 Integer Programming Model

Below are the indices, sets, parameters and decision variables used in the mathematical model.

Indices

i, i' : Lecture indices,
 d, d' : Day indices,
 t, t' : Time range indices,
 r : Class (1 to 4) indices.

Sets

D : Set of days for face-to-face lectures,
 T : Set of time slots,
 I : Set of lectures,
 I_{D1} : Set of lectures taking place in the classroom,
 I_{D2} : Set of lectures taking place in the lecture hall,
 I_{D3} : Set of lectures that take place in the laboratory,
 I_T : Theoretical lecture set of courses taught in the form of theory and practice,
 I_U : The practice lecture set of the courses taught in the form of theory and practice,
 I_{S2} : Set of lectures lasting two hours,
 I_{S3} : Set of lectures lasting three hours,
 I_r : Set of courses that belong to the r class,
 I_h : Set of lectures given by instructor h ,
 I_{rz} : set of compulsory lectures for class r ,
 α_0 : Undesired time slots for instructor-time matches.

Parameters

p : Number of classrooms to be used,
 j : Number of lecture halls to be used,
 k : Number of laboratories to be used,
 w_i : duration of lecture i ,
 $c_{dt1}, c_{dt2}, c_{dt3}, c_{dt4}, c_{dt5}$: Penalty costs of the deviation variables.

Decision Variables

X_{idt} : 1 if lecture i is assigned to day d and time slot t , 0 otherwise,
 Z_{dtr}, A_{dt}, B_{dt} : Deviation variables.

Based on the above definitions, below is the objective function of our mathematical model:

$$\text{Min } z = \sum_{d \in D} \sum_{t \in T} (c_{dt1}Z_{dt1} + c_{dt2}Z_{dt2} + c_{dt3}Z_{dt3} + c_{dt4}A_{dt} + c_{dt5}B_{dt}) \quad (1)$$

The use of more classrooms, lecture halls and laboratories than specified by the operational constraints is allowed in our model. However, the number of physical spaces to be used is minimized in the objective function given in Eq. (1) by multiplying the deviation variables that will change according to the number of extra classrooms, lecture halls and laboratories used, and the penalty costs determined.

At the same time, the overlapping of the compulsory courses of consecutive classes, and the absence of the time intervals of the lectures are prevented by the penalty costs and deviation variables in the objective function.

The constraints of the developed integer programming model are as follows.

$$\sum_d \sum_t X_{idt} = w_i \quad i \in I_{s2} \quad (2)$$

$$\sum_d \sum_t X_{idt} = w_i \quad i \in I_{s3} \quad (3)$$

$$X_{idt} + X_{i'dt} \leq 1, \quad i \in I_r, \quad i' \in I_r, \quad d \in D, \quad t \in T \quad (4)$$

$$X_{idt} + X_{i'dt} \leq 1 + A_d, \quad i \in I_r, \quad i' \in I_r, \quad d \in D, \quad t \in T \quad (5)$$

$$X_{idt} + X_{i'dt} \leq 1, \quad i \in I_h, \quad i' \in I_h, \quad d \in D, \quad t \in T \quad (6)$$

$$X_{idt} = 0, \quad i \in I_h, \quad d \in D, \quad t \in \alpha_0 \quad (7)$$

$$X_{idt} + B_{dt} \leq X_{idt-1} + X_{idt+1}, \quad i \in I_{s2}, \quad d \in D, \quad 2 \leq t \leq n-1 \quad (8)$$

$$X_{idt} - X_{idt+1} \leq 0 + B_{dt}, \quad i \in I_{s2}, \quad d \in D, \quad t = 1 \quad (9)$$

$$X_{idt} - X_{idt-1} \leq 0 + B_{dt}, \quad i \in I_{s2}, \quad d \in D, \quad t = n \quad (10)$$

$$X_{idt} + B_{dt} \leq X_{idt-1} + X_{idt+2}, \quad i \in I_{s3}, \quad d \in D, \quad 2 \leq t \leq n-2 \quad (11)$$

$$X_{idt} - X_{idt+2} \leq 0 + B_{dt}, \quad i \in I_{s3}, \quad d \in D, \quad t = 1 \quad (12)$$

$$-X_{idt-1} + X_{idt} \leq 0 + B_{dt}, \quad i \in I_{s3}, \quad d \in D, \quad t = n \quad (13)$$

$$\sum_i i \in I_{D1} X_{idt} \leq p + Z_{dt1}, \quad d \in D, \quad t \in T \quad (14)$$

$$\sum_i i \in I_{D2} X_{idt} \leq j + Z_{dt2}, \quad d \in D, \quad t \in T \quad (15)$$

$$\sum_i i \in I_{D3} X_{idt} \leq k + Z_{dt3}, \quad d \in D, \quad t \in T \quad (16)$$

$$X_{idt} \leq 0, \quad i \in I, \quad d \in D, \quad t \in T, \quad > 4 \quad (17)$$

$$X_{idt} \leq 0, i \in I, d \in D, t \in T, < 4 \quad (18)$$

Constraints (2) and (3) are used to assign the lectures to the time slots. There are 4 levels of undergraduate classes: 1, 2, 3 and 4. Constraint (4) is used so that the lectures allocated to the same classroom are not scheduled within the same time frame. Compulsory courses of consecutive classes should not be scheduled in the same time slot. Constraint (5) is used to prevent this conflict. It is ensured by constraint (6) that the lectures of the faculty members do not overlap. Instructors also teach postgraduate courses. Undergraduate courses and postgraduate courses should not overlap, also. For this reason, it is prevented from assigning lectures to these time intervals with constraint (7). The consecutiveness of the two-hour lectures is provided by constraints (8–10). Similarly, the consecutiveness of the three-hour lectures is provided by constraints (11–13).

The number of classrooms to be used is expressed with parameter p , the number of lecture halls with j , and the number of laboratories with k . However, it is allowed to use more classrooms, lecture halls and laboratories by the model, and the Z deviation variable takes positive value in this case. This is ensured through constraints numbered (14–16), and the deviation is also reflected in the objective function. The assignment of the theoretical lectures of the courses before the practice or lab lectures within the week is provided by constraints (17) and (18).

Separate similar mathematical models were created for the course scheduling problem for online and face-to-face courses. Both models were coded in OPL programming language in solved in IBM ILOG CPLEX Optimization Studio, including also the sign constraints for the decision variables. The optimal solution was obtained using CPLEX solver.

3 Decision Support System

A user-friendly interface has been developed using Excel VBA, so that users can obtain solutions without expert knowledge of the developed mathematical model.

The first step in the interface is to choose the way the lectures are taught. This selection is made with two buttons, “YÜZ YÜZE” and “ONLINE”, from the window in Fig. 1. With these buttons, it is possible to determine the mathematical model to be used according to the selection made, as 30% of the lectures are taught face-to-face and 70% online.

The user is also allowed to choose the days for these different forms of lectures. After the days are selected, the number of rooms of each category are entered, as can be seen in Fig. 1.

Next, the user enters the data using the ‘YÜZ YÜZE/ONLINE PROGRAM OLUŞTUR’ window (Fig. 2). In this form, the user selects the semester and the courses for that semester are chosen automatically from the existing curriculum of the department. The user is also allowed to define new courses and add them to the curriculum, as well as new instructors. The added data are recorded in the database. It is also possible to delete courses or faculty members from the database using the same form. By using the “MÜSAİTLİK DURUMU” button, the unavailable hours of the lecturers in the department are recorded, and these hours are eliminated from consideration by the

Fig. 1. Data entry for mode of lectures and lecture days

model through the inclusion of additional constraints. When filling the form, all necessary notifications, warnings and instructions are shared with the user as necessary. By clicking the ‘KAYDET’ button on the form in Fig. 2, the entered data is recorded in an MS Excel database. The data processed in Excel, the mathematical model is formulated automatically, and the created model is transferred to IBM ILOG CPLEX Solver with the ‘PROGRAM OLUŞTUR’ button.

Fig. 2. Data entry for courses and instructors

As an output, the optimal result from the CPLEX Solver is presented to the user as a weekly course timetable in MS Excel environment. The format of the generated course timetable is illustrated in Fig. 3. The example includes the output of the face-to-face course timetabling model. The output format is in line with the requirements of the

university. When the results were obtained for the 2021–2022 Spring semester, it was seen that the number of classrooms used for conducting all lectures for this semester was minimized, and all the special requests of the instructors were satisfied at the same time. The model was solved in approximately 2.5 min using IBM ILOG CPLEX Optimization Studio on a PC with Intel Core 7, 2.80 GHz processor and 8 GB RAM. In accordance with the aim of minimizing the number of physical spaces, an optimal result was obtained by using 1 classroom, 1 lecture hall and 1 laboratory.

III. SINIF	PZT	SALI	ÇARŞAMBA	PERŞEMBE	CUMA
09:00-09:40		END355-Karar Verme Teknikleri(T)	END351-Yoneylem Arastirmasi III(T)		
10:00-10:40		END355-Karar Verme Teknikleri(T)	END351-Yoneylem Arastirmasi III(T)		
11:00-11:40		END355-Karar Verme Teknikleri(T)			
ÖĞLE ARASI					
12:20-13:00			END351-Yoneylem Arastirmasi III(U)		
13:20-14:00			END351-Yoneylem Arastirmasi III(U)		
14:20-15:00			END313-Muhendislikte Hesaplama Tablosu Uygulamalari(T)		
15:20-16:00		END331-Modelleme ve Simulasyon(U)	END313-Muhendislikte Hesaplama Tablosu Uygulamalari(T)	END331-Modelleme ve Simulasyon(T)	
16:20-17:00		END331-Modelleme ve Simulasyon(U)	END313-Muhendislikte Hesaplama Tablosu Uygulamalari(T)	END331-Modelleme ve Simulasyon(T)	

Fig. 3. Output format

4 Conclusion

In this study, the course timetabling problem of Izmir Bakırçay University Industrial Engineering Department is handled. Since this university is a newly established one, an automatic course timetabling system is not used. Due to this deficiency, the process of preparing the lecture schedule takes quite a long time.

A mathematical model was developed to cover all restrictions for solving the problem optimally. A user-interactive interface that works synchronously with IBM ILOG CPLEX Optimization Studio was also designed using MS Excel VBA environment. In line with the data entered through the user interface, CPLEX Solver is used to obtain an optimal solution for the developed model, and the results are presented to the user in the form of a course timetable automatically via VBA. The developed system is highly user-friendly and obtains solutions in a matter of minutes.

Manual methods for creating such optimal schedules and meeting special requests were hardly possible before. The production of optimal timetables was not guaranteed while the process took much longer. Therefore, the development of this decision support tool, which obtains optimal timetables in mere minutes is extremely beneficial for the institution, saving both time and effort.

The decision support tool designed in this study was only used for the Department of Industrial Engineering as a case study. However, the tool is flexible enough to include other departments, if not the whole university. However, the solution times are expected to increase considerably as the problem instance sizes get larger. For this reason, the tool is open for development. In order to be efficiently extended across the engineering faculty or all faculties, heuristic or metaheuristic approaches would be necessary in addition to the developed mathematical model for solving the course timetabling problem.

In its current form, the data to be used in generating timetables is recorded in and extracted from MS Excel. For a more professional use of the developed software, the

necessary data including course and academician information can be pulled from the existing database used by the university. The output can also be fed to the university information management system.

Acknowledgments. We would like to express our special thanks to our department research assistants Pınar Yunusoğlu and Semiha Küsbeci, who guided us in our coding endeavors for realizing this project in our university. We also thank our department for supplying us with necessary data.

References

- Öztürk, Z.K., et al.: Kullanıcı Tercihlerinin Dikkate Alınması Durumunda Üniversite Ders Çizelgeleme Problemi. *Endüstri Mühendisliği* **27**, 2–16 (2016)
- Köçken, H., Özdemir, R., Ahlatcıoğlu, M.: Üniversite Ders Zaman Çizelgeleme Problemi İçin İkili Tamsayılı Bir Model Ve Bir Uygulama. *İstanbul Üniversitesi İşletme Fakültesi Dergisi* **43**, 28–54 (2014)
- Uçar, U., İşleyen, S., Demir, Y.: Üniversite Ders Çizelgeleme Probleminin Bulanık Ahp Ve Çok Amaçlı Karışık Tam Sayılı Matematiksel Modelle Çözümü. *Gazi University Journal of Science Part C: Design and Technology* **3**, 513–523 (2015)
- Can, E.: Çok Amaçlı Ders Çizelgeleme Probleminin Çözümü İçin Metasezgisel Yaklaşım Önerisi. Ms Thesis, Kütahya Dumlupınar Üniversitesi, Kütahya, Türkiye (2019)
- Altunay, H., Eren, T.: Ders Programı Çizelgeleme Problemi İçin Bir Literatür Taraması. *Pamukkale Univ. J. Eng. Sci.* **23**, 55–70 (2017)
- İ. Ertuğrul ve G. Öztaş. “Ders Programı Oluşturulmasında 0–1 Tam Sayılı Bulanık Hedef Programlama Yaklaşımı”, *Niğde Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, vol. 9, pp. 159–177 (2016)
- Yiğit, T.: Meslek Liseleri Haftalık Ders Çizelgelerinin Genetik Algoritmalar Yardımıyla Oluşturulması. *Gazi Üniversitesi Endüstriyel Sanatlar Eğitim Fakültesi Dergisi* **19**, 25–39 (2006)
- Akay, B.: Nümerik Optimizasyon Problemlerinde Yapay Arı Kolonisi (Artificial Bee Colony) Algoritmasının Performans Analizi. PhD Thesis, Erciyes Üniversitesi, Kayseri, Türkiye (2009)
- Geçici, E., Güler, M.G.: Ders Çizelgeleme Problemi İçin Elektronik Tablo Tabanlı Karar Destek Sistemi. *Konya Mühendislik Bilimleri Dergisi* **9**, 551–567 (2021)
- Demir, Y., Çelik, C.: Müfredat Bazlı Akademik Zaman Çizelgeleme Probleminin Çözümüne Tam Sayılı Doğrusal Programlama Yaklaşımı”, *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, vol. 31, pp. 145–149 (2016)
- Özkan, A.: Üniversite Ders Çizelgeleme Probleminin Tamsayılı Doğrusal Programlama Ve Sezgisel Yaklaşımlar İle Çözümü. PhD Thesis, Hacettepe University, Ankara, Türkiye (2019)
- Gerşil, M., Palamutçuoğlu, T.: Ders Çizelgeleme Probleminin Melez Genetik Algoritmalar İle Performans Analizi. *Niğde Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi* **6**, 242–262 (2013)
- Özdağ, H., Aygör, N., Parlak, A.: Karınca Kolonisi Algoritmasının Zaman Çizelgelemesi Üzerine: Bir Modellemesi ve Uygulaması. *Akademik Bilişim* **12**, 01–03 (2012)
- Daban, F., Ozdemir, E.: Eğitimde Verimliliği Artıran Ders Programlarının Hazırlanması İçin Genetik Algoritma Kullanımı. *J. Educ. Sci. Pract.* **3**, 245–257 (2004)
- Yurtsal, A., Kaynar, O.: Ders Programı Çizelgeleme Probleminin Genetik Algoritma ile Optimizasyonu. *J. Inform. Syst. Manage. Res.* **1**, 9–14 (2019)

16. Özdemir, R.: Üniversite Ders Zaman Çizelgeleme Problemi. Ms Thesis, Yıldız Teknik Üniversitesi, İstanbul, Türkiye (2012)
17. Kalender, M.: Ders Çizelgeleme Programı. TMMOB Elektrik Mühendisleri Odası İstanbul Şubesi 2006–2007 Öğretim Yılı Proje Yarışması (2007)
18. Socha, K., Knowles, J., Sampels, M.: A max-min ant system for the university course timetabling problem. In: Dorigo, M., Di Caro, G., Sampels, M. (eds.) Ant Algorithms, pp. 1–13. Springer, Berlin (2002). https://doi.org/10.1007/3-540-45724-0_1
19. Malim, M.R., Khader, A.T., Mustafa, A.: Artificial immune algorithms for university timetabling. In: Proceedings of the 6th International Conference on Practice and Theory of Automated Timetabling, pp. 234–245, Brno, Czech Republic (2016)