An exact based approach for the Post Enrollment Course Timetabling Problem

ANGELOS DIMITSAS, VASILEIOS NASTOS, and CHRISTOS GOGOS, Dept. of Informatics & Telecommunications, University of Ioannina, Greece

CHRISTOS VALOUXIS, Dept. of Electrical & Computer Engineering, University of Patras, Greece

Educational timetabling has long piqued the interest of the scheduling community. Since timetabling governs how universities operate daily, it has been vital in academia to put substantial effort into creating schedules of the highest caliber. To create schedules that satisfy all stakeholders, various approaches have been tried to solve educational timetabling problems, which are typically NP-Hard. Over the past 20 years, several scheduling contests have been held with a focus on issues related to educational operations. In this paper, we focus on the Post Enrollment Course Timetabling Problem and two datasets from the International Timetabling Competitions held in 2002 and 2007. We propose a local search procedure augmented by a base mathematical model and variations of this model that yields competitive results, within reasonable execution times, to some of the best-known solutions.

CCS Concepts: • **Applied Computing**; • **Operations research**; • **Computing methodologies**; • **Planning and scheduling**; • **Desicion support systems**;

Additional Key Words and Phrases: Mixed Integer Programming, Scheduling, Post Enrollment Course Timetabling Problem

ACM Reference Format:
Angelos Dimitsas, Vasileios Nastos, Christos Gogos, and Christos Valouxis. 2018. An exact based approach for the Post Enrollment Course Timetabling Problem. *J. ACM* 37, 4, Article 111 (August 2018), 10 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

In educational institutions, there are a number of scheduling problems that manifest themselves. Such scheduling problems are high school timetabling, examination timetabling, course timetabling, thesis timetabling, and others. Details of each problem differ among institutions worldwide. For example, in some situations, rooms and lecturer availabilities are the main problem, while in others, course precedences, timeslot restrictions and other restrictions might be more crucial. The focus of attention for most educational timetabling problems is the students. Therefore, the resulting timetable should facilitate students' studies or examinations with adequate breaks and enough time for studying and resting. The primary goal is to construct feasible timetables, so it is unacceptable to schedule two courses that are attended by a student at the same time or leave any event unscheduled. 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.

In this paper, we study the post-enrollment course timetabling (PE-CTT) problem that aims at scheduling a set of events. Each event takes place at a specific time and location with a particular set of students attending it. Moreover,

Authors' addresses: Angelos Dimitsas, a.dimitsas@uoi.gr; Vasileios Nastos, vnastos@uoi.gr; Christos Gogos, cgogos@uoi.gr, Dept. of Informatics & Telecommunications, University of Ioannina, Arta, Epirus, Greece, 47100; Christos Valouxis, Dept. of Electrical & Computer Engineering, University of Patras, Rio, Greece, cvalouxis@upatras.gr.

precedence relations among events, event-timeslot availabilities, and requirements about specific room capacities and features also exist in PE-CTT.

The structure of the paper follows. In section 2 a brief description of the problem and its constraints is presented. Section 3 related work is presented while, in section 4 we propose a preprocessing workflow applicable to all instances. In section 5 descriptive analytics for ITC_2002 and ITC_2007 datasets are described. In section 6 we provide a mathematical model and some model variations. A simulated annealing procedure and the neighborhood operators that it uses are also presented in the same section. Experimental results are given in section 7. Finally, we summarize and conclude in section 8.

2 PROBLEM DESCRIPTION

A well-known scheduling problem having both theoretical and practical significance is course scheduling. Curriculum-based course scheduling (CB-CTT) and post-enrollment course scheduling (PE-CTT) are two problem variations. While we have information about each student's enrollments in PE-CTT, in CB-CTT these enrollments are "hidden" behind courses. The goal of the PE-CTT variation, which is the subject of this paper, is to schedule events in the available timeslots and rooms. Each day has 9 timeslots, and a week consists of 5 days, so 45 timeslots are available for all instances.

Various problem instances for Course Timetabling exist, adhering to several assumptions about the problem. The International Timetabling Competition held in 2002 and its sequel in 2007 [6] 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. As usually constraints are divided into soft constraints and hard constraints, hard constraints are obligatory and soft ones can be violated but will impose a penalty upon the objective function. For these datasets the hard constraints are:

- Each event must be scheduled in a timeslot and a room.
- Events with common students must be placed at different timeslots.
- A room can host at most one event in each timeslot.
- The capacity and feature requirements of each event must be met by the room that will eventually host it.
- Certain timeslot requirements may apply to specific events.
- There may be relationships of precedence between events.

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 attends an event at the last timeslot of a day.
- A student attends three (or more) events in a row on the same day.
- A student attends only one event in a day.

One penalty point is imposed for each of the conditions above.

3 RELATED WORK

Educational timetabling problems have attracted interest from the academia since such problems stem from the needs of universities and other educational institutes. Implicit familiarity makes them easily understandable. Main educational timetabling problems are examination timetabling [9], course timetabling [8], high school timetabling [20] and their variants [5]. Moreover, several other timetabling problems arise in the educational context, like thesis defense timetabling [1], invigilator duty allocation [13], and others.

Recent surveys for educational timetabling have been published by Tan et al. [19] and by Ceschia et al. [4] that complement existing surveys on the field [18].

For the version of the PE-CTT problem that we study, many 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. [10] [11] and Nagata et al. [16]. Metaheuristics like Simulated Annealing are used by Ceschia et al. [3]. Cambazard et al. [2] presented a Constraint Programming approach.

Lewis et al. [15] explored the connectivity of the solution space in course timetabling problems under various neighborhood operators.

4 PREPROCESSING

The format of the problem instances assumes rooms with varying capacities alongside features that each room might have (e.g., video projector, smart-board, laboratory equipment, etc.). Additionally, it lists the events that each student participates and any additional obligations associated with those events. 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 tracking all the feasible combinations consisting of three events in order to reduce the model size.

4.1 Event-Room eligibility

Let $\mathbb{E}$ be the set of all events. Let S_e be the total number of students attending event e. Equation 1 determines when a student attends an event and the total number of attendees of each event is given by Equation 2. This information is extracted from the problem data.

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

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

4.2 Event Conflicts

Events with common students are prohibited from taking place at the same timeslot because, by definition, no student can attend more than one event at once. Such pairs of events are considered 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 increase the number of conflicting events by adding pair of events 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 and, therefore, can not be hosted in the same timeslot. Finally, the pairs of all conflicting events form set $\mathbb{G}$. Based on the event conflicts, we compute the conflict density of each problem which is twice the size of set $\mathbb{C}$ divided by the square of the size of set E. Conflict density can be considered a measure of each problem instance's difficulty, but room existence seems to distort its relevance.

4.3 Event Combinations

In the preprocessing stage, we compute all combinations of three events and store the number of students participating in each combination. 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 student s, all possible combinations consisting of three events e_1, e_2, e_3, where $e_1 \in \mathbb{E}_s, e_2 \in \mathbb{E}_s, e3 \in \mathbb{E}_s$ are generated. Finally, $\mathbb{C}$ is the set comprised of all previously generated three event combinations.

5 DATASET

Details about problem instances belonging to datasets ITC2002 and ITC2007 are presented in Table 1 and Table 2 respectively. The former table has three fewer columns than the latter since problem instances of ITC2002 have neither event-precedence relations nor event-timeslot restrictions.

Table 1. Dataset ITC2002
DS: Instance Name, E: Events, R: Rooms, F: Features, CD: Conflict Density, PA: Average Period UnAvailability, RC:Average Room Size(Capacity), RS: Average Room Suitability

DS	E	R	F	S	CD	RC	RS
o01.tim	400	10	10	200	0.20	10.40	1.96
o02.tim	400	10	10	200	0.21	10.40	1.92
o03.tim	400	10	10	200	0.23	10.80	3.42
o04.tim	400	10	5	300	0.23	15.30	2.45
o05.tim	350	10	10	300	0.31	17.30	1.78
o06.tim	350	10	5	300	0.26	17.90	3.59
o07.tim	350	10	5	350	0.21	20.60	2.87
o08.tim	400	10	5	250	0.17	12.80	2.93
o09.tim	440	11	6	220	0.17	10.36	2.58
o10.tim	400	10	5	200	0.20	10.70	3.49
o11.tim	400	10	6	220	0.20	11.40	2.06
o12.tim	400	10	5	200	0.20	10.30	1.96
o13.tim	400	10	6	250	0.21	12.60	2.43
o14.tim	350	10	5	350	0.25	20.30	3.08
o15.tim	350	10	10	300	0.25	17.40	2.19
o16.tim	440	11	6	220	0.18	10.73	3.17
o17.tim	350	10	10	300	0.31	17.20	1.11
o18.tim	400	10	10	200	0.21	10.50	1.75
o19.tim	400	10	5	300	0.20	15.30	3.94
o20.tim	350	10	5	300	0.25	17.50	3.43

6 FORMULATION

6.1 Mathematical Model

In this section we will present the base mathematical model of the problem.

Let $\mathbb{S}$ be the set of all students.

Let $\mathbb{E}$ be the set of all events.

Let $\mathbb{R}$ be the set of all rooms.

Let R_e be the set of rooms that can not host event e.

Let $\mathbb{T}$ be the set of all timeslots.

Let T_e be the set of timeslot that event e cannot be scheduled.

Let $\mathbb{L} = [9, 18, 27, 36, 45]$. These numbers refer to the last timeslot of each one of the 5 days.

Let $\mathbb{G}$ be the set of conflicting event pairs.

Table 2. Dataset ITC2007
DS: Instance Name, E: Events, R: Rooms, F: Features, CD: Conflict Density, PA: Average Period UnAvailability, RC:Average Room Size(Capacity), RS: Average Room Suitability

DS	E	R	F	S	CD	PA	RC	RS
i01.tim	400	10	10	500	0.34	0.44	37.70	4.08
i02.tim	400	10	10	500	0.37	0.43	36.10	3.95
i03.tim	200	20	10	1000	0.47	0.43	86.60	5.04
i04.tim	200	20	10	1000	0.52	0.43	89.15	6.40
i05.tim	400	20	20	300	0.31	0.43	21.55	6.80
i06.tim	400	20	20	300	0.30	0.44	21.80	5.07
i07.tim	200	20	20	500	0.53	0.60	42.00	1.57
i08.tim	200	20	20	500	0.51	0.62	44.50	1.92
i09.tim	400	10	20	500	0.34	0.44	37.90	2.91
i10.tim	400	10	20	500	0.38	0.43	36.30	3.20
i11.tim	200	10	10	1000	0.50	0.44	84.10	3.38
i12.tim	200	10	10	1000	0.58	0.43	84.10	3.35
i13.tim	400	20	10	300	0.32	0.43	22.10	8.68
i14.tim	400	20	10	300	0.32	0.43	22.25	7.56
i15.tim	200	10	20	500	0.53	0.61	44.00	2.23
i16.tim	200	10	20	500	0.45	0.61	43.90	1.74
i17.tim	100	10	10	500	0.70	0.43	138.20	2.77
i18.tim	200	10	10	500	0.65	0.43	70.40	3.48
i19.tim	300	10	10	1000	0.47	0.44	56.30	3.66
i20.tim	400	10	10	1000	0.28	0.44	44.80	3.73
i21.tim	500	20	20	300	0.23	0.42	17.40	7.36
i22.tim	600	20	20	500	0.26	0.43	24.85	5.65
i23.tim	400	20	30	1000	0.44	0.21	68.65	2.89
i24.tim	400	20	30	1000	0.31	0.44	42.65	1.59

Let $\mathbb{C}$ be the set consisting of combinations of three events with students in common.

Let C_{e_1,e_2,e_3} be the total number of students attending all three events e_1, e_2 and e_3.

Let $\mathbb{P}$ be the set of pairs of events having a precedence relation.

We define binary decision variables x_{etr} as seen in equation 3. Binary variables y_{sd} and $z_{e_1e_2e_3}$ in equations 4 and 5 are auxiliary variables.

$$x_{etr} = \begin{cases} 1 & \text{if event } e \text{ is scheduled in timeslot } t \text{ at room } r \\ 0 & \text{otherwise} \end{cases} \qquad \forall e \in \mathbb{E}, \forall t \in \mathbb{T}, \forall r \in \mathbb{R} \tag{3}$$

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

$$z_{e_1e_2e_3} = \begin{cases} 1 & \text{if events } e_1, e_2 \text{ and } e_3 \text{ are placed in 3 sequential timeslots in the same day} \\ 0 & \text{otherwise} \end{cases} \quad \forall (e_1, e_2, e_3) \in \mathbb{C} \tag{5}$$

$$\text{Minimize} \sum_{e\in\mathbb{E}} \sum_{t\in\mathbb{L}} \sum_{r\in\mathbb{R}} S_e * x_{etr} + \sum_{s\in\mathbb{S}} \sum_{d=1}^{5} y_{sd} + \sum_{(e_1,e_2,e_3)\in\mathbb{C}} C_{e_1,e_2,e_3} z_{e_1,e_2,e_3} \tag{6}$$

Subject to

$$\sum_{t\in\mathbb{T}_e} \sum_{r\in\mathbb{R}} x_{etr} = 0 \quad \forall e \in \mathbb{E} \tag{7}$$

$$\sum_{t\in\mathbb{T}} \sum_{r\in\mathbb{R}_e} x_{etr} = 0 \quad \forall e \in \mathbb{E} \tag{8}$$

$$\sum_{t\in\mathbb{T}} \sum_{r\in\mathbb{R}} x_{etr} = 1 \quad \forall e \in \mathbb{E} \tag{9}$$

$$\sum_{e\in\mathbb{E}, r\in\mathbb{R}} x_{etr} \leq 1 \quad \forall r \in \mathbb{R}, \quad \forall t \in \mathbb{T} \tag{10}$$

$$\sum_{r\in\mathbb{R}} x_{e_1tr} + \sum_{r\in\mathbb{R}} x_{e_2tr} \leq 1 \quad \forall(e_1, e_2) \in \mathbb{G}, \quad \forall t \in \mathbb{T} \tag{11}$$

$$\sum_{t\in\mathbb{T}} \sum_{r\in\mathbb{R}} t * x_{e_1tr} + 1 \leq \sum_{t\in\mathbb{T}} \sum_{r\in\mathbb{R}} t * x_{e_2tr} \quad \forall(e_1, e_2) \in \mathbb{P} \tag{12}$$

$$\sum_{t=t'}^{t'+2} \sum_{r\in\mathbb{R}} x_{e_1tr} + x_{e_2tr} + x_{e_3tr} \leq 2 + z_{e_1e_2e_3} \quad \forall(e_1, e_2, e_3) \in \mathbb{C}, \quad \forall t' \in T, \quad t' \notin [8, 9, 17, 18, 26, 27, 35, 36, 44, 45] \tag{13}$$

$$y_{sd} = 1, \quad \text{if} \sum_{e\in\mathbb{E}_s} \sum_{t=1+(d-1)*9}^{d*9} \sum_{r\in\mathbb{R}} x_{etr} = 1 \quad \forall s \in \mathbb{S}, \quad \forall d \in [1..5] \tag{14}$$

Equation 6 is the objective function that incorporates the costs associated with the three soft constraints. The first term imposes penalty S_e for any event e scheduled in the day's final time slot. The second term imposes a single penalty point for each student who participates in only one event during a day. The last term imposes a penalty equal to the total number of students attending a combination of three events if these events are in three adjacent timeslots on the same day.

Constraints 7, 8 handle timeslot and room availability limitations, while constraint 9 ensures that each event is scheduled once and only once. Constraint 10 ensures that at most one event is scheduled in each room in each timeslot. Conflicting events are banned from the same timeslot through the use of constraint 11, and precedence relations are respected as a consequence of constraint 12. To enforce a penalty for three consecutive events constraint 13 will activate z decision variables. Finally, since constraint 14 is nonlinear, it can be handled by CP solvers and MIP solvers equipped with automatic linearization capabilities.

6.1.1 Model Modifications. The base mathematical model may not be able to handle large instances, mainly due to the large size of $\mathbb{C}$. However, a decomposition of the problem that operates over 2 or 3 days can produce partial solutions that hopefully will drive the Simulated Annealing procedure described in section 6.3 to promising subdomains of the search space.

The three modifications to the base model follow.

- **Improve day by day** Optimizing each day in isolation results in some advantages. Since the number of events that are scheduled in one day is fewer than all events, the number of combinations of three events becomes significantly smaller. Moreover, the second term of the objective function is redundant now because the events of the day are determined. Finally, since the involved students participate in a number of events that cannot be changed, constraint 14 becomes redundant.

- **Improve days** In this modification of the base model, we consider two or three days. Again, as in the previous modification, the three event combinations are fewer than combinations involving all events. In this setting, students can be considered to attend a subset of the events they actually attend since some events have not been scheduled on the days we consider. The advantage is that now more, temporarily identical students can be identified and consolidated in the second term of the objective function and constraint 14.
- **Fix room** Starting from a given solution, we can enforce all events not to change rooms but allow them to change timeslots. Practically, we allow events placed in the same room at different timeslots to swap places.

6.2 Neighborhood operators

Three different neighborhood operators are used in this work:

- **Transfer Event**: An event $e \in \mathbb{E}$ is moved from its currently designated timeslot $t \in \mathbb{T}$, to a new, randomly selected timeslot $t_1 \in \mathbb{T}$. The move is executed only if t_1 is available for e, a compatible free room $r \in R$ exists in t_1 for event e and the precedence relations for event e are not violated .
- **Swap Events**: Two timeslots $t_1, t_2 \in \mathbb{T}$ that are designated to two events $e1, e2 \in \mathbb{E}$ are swapped. The move is executed only if $e1$ and $e2$ are in conflict, a suitable room $r_1 \in \mathbb{R}$ for $e1$ is available at t_2 and a suitable room $r_2 \in \mathbb{R}$ is available at t_1 and is all the precedence relations for e_1 and e_2 are not violated.
- **Kempe Chain**: An event e residing in timeslot $t_1 \in \mathbb{T}$ is selected randomly and moved to timeslot $t_2 \in \mathbb{T}$. All events in t_2 conflicting with e are moved to t_1. An ejection procedure follows until no conflicting events co-exist in t_1 or t_2. The move is executed if:
  - At each step for an event $e \in \mathbb{E}$ a compatible room exists in the selected timeslot.
  - All the precedence relations of e are not violated.
  - The selected period is available for the event e.

6.3 Simulated Annealing(SA)

Many versions of simulated annealing have been proposed in the literature [7]. The version we use is based on the classic one proposed by Kirkpatrick [14]. At each iteration a neighborhood operator 6.2 is randomly selected. The move is performed if the objective value is to be reduced. If $D_f > 0$, where D_f is the difference between the cost of the current iteration and the cost of the previous iteration, then the candidate solution has the potential of being accepted. The acceptance depends upon the probability defined in equation 15 where T is the current temperature value and T_s is the temperature value used for initiating the procedure. We employ a geometric cooling scheme $T' = a * T$, where $a \in [0.9, 0.999]$ is the cooling rate. Also, a freezing temperature $F_t = 1$ is used in the procedure. When T reaches F_t, two or three random days are selected, and an improve_days model is used in order to reduce the objective value. When the previous step ends, the temperature is set to a random value in the range $[0.5 * T_s, 1.5 * T_s]$. The procedure terminates when the time limit expires.

$$P = e^{-D_f/T} \tag{15}$$

7 EXPERIMENTS

Our experiment were programmed in Python, and we used Gurobi MIP solver [12] and Google OR-Tools CP-SAT solver [17]. The Gurobi MIP solver constructs the initial solution using the base model. Then, the solution improves by employing the day-by-day modification of the base model. The simulated annealing procedure follows as described in section 6.3. In each iteration of the simulated annealing procedure, a neighborhood operator is selected randomly either from the operators described in subsection 6.2, or one of the model modifications described in subsection 6.1.1. The implementations of the model modifications were programmed using CP-SAT solver. We provided a generous time duration for each execution (all steps) which amounted to approximately two hours for each problem instance. The experiments ran in a workstation equipped with an AMD Ryzen 5700G(8C/16T) processor and 32GB of RAM, running Windows 11. Results are presented in table 3 for itc2002 dataset and table 4 for itc2007 dataset.

Table 3. ITC_2002 results

Dataset	Best solution	Our solution
o01.tim	N/A	454
o02.tim	14	19
o03.tim	36	38
o04.tim	76	77
o05.tim	56	65
o06.tim	1	8
o07.tim	2	18
o08.tim	6	14
o09.tim	8	12
o10.tim	41	50
o11.tim	19	27
o12.tim	N/A	460
o13.tim	51	58
o14.tim	13	28
o15.tim	3	17
o16.tim	4	13
o17.tim	35	41
o18.tim	11	20
o19.tim	46	48
o20.tim	0	20

s

8 CONCLUSION

In this work, we presented a hybrid approach for the Post Enrollment Timetabling problem that returns results close to best known in several instances of a publicly available dataset. The contribution of our approach is the exact model we proposed and its modifications used for attaining those good solutions. The basic idea is that we try to reduce the complexity of the model by identifying groups of identical students. In effect, the model disengages from each student, becoming more event-centric. This alternate view of the problem gives the advantage of having manageable model sizes, especially when we solve subparts of the problem involving either a single or a few days. Given that the constraints added in ITC 2007 and room requirements kept the problem out of reach for modern solvers, we believe that our approach is a step forward in using exact solvers for approaching the Post Enrollment Timetabling problem.

Table 4. ITC_2007 results

Dataset	Best solution	Our solution
i01.tim	0	0
i02.tim	0	0
i03.tim	31	193
i04.tim	21	92
i05.tim	0	35
i06.tim	0	0
i07.tim	0	0
i08.tim	0	39
i09.tim	0	31
i10.tim	0	31
i11.tim	39	76
i12.tim	0	0
i13.tim	0	0
i14.tim	0	0
i15.tim	0	10
i16.tim	0	34
i17.tim	0	89
i18.tim	0	56
i19.tim	0	50
i20.tim	543	574
i21.tim	5	6
i22.tim	5	19
i23.tim	1292	1335
i24.tim	0	12

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] Michele Battistutta, Sara Ceschia, Fabio De Cesco, Luca Di Gaspero, and Andrea Schaerf. 2019. Modelling and solving the thesis defense timetabling problem. *Journal of the Operational Research Society* 70, 7 (2019), 1039–1050.

[2] Hadrien Cambazard, Emmanuel Hebrard, Barry O'Sullivan, and Alexandre Papadopoulos. 2012. Local search and constraint programming for the post enrolment-based course timetabling problem. *Annals of Operations Research* 194, 1 (2012), 111–135.

[3] Sara Ceschia, Luca Di Gaspero, and Andrea Schaerf. 2012. Design, engineering, and experimental analysis of a simulated annealing approach to the post-enrolment course timetabling problem. *Computers & Operations Research* 39, 7 (2012), 1615–1624.

[4] Sara Ceschia, Luca Di Gaspero, and Andrea Schaerf. 2022. Educational Timetabling: Problems, Benchmarks, and State-of-the-Art Results. *arXiv preprint arXiv:2201.07525* (2022).

[5] Marco Chiarandini, Luca Di Gaspero, Stefano Gualandi, and Andrea Schaerf. 2012. The balanced academic curriculum problem revisited. *Journal of Heuristics* 18, 1 (2012), 119–148.

[6] Luca Di Gaspero, Barry McCollum, and Andrea Schaerf. 2007. *The second international timetabling competition (ITC-2007): Curriculum-based course timetabling (track 3).* Technical Report. Technical Report QUB/IEEE/Tech/ITC2007/CurriculumCTT/v1. 0, Queen's ….

[7] Kathryn Anne Dowsland and Jonathan Thompson. 2012. Simulated annealing. *Handbook of natural computing* (2012), 1623–1655.

[8] Luca Di Gaspero and Andrea Schaerf. 2002. Multi-neighbourhood local search with application to course timetabling. In *International Conference on the Practice and Theory of Automated Timetabling*. Springer, 262–275.

[9] Christos Gogos, Panayiotis Alefragis, and Efthymios Housos. 2012. An improved multi-staged algorithmic process for the solution of the examination timetabling problem. *Annals of Operations Research* 194, 1 (2012), 203–221.

[10] Say Leng Goh, Graham Kendall, and Nasser R Sabar. 2019. Simulated annealing with improved reheating and learning for the post enrolment course timetabling problem. *Journal of the Operational Research Society* 70, 6 (2019), 873–888.

[11] Say Leng Goh, Graham Kendall, Nasser R Sabar, and Salwani Abdullah. 2020. An effective hybrid local search approach for the post enrolment course timetabling problem. *Opsearch* 57, 4 (2020), 1131–1163.

[12] Gurobi Optimization, LLC. 2022. Gurobi Optimizer Reference Manual. https://www.gurobi.com

[13] Mohmad Kahar and Graham Kendall. 2014. Universiti Malaysia Pahang examination timetabling problem: scheduling invigilators. *Journal of the Operational Research Society* 65, 2 (2014), 214–226.

[14] Scott Kirkpatrick, C. Gelatt, and M. Vecchi. 1983. Optimization by Simulated Annealing. *Science (New York, N.Y.)* 220 (06 1983), 671–80. https://doi.org/10.1126/science.220.4598.671

[15] Rhydian Lewis and J Thompson. 2015. Analysing the effects of solution space connectivity with an effective metaheuristic for the course timetabling problem. *European Journal of Operational Research* 240, 3 (2015), 637–648.

[16] Yuichi Nagata. 2018. Random partial neighborhood search for the post-enrollment course timetabling problem. *Computers & Operations Research* 90 (2018), 84–96.

[17] Laurent Perron and Vincent Furnon. [n. d.]. OR-Tools. ([n. d.]). https://developers.google.com/optimization/

[18] Rong Qu, Edmund K Burke, Barry McCollum, Liam TG Merlot, and Sau Y Lee. 2009. A survey of search methodologies and automated system development for examination timetabling. *Journal of scheduling* 12, 1 (2009), 55–89.

[19] Joo Siang Tan, Say Leng Goh, Graham Kendall, and Nasser R Sabar. 2021. A survey of the state-of-the-art of optimisation methodologies in school timetabling problems. *Expert Systems with Applications* 165 (2021), 113943.

[20] Christos Valouxis, Christos Gogos, Panayiotis Alefragis, and Efthymios Housos. 2012. Decomposing the high school timetable problem. *Practice and Theory of Automated Timetabling (PATAT 2012), Son, Norway* 61 (2012).