A mathematical formulation for constructing feasible solutions for the Post Enrollment Course Timetabling Problem

Angelos Dimitsas*, Vasileios Nastos*, Christos Valouxis† and Christos Gogos*
*Dept. of Informatics & Telecommunications, University of Ioannina, Arta, Greece
Email: {a.dimitsas, vnastos, cgogos}@uoi.gr
†Dept. of Electrical & Computer Engineering, University of Patras, Rio, Greece
Email: cvalouxis@upatras.gr

2022 7th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM) | 979-8-3503-9858-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/SEEDA-CECNSM57760.2022.9932946

***Abstract*—The scheduling community has long been interested in educational timetabling. Particularly in academia, since timetabling dictates the day to day operation of Universities, great effort has been exercised to produce high quality schedules. Typically, timetabling problems are NP-Hard and several approaches have been tried in order to generate schedules that satisfy all stakeholders. A number of timetabling competitions have been organized through the last two decades focusing on problems stemming from educational operations. In this paper we use data from two such competitions, ITC2002 and ITC2007 about the post enrollment course timetabling problem. We propose a mathematical model that captures the problem in its entirety and we use it in order to construct feasible solutions initially, and then explore the prospect of optimization. We employ a pre-process stage that attempts to reduce the size of the model and then use an open source solver, that produces solutions in reasonable time for most of the cases. We also propose a simple decomposition of the problem in a day by day basis that can improve the initial feasible solutions.**

I. Introduction

Several timetabling problems arise in educational institutions. Such problems are high school timetabling, examination timetabling, course timetabling, thesis timetabling and others. Timetable construction is notoriously difficult and time consuming in Universities and High Schools. Moreover, the details of each problem differ among institutions worldwide. For example, in certain cases, room and lecturer availabilities are a major concern while in other cases course precedences, timeslot availabilities and other constraints might be more important. For most of the educational timetabling problems, at the center of attention are the students. So, the resulting timetable should facilitate students' studies or examinations by providing balanced timetables with appropriate breaks and enough time for study and rest. The main concern is to create feasible timetables, meaning that situations that might leave certain courses unscheduled, or timetables that might put two courses that are attended by a student at the same time are unacceptable. Feasible timetables are diversely perceived, their assessment depends on quality metrics that may vary from institution to institution. These desires result in a multitude of difficult optimization problems.

979-8-3503-9858-8/22/$31.00 ©2022 IEEE

In this paper, we study the post enrollment course timetabling (PE-CTT) problem that aims at scheduling a set of events. An event is teaching of a course to a set of students at a room that possesses certain features, during a timeslot. In PE-CTT, the events that each student attends are known in advance. Moreover, precedence among events, event-timeslot availabilities and requirements about certain rooms' capacities and features if they exist are also known.

The structure of the paper follows. In Section II we provide the description of the problem. In Section III related work is presented, while in Section IV preprocessing tasks that create more compact versions of the problem instances are described. Then, in Section V we present characteristics of the problem instances that we have used. Section VI contains the proposed mathematical model, which is the main contribution of the paper and Section VII presents results produced by experimenting with this mathematical model. Finally, Section VIII describes the idea of reusing the mathematical model in a day by day basis to produce better quality results.

II. Problem Description

Course timetabling is a well known timetabling problem with practical importance and theoretical interest. Two variants of the problem are Curriculum Based Course Timetabling (CB-CTT) and Post Enrollment Course Timetabling (PE-CTT). In CB-CTT, students are "hidden" behind the courses that they attend, while in PE-CTT each student's enrollments are known. The PE-CTT variant of the problem, that is the focus of this paper, has the goal of optimally placing events in the available timeslots and rooms. For all instances, the total number of timeslots is 45, since the time horizon is 5 days and each day consists of 9 timeslots.

Various problem instances for Course Timetabling have been used over the years, adhering several assumptions about the problem itself. Fortunately, the International Timetabling Competition that was held in 2002 and its sequel in 2007 [1] provided a set of instances that were later used by several researchers as a common testbed. These are the datasets, ITC2002 and ITC2007, that we use hereafter.

For these datasets the hard constraints are:

- Each event should be scheduled to a timeslot.
- No student can attend more than one event simultaneously.
- A room can host a single event in each timeslot.
- A room must satisfy the capacity and feature requirements of the event that it hosts.
- Events may have specific timeslot requirements.
- Precedence relations between events might exist.

Note that, the two last hard constraints of the above list manifest themselves only in problem instances of ICT2007 and not in ITC2002. On the other hand, the soft constraints are:

- A student should not attend events at the last timeslot of a day.
- A student should not attend three (or more) events in a row in the same day.
- A student should not attend single events during a day.

Each violation of the above prepositions is penalized by 1-point, and each point is associated with a specific student.

III. Related Work

Educational timetabling problems have attracted interest from the academia since such problems stem from needs of universities and other educational institutes. Implicit familiarity makes them easily understandable. Main educational timetabling problems are examination timetabling [2], course timetabling [3], high school timetabling [4] and their variants [5]. Moreover, several other timetabling problems arise at the educational context, like thesis defense timetabling [6], invigilator duty allocation [7] and others.

Recent surveys for educational timetabling have been published by Tan et al. [8] and by Ceschia et al. [9] that complement existing surveys on the field [10].

For the version of the PE-CTT problem that we study, a plethora of papers were published during and after the competitions. The interest in this problem remains strong with recent publications proposing various methodologies. Recent papers focusing on local search methods are the papers by Goh et al. [11] [12] and Nagata et al. [13]. Metaheuristics like Simulated Annealing were also effective in tackling the problem by Ceschia et al. [14]. A Constraint Programming approach was presented by Cambazard et al. [15].

Lewis et al. [16] explored the connectivity of the solution space in course timetabling problems under various neighborhood operators.

IV. Preprocessing

The format of the problem instances dictates rooms with varying capacities alongside with features that each room has (e.g. video projector, smart-board, laboratory equipment etc.). It also includes for each student the set of events they have to attend and for events the feature requirements that each event might have. In ITC2007 dataset, events can additionally have timeslot restrictions (i.e. a timeslot might be prohibited for certain events) and precedence relations (i.e. an event may be required to take place earlier than another event). We investigated the possibility of reducing the size of problem instances by identifying students that share the same set of events. Exploiting this information might lead to models with smaller sizes and therefore easier to be solved. The result of our exploration follows.

A. Identical students

We consider students to be "identical" when their enrollments share the same set of events. More formally, let $\mathbb{S}$ be the set of students and $\mathbb{E}_s$ be the set of events that student s attends. For every pair of students s_1 and s_1 if $\mathbb{E}_{s_1} \equiv \mathbb{E}_{s_2}$, then those students have to attend the same events. Since soft constraints are expressed according to students, these students should be able to consolidate in the objective function. Dataset ITC2002 includes only two problem instances with identical students (o04.tim and o08.tim), while the majority of problem instances in ITC2007 have identical students.

B. Event-Room eligibility

We assume that $\mathbb{E}$ is the set of all events. Equation 1 dictates when a student attends an event, so the total number of attendees of each event is given by Equation 2. This information is extracted by problem data.

$$a_{se} = \begin{cases} 1 & \text{if student } s \text{ attends event } e \\ 0 & \text{otherwise} \end{cases} \quad \forall s \in \mathbb{S}, \quad \forall e \in \mathbb{E} \tag{1}$$

$$S_e = \sum_{s \in \mathbb{S}} a_{se} \quad \forall e \in \mathbb{E} \tag{2}$$

Let C_r be the capacity of room r, and $\mathbb{F}_r$ be the set of features that the same room has. Also, let $\mathbb{G}_e$ be the set of features that are required by event e. Then, event e can be hosted in room r if and only if $C_r \leq S_e$ and $\mathbb{G}_e \subseteq \mathbb{F}_r$. These relations are used in pre-processing so as to feed the solver only with valid combinations of events to rooms.

C. Event Conflicts

By definition no student can attend more than one event at the same time, i.e. events with common students are banned from the same timeslot. We consider such pairs of events as conflicting events. Let $\mathbb{R}$ be the set of all rooms, and $\mathbb{R}_e$ be the set of rooms that can host event e. We augment the set of conflicting pairs of events by incorporating into it pairs without common students if for two events e_1 and e_2 the relations $\mathbb{R}_{e_1} \equiv \mathbb{R}_{e_2}$ and $|\mathbb{R}_{e_1}| = |\mathbb{R}_{e_2}| = 1$ hold true (singleton sets). This condition means that e_1 and e_2 can be hosted only in the same room. Since the room can host at most one event, we can add the pair of events to the set of conflicting events. Finally, the set of all pairs of conflicting events forms set $\mathbb{C}$. Based on the event conflicts we compute conflict density of each problem instance which is twice the size of set $\mathbb{C}$ divided by the square of the size of set $\mathbb{E}$. Conflict density can be considered as a measure of the difficulty that each problem instance exhibits.

V. Datasets

Details about problem instances belonging to datasets ITC2002 and ITC2007 are presented in TABLE I and TABLE II respectively. The former table has two less columns than the later, since problem instances of ITC2002 have neither event-precedence relations nor event-timeslot restrictions.

TABLE I
DATASET ITC2002

Instance	Events	Rooms	Features	Students	Conflict Density
o01.tim	400	10	10	200	0.2025
o02.tim	400	10	10	200	0.2075
o03.tim	400	10	10	200	0.2342
o04.tim	400	10	5	300	0.2257
o05.tim	350	10	10	300	0.3100
o06.tim	350	10	5	300	0.2599
o07.tim	350	10	5	350	0.2060
o08.tim	400	10	5	250	0.1696
o09.tim	440	11	6	220	0.1705
o10.tim	400	10	5	200	0.2006
o11.tim	400	10	6	220	0.2045
o12.tim	400	10	5	200	0.2039
o13.tim	400	10	6	250	0.2079
o14.tim	350	10	5	350	0.2454
o15.tim	350	10	10	300	0.2469
o16.tim	440	11	6	220	0.1802
o17.tim	350	10	10	300	0.3084
o18.tim	400	10	10	200	0.2099
o19.tim	400	10	5	300	0.2017
o20.tim	350	10	5	300	0.2458

TABLE II
DATASET ITC2007

Instance	Events	Rooms	Features	Students	Conflict Density	Precedence	Timeslot restrictions
i01.tim	400	10	10	500	0.3409	40	7863
i02.tim	400	10	10	500	0.3735	36	7724
i03.tim	200	20	10	1000	0.4724	20	3893
i04.tim	200	20	10	1000	0.5173	20	3867
i05.tim	400	20	20	300	0.3074	120	7830
i06.tim	400	20	20	300	0.3021	119	7843
i07.tim	200	20	20	500	0.5310	20	5428
i08.tim	200	20	20	500	0.5149	21	5566
i09.tim	400	10	20	500	0.3400	41	7833
i10.tim	400	10	20	500	0.3834	40	7813
i11.tim	200	10	10	1000	0.4980	21	3936
i12.tim	200	10	10	1000	0.5813	20	3866
i13.tim	400	20	10	300	0.3227	116	7699
i14.tim	400	20	10	300	0.3201	118	7824
i15.tim	200	10	20	500	0.5339	21	5525
i16.tim	200	10	20	500	0.4536	19	5486
i17.tim	100	10	10	500	0.6986	11	1927
i18.tim	200	10	10	500	0.6480	20	3866
i19.tim	300	10	10	1000	0.4698	31	5891
i20.tim	400	10	10	1000	0.2780	40	7863
i21.tim	500	20	20	300	0.2316	147	9518
i22.tim	600	20	20	500	0.2609	176	11702
i23.tim	400	20	30	1000	0.4390	41	3869
i24.tim	400	20	30	1000	0.3111	197	7897

VI. Mixed Integer Programming Formulation

In Section IV we defined S_e and sets $\mathbb{S}$, $\mathbb{E}$, $\mathbb{R}$, $\mathbb{R}_e$, $\mathbb{C}$. In order to formulate our model, we need a few more definitions. So, let $\mathbb{P}$ be the set of pairs of events having a precedence relation, $\mathbb{T}$ be the set of timeslots and $\mathbb{T}_e$ be the set of timeslots that event e can possibly occur. Furthermore, let $\mathbb{L} = \{9, 18, 27, 36, 45\}$ be the set of timeslots formed by the ordinal numbers of the last timeslot of each day, and let $\mathbb{E}_s$ be the set of events that student s has to attend.

Equation 6 is the objective function that captures costs corresponding to the three soft constraints referred in Section II. The first term incurs penalty S_e for each event e that is scheduled at the last timeslot of a day, i.e. a timeslot in $\mathbb{L}$. The second term, incurs a penalty of 1 point for each day and for each student that attends a single event in this day. The last term of the objective function, incurs penalty of m points for each time that a student has to attend m consecutive events in a day.

The main binary decision variables are defined in Equation 3 and represent the room and timeslot that each event will eventually be scheduled. Secondary decision variables that are defined in Equations 4 and 5 play the role of detecting single event days and consecutive events respectively.

The constraint in Equation 7 forces each event to be scheduled in exactly one room and one timeslot. The constraint in Equation 8 guarantees that at most one event will be scheduled in each room in each timeslot. These two constraints might seem similar, but note that the second one focuses on rooms, disallowing more than one events to be scheduled in the same room for each timeslot. Next, the constraint in Equation 9 forbids conflicting events, as defined in set $\mathbb{C}$, to be scheduled simultaneously. The constraint in Equation 10 ensures that the order of events at the resulting schedule will satisfy precedence relations. For each precedence restriction between events (e.g. event e_1 to occur before event e_2), the left term of the inequality corresponds to the period plus one that e_1 will be scheduled and the right term corresponds to the period that e_2 will be scheduled. Finally, Equations 11 and 12 ensure that penalty points will be imposed for decision variables y_{sd} and z_{sdm} respectively. In particular, Equation 11 forces y_{sd} to assume value 1 when an event that student s attends is scheduled at timeslot t' and no other events that student s attends are scheduled at the rest timeslots of the same day, d. Likewise, Equation 12 needs m consecutive events to enforce z_{sdm} to assume value 1.

Decision Variables

$$x_{etr} = \begin{cases} 1 & \text{if } e \text{ is scheduled on } t \text{ and } r \\ 0 & \text{otherwise} \end{cases} \qquad \forall e \in \mathbb{E}, \forall t \in \mathbb{T}_e, \forall r \in \mathbb{R}_e \quad (3)$$

$$y_{sd} = \begin{cases} 1 & \text{if } s \text{ has a single event day in } d \\ 0 & \text{otherwise} \end{cases} \qquad \forall s \in \mathbb{S}, \forall d \in [1..5] \quad (4)$$

$$z_{sdm} = \begin{cases} 1 & \text{if } s \text{ has } m \text{ consecutive events in } d \\ 0 & \text{otherwise} \end{cases} \qquad \forall s \in \mathbb{S},\ \forall d \in [1..5],\ \forall m \in [3..9] \quad (5)$$

$$\text{Minimize} \sum_{e \in \mathbb{E}} \sum_{t \in \mathbb{L}} \sum_{r \in \mathbb{R}_e} S_e * x_{etr} + \sum_{s \in \mathbb{S}} \sum_{d=1}^{5} y_{sd} + \sum_{s \in \mathbb{S}} \sum_{d=1}^{5} \sum_{m=3}^{9} m * z_{sdm} \quad (6)$$

Subject to

$$\sum_{t\in\mathbb{T}_e}\sum_{r\in\mathbb{R}_e} x_{etr} = 1 \quad \forall e\in\mathbb{E} \tag{7}$$

$$\sum_{e\in\mathbb{E}, r\in\mathbb{R}_e} x_{etr} \leq 1 \quad \forall r \in \mathbb{R}, \quad \forall t \in \mathbb{T} \tag{8}$$

$$\sum_{r\in\mathbb{R}_{e_1}} x_{e_1tr} + \sum_{r\in\mathbb{R}_{e_2}} x_{e_2tr} \leq 1 \quad \forall (e_1,e_2)\in\mathbb{C}, \quad \forall t\in\mathbb{T} \tag{9}$$

$$\sum_{t\in\mathbb{T}_{e_1}}\sum_{r\in\mathbb{R}_{e_1}} t * x_{e_1tr} + 1 \leq \sum_{t\in\mathbb{T}_{e_2}}\sum_{r\in\mathbb{R}_{e_2}} t * x_{e_2tr} \quad \forall (e_1,e_2)\in\mathbb{P} \tag{10}$$

$$1 - \sum_{e\in\mathbb{E}_s}\sum_{r\in\mathbb{R}} x_{et'r} + \sum_{e\in\mathbb{E}_s}\sum_{t\in[9(d-1)+1..9d]\setminus k}\sum_{r\in\mathbb{R}} x_{etr} + y_{sd} \geq 1$$
$$\text{where } t' = 9(d-1)+1+k, \quad \forall s\in\mathbb{S}, \forall d\in[1..5], \forall k\in[1..9] \tag{11}$$

$$\sum_{i=1}^{m}(1 - \sum_{e\in\mathbb{E}_s}\sum_{r\in\mathbb{R}} x_{e(t'+i)r}) +$$
$$\sum_{e\in\mathbb{E}_s}\sum_{t\in[9(d-1)+1..9d]\setminus[k..k+m]}\sum_{r\in\mathbb{R}} x_{etr} + z_{sdm} \geq 1$$
$$\text{where } t' = 9(d-1)+1+k, \quad \forall s\in\mathbb{S}, \quad \forall d\in[1..5],$$
$$\forall m\in[3..9], \forall k\in[1..9] : k+m\leq 9 \tag{12}$$

The above formulation aims to solve the full problem and succeeds for moderate problem sizes. Unfortunately, the sizes of problem instances in ITC2002 and ITC2007 inhibit current state of the art mixed integer programming solvers from efficiently constructing the model that will be then solved. Nevertheless, we can use part of the model by keeping only Equations 3, 7, 8, 9 and 10. Thus, the model looses its objective, and effectively becomes a satisfiability formulation that has the potential of finding feasible solutions in problem instances of relatively big sizes.

VII. Results

The satisfiability portion of the mathematical model described in Section VI was benchmarked for all instances of datasets ITC2002 and ITC2007. We tried several Linear/Integer Programming and Constraint Programming solvers including IBM's CPLEX solver (both the IP solver and the CP solver), the Gurobi solver and the open source Google ORTools [17] IP solver and CPSAT solver, all with default settings. All solvers were competitive with each other. For the rest of the paper we present results of experiments using ORTools CPSAT. We allowed the solver to execute for 10 minutes per instance and our experiments were performed in a workstation with 32GB RAM and 8 cores, 16 threads. Table III presents the results that show the time needed to reach a feasible solution, or N/A (Not Available) when no such solution was found during the available execution time. Overall, ORTools CPSAT was able to find solutions for all 20 ITC2002 instances and for 18 out of the 24 ITC2007 instances.

TABLE III
Time needed to reach a feasible solution, within 600 seconds

ITC2002	Solution time (sec)	ITC2007	Solution time (sec)
o01.tim	112	i01.tim	442
o02.tim	96	i02.tim	N/A
o03.tim	174	i03.tim	33
o04.tim	133	i04.tim	46
o05.tim	89	i05.tim	112
o06.tim	151	i06.tim	90
o07.tim	83	i07.tim	20
o08.tim	99	i08.tim	13
o09.tim	115	i09.tim	N/A
o10.tim	146	i10.tim	N/A
o11.tim	95	i11.tim	26
o12.tim	81	i12.tim	86
o13.tim	104	i13.tim	400
o14.tim	106	i14.tim	50
o15.tim	87	i15.tim	16
o16.tim	134	i16.tim	11
o17.tim	72	i17.tim	9
o18.tim	80	i18.tim	50
o19.tim	154	i19.tim	N/A
o20.tim	133	i20.tim	70
		i21.tim	161
		i22.tim	N/A
		i23.tim	N/A
		i24.tim	43

VIII. Future work

Our exploration of the problem showed that big problems were unable to be solved when formulas with numerous terms like the objective function was included. Nevertheless, reaching feasibility was relatively easy for most of the problem instances that we have tested. We expect the full model described in Section VI to work as an effective large neighborhood search move, capable of exploring the search space in dimensions that typical heuristic moves cannot. We plan to investigate the opportunity of exploiting the concept of identical students to reduce the number of terms in the objective function formula. This should speed up the large neighborhood search execution time.

Another idea is to gradually solve, using the full model, the problem day by day, meaning that all decision variables that refer to a certain day could be allowed to assume values that would minimize the partial objective function. Once a solution to the day under optimization is found, the approach should proceed to the next day. This could occur in a circular way, possibly involving combinations of days instead of single days. We can use as initial solutions, the feasible solutions that are produced by the satisfiability portion of the model, and/or use constructive heuristics for problem instances that the satisfiability model struggles to find feasible solutions.

IX. Conclusion

In this work, we approached the Post Enrollment Course Timetabling problem by proposing a mathematical formulation. Once a feasible solution is attained, the degree of violation of soft constraints determines the quality of the solution. Soft constraints involve consecutive events, events scheduled at the last timeslot of each day, and precedence relations suggesting that certain events should come before other events. Since, each individual student may contribute to the penalties, the objective function and the constraints that define auxiliary decision variables used in the objective function become unwieldy expressions constituted by numerous terms. Nevertheless, our mathematical model has the potential of contributing feasible and possibly good solutions. The former was validated against a number of problem instances of two public datasets, while the latter can be integrated in hybrid approaches that combine heuristics and metaheuristics contributing a large neighborhood search move.

Acknowledgment

This research was supported by project "Dioni: Computing Infrastructure for Big-Data Processing and Analysis" (MIS No. 5047222) co-funded by European Union (ERDF) and Greece through Operational Program "Competitiveness, Entrepreneurship and Innovation", NSRF 2014-2020.

References

[1] L. Di Gaspero, B. McCollum, and A. Schaerf, "The second international timetabling competition (itc-2007): Curriculum-based course timetabling (track 3)," Technical Report QUB/IEEE/Tech/ITC2007/CurriculumCTT/v1. 0, Queen's ..., Tech. Rep., 2007.

[2] C. Gogos, P. Alefragis, and E. Housos, "An improved multi-staged algorithmic process for the solution of the examination timetabling problem," *Annals of Operations Research*, vol. 194, no. 1, pp. 203–221, 2012.

[3] L. D. Gaspero and A. Schaerf, "Multi-neighbourhood local search with application to course timetabling," in *International Conference on the Practice and Theory of Automated Timetabling*. Springer, 2002, pp. 262–275.

[4] C. Valouxis, C. Gogos, P. Alefragis, and E. Housos, "Decomposing the high school timetable problem," *Practice and Theory of Automated Timetabling (PATAT 2012), Son, Norway*, vol. 61, 2012.

[5] M. Chiarandini, L. Di Gaspero, S. Gualandi, and A. Schaerf, "The balanced academic curriculum problem revisited," *Journal of Heuristics*, vol. 18, no. 1, pp. 119–148, 2012.

[6] M. Battistutta, S. Ceschia, F. De Cesco, L. Di Gaspero, and A. Schaerf, "Modelling and solving the thesis defense timetabling problem," *Journal of the Operational Research Society*, vol. 70, no. 7, pp. 1039–1050, 2019.

[7] M. Kahar and G. Kendall, "Universiti malaysia pahang examination timetabling problem: scheduling invigilators," *Journal of the Operational Research Society*, vol. 65, no. 2, pp. 214–226, 2014.

[8] J. S. Tan, S. L. Goh, G. Kendall, and N. R. Sabar, "A survey of the state-of-the-art of optimisation methodologies in school timetabling problems," *Expert Systems with Applications*, vol. 165, p. 113943, 2021.

[9] S. Ceschia, L. Di Gaspero, and A. Schaerf, "Educational timetabling: Problems, benchmarks, and state-of-the-art results," *arXiv preprint arXiv:2201.07525*, 2022.

[10] R. Qu, E. K. Burke, B. McCollum, L. T. Merlot, and S. Y. Lee, "A survey of search methodologies and automated system development for examination timetabling," *Journal of scheduling*, vol. 12, no. 1, pp. 55–89, 2009.

[11] S. L. Goh, G. Kendall, and N. R. Sabar, "Simulated annealing with improved reheating and learning for the post enrolment course timetabling problem," *Journal of the Operational Research Society*, vol. 70, no. 6, pp. 873–888, 2019.

[12] S. L. Goh, G. Kendall, N. R. Sabar, and S. Abdullah, "An effective hybrid local search approach for the post enrolment course timetabling problem," *Opsearch*, vol. 57, no. 4, pp. 1131–1163, 2020.

[13] Y. Nagata, "Random partial neighborhood search for the post-enrollment course timetabling problem," *Computers & Operations Research*, vol. 90, pp. 84–96, 2018.

[14] S. Ceschia, L. Di Gaspero, and A. Schaerf, "Design, engineering, and experimental analysis of a simulated annealing approach to the post-enrolment course timetabling problem," *Computers & Operations Research*, vol. 39, no. 7, pp. 1615–1624, 2012.

[15] H. Cambazard, E. Hebrard, B. O'Sullivan, and A. Papadopoulos, "Local search and constraint programming for the post enrolment-based course timetabling problem," *Annals of Operations Research*, vol. 194, no. 1, pp. 111–135, 2012.

[16] R. Lewis and J. Thompson, "Analysing the effects of solution space connectivity with an effective metaheuristic for the course timetabling problem," *European Journal of Operational Research*, vol. 240, no. 3, pp. 637–648, 2015.

[17] L. Perron and V. Furnon, "Or-tools." [Online]. Available: https://developers.google.com/optimization/