EDUCATIONAL TIMETABLING: PROBLEMS, BENCHMARKS, AND STATE-OF-THE-ART RESULTS

A PREPRINT

Sara Ceschia, Luca Di Gaspero, Andrea Schaerf*
DPIA, University of Udine, Via delle Scienze 206, 33100 Udine, Italy
email: {sara.ceschia,luca.digaspero,andrea.schaerf}@uniud.it

January 20, 2022

ABSTRACT

We propose a survey of the research contributions on the field of Educational Timetabling with a specific focus on "standard" formulations and the corresponding benchmark instances. We identify six of such formulations and we discuss their features, pointing out their relevance and usability. Other available formulations and datasets are also reviewed and briefly discussed. Subsequently, we report the main state-of-the-art results on the selected benchmarks, in terms of solution quality (upper and lower bounds), search techniques, running times, statistical distributions, and other side settings.

Keywords Timetabling · Validation · Benchmarks · Reproducibility

1 Introduction

Educational Timetabling, in essence, consists in assigning teacher/student meetings to days, timeslots, and classrooms. Despite this apparent simplicity, experience teaches us that every single institution has its own rules, conventions, and fixations, thus making each specific problem almost unique. As a consequence, uncountably many different problem formulations have been proposed in the literature on Educational Timetabling, depending on the type of institution (high-school, university, or other), the type of meetings (lectures, exams, ...), and the different settings, constraints, and objectives.

Many papers in the literature tackle a specific problem using a selected search method. The authors normally claim the success of the application, though rarely dispelling the doubt over the readers that the method used was more the authors' "favorite" rather than the most suitable for the problem under consideration. A few previous surveys have tried to put in order this situation by creating a taxonomy of both problem formulations and corresponding search methods used for their solution, in order to draw some conclusions about what works best in each specific case (see Section 2).

In this survey, we want to take a somewhat different point of view. Specifically, we focus on the review of the problem formulations and their publicly available datasets, critically discussing their practical relevance and usability. To this aim, we highlight which datasets have been considered most frequently in the literature, so that they have risen to the status of *benchmarks*, and the corresponding formulation to the status of a *de facto* standard.

We identified six standard formulations, which are presented in chronological order in Sections 3.1—3.6. Incidentally, the chronological order corresponds also to the order of increasing complexity and adherence to the real-world situation. Indeed, we can see that the research has moved continuously from very simplified problems toward full-fledged ones. Nonetheless, in our opinion the early simplified formulations are still interesting testbeds for new search methods, and they have not yet finished to serve their purpose. On the contrary, the accumulated bulk of results and techniques make them even more interesting and challenging.

*Corresponding author

For these formulations and benchmarks, we discuss state-of-the-art results, in terms of solution quality, search techniques, running times, and other side settings. We will also discuss the availability of upper and lower bounds, in order to identify which are the most challenging instances for future comparisons.

We also review and discuss other formulations that have not attracted general interest so far, but still provide real-world publicly available datasets and could be potentially interesting for the community.

Finally, we consider the issue of reliability of the results claimed in the literature, stressing the importance of the presence of instance and solution checkers, so as to provide against possible errors and misunderstandings. To this aim, we developed a web application, named OPTHUB (`https://opthub.uniud.it`), that allows users to check and upload both new instances and solutions. All data, properly validated and timestamped, is available for download and inspection, along with scoreboards and statistics. The system is meant to provide a unified and up-to-date site for current contributions, so as to facilitate and encourage further research and future comparisons. OPTHUB, whose development is still ongoing, currently hosts four of the formulations discussed in this survey. The formulations hosted are the early ones that do not have a dedicated and updated online repository on their own.

In a way, this survey is meant for researchers interested in writing what Johnson [2002] called a *horse race paper*, in which the authors assess the quality of their methods by the comparison to previous research on the designated benchmarks. We aim to help such perspective researchers to be rigorous, fair, and comprehensive as much as possible in such a complex task of comparing with the whole literature.

However, our hope is that this effort could be useful also for the authors of an *application paper* (still following Johnson's terminology), that aims at solving one specific original problem. Indeed, those authors could evaluate the quality of their search method by identifying an underlying standard problem that could be a simplified version of their own specific one, adapt their search method to solve it, and report the corresponding results. Naturally, it is not expected that a solver for a complex, full-fledged problem could outperform specialized ones for the benchmarks, but this would give a reasonable measure of the quality of the proposed approach.

This survey is organized as follows. In Section 2, we list the various problems within the scope of the Educational Timetabling area. In Section 3, we introduce and discuss the available formulations and datasets for these problems. In Section 4, we illustrate and comment the state-of-the-art results for the benchmarks. Finally, conclusions and future directions are discussed in Section 5.

2 Educational Timetabling

In this section, we introduce the educational timetabling problems and discuss various general issues of the research area.

2.1 Educational timetabling problems

According to the literature on timetabling [see, e.g., Schaerf, 1999, Kingston, 2013], there are three main problems in the educational timetabling area:

High-School Timetabling (HTT) The weekly scheduling for all the classes of a high-school, avoiding teachers meeting two classes at the same time, and vice versa.

University Course Timetabling (CTT) The weekly scheduling for all the lectures of a set of university courses, minimizing the overlaps of lectures of courses having common students.

University Examination Timetabling (ETT) The scheduling for the exams of a set of university courses, avoiding overlap of exams of courses having common students, and spreading the exams for the students as much as possible.

Even though a clear cut between HTT, CTT, and ETT is not possible (e.g., some high-schools are organized in a university fashion), they normally differ from each other significantly, and most of the papers in the literature can be classified within one of these three problems.

2.2 Previous surveys

Many surveys on educational timetabling have recently appeared in the literature. However, due to the vastness of the research area, all of them focus an a subset of the problems introduced in the previous section, in order to reduce their scope. For example, Burke and Petrovic [2002] and MirHassani and Habibi [2013] focus on university timetabling (CTT and ETT), likewise Lewis [2008] who further limits his study to metaheuristic techniques. Similarly, the survey

by Qu et al. [2009] is dedicated to ETT, whereas the one by Pillay [2014] is only on HTT, and the recent ones by Chen et al. [2021] and Tan et al. [2021] are on CTT and HTT, respectively. The survey by Bettinelli et al. [2015] reviews only one specific formulation of CTT, namely the curriculum-based course timetabling (CB-CTT), that will be introduced and discussed in Section 3.3.

2.3 Other timetabling problems

There are also other problems within the Educational Timetabling field that have been addressed in the literature, although they are less popular than the previous three. Among these "minor" problems we can include *Student Sectioning* [Müller and Murray, 2010], *Thesis Defense Timetabling* [Battistutta et al., 2019], *Trainee/Intern/Resident Assignment* (for medical and military schools) [Akbarzadeh and Maenhout, 2021], and *Conference Scheduling* [Stidsen et al., 2018]. We do not discuss the above problems in details, as there are no available datasets that have reached the status of benchmarks. An exception is Student Sectioning that is included together with CTT in the ITC-2019 formulation, that will be discussed in Section 3.6.

Other timetabling problems, which fall outside the scope of Educational Timetabling, such as Employee Timetabling [Meisels and Schaerf, 2003], Transportation (trains and airplanes) Timetabling [Cacchiani and Toth, 2012], and Sport Timetabling [Van Bulck et al., 2020] are not discussed here.

2.4 Timetabling initiatives

The timetabling community is quite active. There are a biannual conference series (`http://patatconference.org`) and a EURO Working Group (`https://www.euro-online.org/web/ewg/14/`), both called PATAT (Practice and Theory of Automated Timetabling) and dedicated to the whole area of Timetabling problems. One of their activities has been the organization of five International Timetabling Competitions: ITC-2002, ITC-2007 [McCollum et al., 2010], ITC-2011 [Post et al., 2016], ITC-2019 [Müller et al., 2018], and ITC-2021 [Van Bulck et al., 2021]. These competitions have brought forth most of the standard formulations and benchmarks discussed in Section 3. Incidentally, the most recent one, ITC-2021, did not focus on educational timetabling like the previous ones but on sports timetabling.

2.5 Multiobjective formulations

For all the standard formulations that we will introduce in Section 3, there is a single objective function, defined as a weighted sum of the various penalty terms to be minimized. Therefore, we do not include in this survey the issues related to multiobjective optimization [Silva et al., 2004], although the multiobjective perspective would be surely useful in this context, as objectives in timetabling could be rather intangible and thus not always commensurable. Indeed, many objectives are related to the comfort of the participants (students or teachers), so that it is difficult to assign to them a specific numeric weight. Furthermore, besides the classical objectives measuring the general comfort, some authors include also other notions, which are even more difficult to be put in the same scale of the other objectives. These include the *fairness* [Mühlenthaler and Wanka, 2016], that takes care for the balanced distribution of the discomfort among the participants (teachers and students) and the *robustness* [Akkan and Gülcü, 2018], that measures the possibility to do not deteriorate the quality in presence of unforeseen disruptions.

2.6 Terminology and taxonomy

We define here some common terms in the timetabling vocabulary that will be used throughout this survey. Concepts that are specific of one formulation are introduced in the dedicated section.

Times: The time *horizon* is divided into *days* and each day is split into *timeslots* (in general, the same number of timeslots is given in each day). A *period* is a pair $\langle \text{day}, \text{timeslot} \rangle$.

Events: An *event* is a meeting between students and one or more teachers. Events can be of different types: *lectures* or *exams* of a *course*, *laboratories*, or *seminars*.

Resources: We consider three main kinds of resources: students, teachers, and rooms. Events have to be scheduled taking into account resource restrictions, such as *students*' enrollments, *teachers*' requests and *rooms*' availabilities.

Constraints: As customary, constraints are split into *hard* and *soft* ones (soft constraints are also called objectives). The hard constraints must be always satisfied, whereas the soft ones contribute to the objective function, which is a weighted sum of all soft constraint penalties.

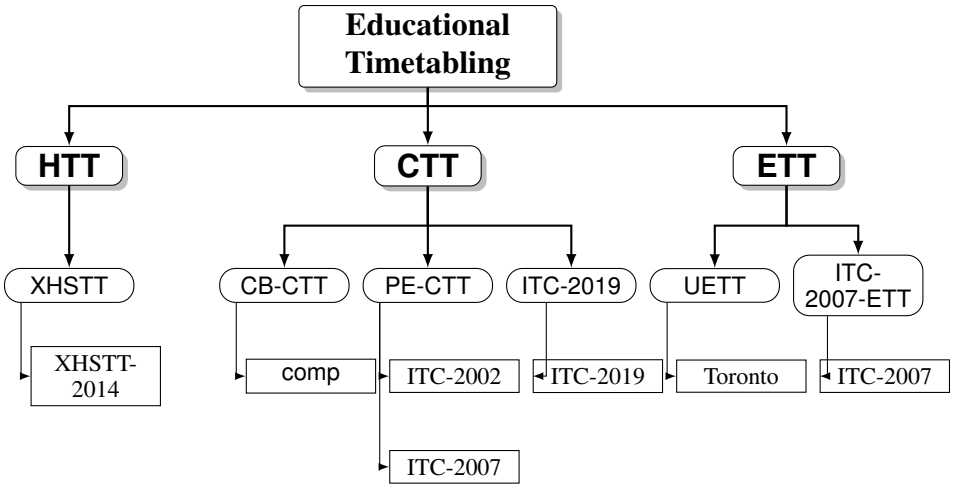

Figure 1: Educational timetabling problems, formulations, and benchmarks.

Figure 1 shows the taxonomy of the problems presented in Section 2.1, and the corresponding formulations introduced one by one in Section 3. For each formulation, the figure reports the datasets used as benchmarks.

3 Formulations and Datasets

We introduce the selected formulations and the corresponding datasets in Sections 3.1—3.6. For each of these formulations, we present in turn (i) a brief specification, (ii) the benchmarks with their main features, (iii) the file formats and their usability, (iv) the presence of additional datasets and instance generators, and (v) some discussion, including the assessment of the gap w.r.t. the complete real-world problem.

From the benchmarks, we identify and remove the instances that are too easy to be kept in the pool, and their presence results only on a waste of computational time. We name an instance as easy when all runs of the top search techniques always find the same score, which is likely to be the optimal one.

Finally, Section 3.7 is devoted to list and discuss the other real-world formulations that provide available (and usable) public datasets.

3.1 Uncapacitated Examination Timetabling (UETT)

The first formulation that we consider is the classical version of ETT proposed by Carter et al. [1996], that we name UETT (U for uncapacitated, as explained below). This is a very essential view of the examination timetabling problem, which extends just slightly the underlying *graph coloring* problem, with exams as nodes and periods as colors.

Short specification The main input data of UETT is the Boolean-valued *enrollment* matrix, that stores for each pair $\langle \text{student}, \text{exam} \rangle$ the information about whether the student has to take the exam or not.

Two exams with at least one student in common are in conflict, so that they cannot to be scheduled in the same period. Conflicts are the sole constraints. In particular, rooms are not taken into account, and for this reason the problem is known as *uncapacitated*.

The objective function is related to the distance between exams with students in common. Distances are penalized in the following fixed way: the cost of scheduling two exams with k students in common at distance of 1, 2, 3, 4, and 5 periods is $16k$, $8k$, $4k$, $2k$, and k, respectively.

Benchmarks The UETT formulation comes with a dataset of 13 real-world instances mainly from North American universities, known as Toronto instances (or Carter's instances), whose main features are illustrated in Table 1 [taken from Bellio et al., 2021].

As remarked by Alefragis et al. [2021], the instances have some unnecessary data that could be removed by preprocessing. Indeed, some students are enrolled in only one exam, so that they can affect neither the constraints nor the objective. These students are called *noise students* by Alefragis et al. In turn, exams taken only by noise students do

not contribute to the constraints and the objective, and they are called *noise exams*. In Table 1, we report both the total number of students and exams and the *active* (non-noise) ones. The two columns **W** (Workload) represent the number of exams per active student (average and maximum value). The rightmost column **Gd** is the density of the conflict graph, which is computed as the number of conflicts divided by $n \cdot (n-1)/2$, where n is the number of active exams.

Inst.	**E**		**S**		**P**	**W**		**Gd**
	total	active	total	active		avg	max	
car91	682	678	16925	13516	35	4.20	9	0.13
car92	543	542	18419	14450	32	3.84	7	0.14
ear83	190	190	1125	1124	24	7.21	10	0.27
hec92	81	81	2823	2502	18	4.25	7	0.42
kfu93	461	444	5349	5073	20	4.92	8	0.06
lse91	381	379	2726	2627	18	4.16	8	0.06
pur93	2627	2413	30032	27405	42	4.40	9	0.03
rye93	486	485	11483	9458	23	4.76	10	0.08
sta83	139	139	611	611	13	9.41	11	0.14
tre92	261	260	4360	3693	23	4.03	6	0.18
uta92	622	622	21266	15086	35	3.91	7	0.13
ute92	184	184	2750	2672	10	4.41	6	0.08
yor83	181	181	941	940	21	6.42	14	0.29

Table 1: Features of the Toronto benchmark instances. Symbol definition: **E** (Exams), **S** (Students), **P** (Periods), **W** (Workload), **Gd** (Graph density).

File formats and repositories Instances are available in plain text and split in two separate files: one containing the exams and one with the student enrollments. The files were originally posted via FTP in the website of the University of Toronto (not active anymore), and are now available at `http://www.cs.nott.ac.uk/~pszrq/data.htm`. The same instances are posted on OPTHUB with a slightly modified (more robust), single-file format.

The original data format is unfortunately very fragile, as for example the accidental insertion of a newline character would result in a different (but still valid) instance. This has actually happened as discussed below.

Other datasets and generators Other instances of UETT are available. First, there is a set of 9 instances, called *apocryphal* by Bellio et al. [2021], that are variants of some of Toronto ones that were created by accidental perturbation of the original files and used unwittingly in a few experimental analyses [see Qu et al., 2009, for a discussion about them]. Even though they have been considered by a few authors, given that they are just arbitrary perturbations of the real instances, we do not consider them as benchmarks.

Another set of 20 instances, obtained by translating real-world instances for other examination timetabling formulations, have been made available by Bellio et al. on OPTHUB.

Finally, Bellio et al. [2021] developed a parametric generator that creates artificial instances with the prescribed values of the main features. A set of 100 generated instances, selected based on feasibility and computational hardness, are also available on OPTHUB.

Discussion UETT is surely a simplified formulation, as the authors themselves admit that "all side constraints have been removed" [Carter et al., 1996]. Indeed, they list in their original work a set of constraints that apply to some of the real-world cases, but have been neglected in the proposed formulation, in order to have a common ground for many different cases.

Despite its extreme simplicity, or perhaps actually due to it, UETT has been and still is an active subject of studies (see Section 4.1). The main reason could also be that the benchmarks proposed are very challenging. In fact, to the best of our knowledge, none of such instances has been solved to proven optimality so far.

3.2 Post-Enrolment Course Timetabling (PE-CTT)

The second formulation that we consider is the so-called Post-Enrolment Course Timetabling (PE-CTT) problem that is the first standard formulation of CTT. It has been proposed within the Metaheuristics Network project (2000-04), then used as the subject of ITC-2002, and used again for ITC-2007 with a slightly more complex formulation, which is the one discussed here. The full specification can be found in the work by Lewis et al. [2007, §3].

Short specification In PE-CTT it is given a set of events, a set of periods, and a set of rooms. It is also defined a set of days, such that each period is a timeslot belonging to one day. Students enroll in events causing conflicts between them.

Furthermore, there is a set of room features that may be required by events. Room features and capacity (in terms of seats) together result in a compatibility relation between rooms and events.

In addition, it is defined a precedence relation between events, such that some events must be scheduled before others. Finally, the last constraints are the ones originated from an unavailability relation, stating that an event cannot be scheduled in some specified periods.

The objective function is composed by three components that penalize the following cases: (i) a student attending an event in the last timeslot of a day, (ii) a student attending three (or more) events in successive timeslots in the same day, (iii) a student attending only one event in a day.

Benchmarks There are two datasets that can be considered as consolidated benchmarks for PE-CTT, which are the ones coming from the competitions ITC-2002 and ITC-2007. The dataset from ITC-2002 is on a simplified version of the problem that does not consider precedences and unavailabilities.

Inst.	**E**	**R**	**S**	**Ro**	**Cgd**	**SE**	**ES**	**RE**
01	400	10	200	0.89	0.20	8.88	17.75	1.96
02	400	10	200	0.89	0.21	8.61	17.23	1.92
03	400	10	200	0.89	0.23	8.85	17.70	3.42
04	400	10	300	0.89	0.23	13.07	17.43	2.45
05	350	10	300	0.78	0.31	15.24	17.78	1.78
06	350	10	300	0.78	0.26	15.23	17.77	3.59
07	350	10	350	0.78	0.21	17.48	17.48	2.87
08	400	10	250	0.89	0.17	10.99	17.58	2.93
09	440	11	220	0.89	0.17	8.68	17.36	2.58
10	400	10	200	0.89	0.20	8.89	17.78	3.49
11	400	10	220	0.89	0.20	9.58	17.41	2.07
12	400	10	200	0.89	0.20	8.79	17.58	1.96
13	400	10	250	0.89	0.21	11.06	17.69	2.43
14	350	10	350	0.78	0.25	17.42	17.42	3.08
15	350	10	300	0.78	0.25	15.07	17.58	2.19
16	440	11	220	0.89	0.18	8.88	17.75	3.17
17	350	10	300	0.78	0.31	15.15	17.67	1.11
18	400	10	200	0.89	0.21	8.78	17.56	1.75
19	400	10	300	0.89	0.20	13.28	17.71	3.94
20	350	10	300	0.78	0.25	14.99	17.49	3.43

Table 2: Features of the ITC-2002 benchmark instances. Symbol definition: **E** (Events), **R** (Rooms), **S** (Students), **Ro** (Room occupancy), **Cgd** (Conflict graph density), **SE** (average Students per Exam), **ES** (average Exam per Student), **RE** (average suitable Rooms per Event).

The main features of the instances are illustrated in Tables 2 and 3. The number of periods is 45 for all instances, thus it is not listed in the tables. All instances are artificial and obtained by a generator. In addition, they have the peculiarity of having been generated in such a way that at least one *perfect* (zero cost) solution exists.

For ITC-2007 instances it is generally more difficult than ITC-2002 ones to find a feasible solution. Indeed, looking at the tables, we see that they have a higher density (column **Cgd**), in addition to unavailabilities and precedences that are absent in the ITC-2002 version.

File formats and repositories All instances are available in a lengthy text-only format, in which all elements of the matrices are written explicitly, one per line. As a consequence, the files are easy to parse, but rather verbose, not human readable, and fragile. They are available at the websites of the competitions, reachable from the PATAT conference website (`http://patatconference.org/`). Instances are available also from OPTHUB.

Other datasets and generators Two other datasets are publicly available, and they have been considered in some papers, though less frequently than the two mentioned above. The first one is a dataset proposed by the Metaheuristics Network [see Rossi-Doria et al., 2003] before the competitions, using the simplified formulation of ITC-2002, which are available at `http://iridia.ulb.ac.be/supp/IridiaSupp2002-001`. For recent results on these instances see for example [Goh et al., 2017].

Inst.	E	R	S	Ro	Cgd	SE	ES	RE	TE	P
01	400	10	500	0.89	0.34	26.27	21.02	4.08	0.56	0.10
02	400	10	500	0.89	0.37	26.29	21.03	3.95	0.57	0.09
03	200	20	1000	0.22	0.47	66.92	13.38	5.05	0.56	0.10
04	200	20	1000	0.22	0.52	66.98	13.40	6.40	0.57	0.10
05	400	20	300	0.44	0.31	15.69	20.92	6.80	0.56	0.37
06	400	20	300	0.44	0.30	15.55	20.73	5.07	0.56	0.35
07	200	20	500	0.22	0.53	33.67	13.47	1.58	0.39	0.10
08	200	20	500	0.22	0.52	34.58	13.83	1.91	0.38	0.10
09	400	10	500	0.89	0.34	26.78	21.43	2.91	0.56	0.11
10	400	10	500	0.89	0.38	26.23	20.98	3.20	0.56	0.10
11	200	10	1000	0.44	0.50	68.04	13.61	3.38	0.56	0.10
12	200	10	1000	0.44	0.58	68.04	13.61	3.36	0.57	0.10
13	400	20	300	0.44	0.32	15.90	21.19	8.68	0.56	0.34
14	400	20	300	0.44	0.32	15.64	20.86	7.56	0.56	0.36
15	200	10	500	0.44	0.54	32.63	13.05	2.23	0.38	0.10
16	200	10	500	0.44	0.46	34.10	13.64	1.74	0.39	0.11
17	100	10	500	0.22	0.71	97.67	19.53	2.77	0.57	0.12
18	200	10	500	0.44	0.65	51.42	20.57	3.47	0.57	0.10
19	300	10	1000	0.67	0.47	44.78	13.44	3.66	0.56	0.10
20	400	10	1000	0.89	0.28	33.92	13.57	3.73	0.56	0.10
21	500	20	300	0.56	0.23	12.40	20.67	7.36	0.57	0.36
22	600	20	500	0.67	0.26	17.42	20.90	5.65	0.56	0.39
23	400	20	1000	0.44	0.44	53.42	21.37	2.89	0.78	0.12
24	400	20	1000	0.44	0.31	33.34	13.34	1.59	0.55	0.72

Table 3: Features of ITC-2007 benchmark instances. Symbol definition: **E** (Events), **R** (Rooms), **S** (Students), **Ro** (Room occupancy), **Cgd** (Conflict graph density), **SE** (average Students per Exam), **ES** (average Exam per Student), **RE** (average suitable Rooms per Event), **TE** (average availability of Timeslots for Event), **P** (average ratio of Precedences per Event).

The second dataset, including much larger instances, has been introduced by Lewis and Paechter [2007] with the aim of having more difficult cases and is available at `http://www.rhydlewis.eu/hardTT/`. Indeed, for these instances, feasibility is quite difficult to be obtained, and the comparison is on the number of violations rather than on the objective function. For results on these instances see for example [Ceschia et al., 2012].

The generator used for the instances of ITC-2002 and ITC-2007 was never made public, and no other one has been developed and made available for this formulation.

Discussion Like UETT discussed in the previous section, PE-CTT is rather a simplified formulation with respect to the real-world problem, as many aspects are deliberately removed in order to make the problem more manageable [see Lewis et al., 2007, §5 for a discussion about the neglected features].

In addition, as shown in Tables 2 and 3, the diversity of the features of the instances is quite limited. For example, all instances have exactly 45 periods divided in 5 days of 9 timeslots each, with no variability at all. Similarly, the number of rooms is restricted to just three different values, namely 10, 11, and 20.

3.3 Curriculum-Based Course Timetabling (CB-CTT)

The next formulation that we discuss is the so-called Curriculum-based Course Timetabling (CB-CTT), which has been proposed by Di Gaspero and Schaerf [2003], and subsequently adopted, in a slightly modified version, as the third track of ITC-2007.

The name of the problem comes from the fact that conflicts are determined by predefined curricula, opposed to the use of explicit student enrolments as in PE-CTT. This is however not the most important difference between PE-CTT and CB-CTT as the notions of student and curriculum are formally interchangeable, because a student can be expressed as a curriculum and vice versa. On the contrary, the main difference stems from the notion of *course* as a set of lectures that is absent in PE-CTT. Many constraints and objectives in CB-CTT are defined at the level of a course, whereas in PE-CTT constraints and objectives are always expressed at the level of the single event/lecture.

A few variants of this formulation have been subsequently proposed by Bonutti et al. [2012]. The most studied one however remains the one used for ITC-2007 [named UD2 in Bonutti et al., 2012], which is thus the one that we consider here. The full description is provided by Di Gaspero et al. [2007].

Inst.	C	L	R	PpD	D	Cu	MML	Co	TA	CL	RO
comp01	30	160	6	6	5	14	2-5	13.2	93.1	3.24	88.9
comp02	82	283	16	5	5	70	2-4	7.97	76.9	2.62	70.8
comp03	72	251	16	5	5	68	2-4	8.17	78.4	2.36	62.8
comp04	79	286	18	5	5	57	2-4	5.42	81.9	2.05	63.6
comp05	54	152	9	6	6	139	2-4	21.7	59.6	1.8	46.9
comp06	108	361	18	5	5	70	2-4	5.24	78.3	2.42	80.2
comp07	131	434	20	5	5	77	2-4	4.48	80.8	2.51	86.8
comp08	86	324	18	5	5	61	2-4	4.52	81.7	2	72
comp09	76	279	18	5	5	75	2-4	6.64	81	2.11	62
comp10	115	370	18	5	5	67	2-4	5.3	77.4	2.54	82.2
comp12	88	218	11	6	6	150	2-4	13.9	57	1.74	55.1
comp13	82	308	19	5	5	66	2-3	5.16	79.6	2.01	64.8
comp14	85	275	17	5	5	60	2-4	6.87	75	2.34	64.7
comp16	108	366	20	5	5	71	2-4	5.12	81.5	2.39	73.2
comp17	99	339	17	5	5	70	2-4	5.49	79.2	2.33	79.8
comp18	47	138	9	6	6	52	2-3	13.3	64.6	1.53	42.6
comp19	74	277	16	5	5	66	2-4	7.45	76.4	2.42	69.2
comp20	121	390	19	5	5	78	2-4	5.06	78.7	2.5	82.1
comp21	94	327	18	5	5	78	2-4	6.09	82.4	2.25	72.7

Table 4: Features of comp benchmark instances. Symbol definition: **C** (Courses), **L** (total Lectures), **R** (Rooms), **PpD** (Periods per Day), **D** (Days), **Cu** (Curricula), **MML** (Min and Max Lectures per day per curriculum), **Co** (average number of Conflicts), **TA** (average Teacher Availability), **CL** (average number of Lectures per Curriculum per day), **RO** (average Room Occupation).

Short specification As mentioned above, the key notions of CB-CTT are courses and curricula. Each course consists of a fixed number of *lectures* to be scheduled in different periods. A course is attended by a number of *students*, and is taught by a *teacher*. For each course, there are a minimum number of days over which the lectures of the course should be spread. Moreover, there are some unavailable periods in which the course cannot be scheduled.

Like in PE-CTT, we are given a number of periods divided in days and timeslots in the day. Each *room* has a *capacity*, specified as the number of available seats, but no other features.

A *curriculum* is a group of courses that potentially have students in common. As a consequence, lectures of courses belonging to the same curriculum are in conflict and cannot be scheduled in the same period. Two courses are in conflict also if they are taught by the same teacher.

The hard constraints regard conflicts, teacher availability and room occupancy. The objective function (soft constraints) is composed by four components that penalize the following cases: (i) the capacity of the room assigned to a lecture is less than the number of students attending the course, (ii) the lectures of a course are not spread into the given minimum number of days, (iii) a lecture is isolated, i.e., not adjacent to any other in the same curriculum, (iv) the lectures of a course are not given all in the same room.

Benchmarks Quite a few real-world datasets are available for the formulation. By far the most used one is the one from ITC-2007, known as the `comp` dataset that we consider as benchmark.

Table 4 [taken from Bonutti et al., 2012] shows the main features of these instances. It could be noticed that we removed instances comp11 and comp15. They have been remove for different reasons: comp11 is an easy one, as all competitive search methods always find a solution of cost zero, while `comp15` is actually identical to `comp03` in the problem variant UD2 that we consider here.

File formats and repositories Instances are available in an ad-hoc text-only format, which is reasonably human-readable. There are actually two versions of the format, the original `.ctt` one used for the competition, and the newer `.ectt` (`e` for extended) proposed by Bonutti et al. [2012] that includes additional data necessary for the other versions of the problem.

Instances are available on OPTHUB in `.ectt` format along with several results from the literature. The solutions were imported from the original CB-CTT website (`satt.diegm.uniud.it/ctt`) not available anymore.

Other datasets and generators A few other datasets of real-world instances coming mainly from Italian universities are available on OPTHUB.

An instance generator has been developed by Burke et al. [2008], which has been subsequently revised by Lopes and Smith-Miles [2010, 2013] in such a way to obtain more realistic instance, in particular more similar to the comp dataset. The latter generator has been further refined by De Coster et al. [2021] in order to enlarge the region of the instance space covered by the generated instances. The generator by De Coster et al. is available at https://cdlab-artis.dbai.tuwien.ac.at/papers/cb-ctt/.

Discussion Like UETT and PE-CTT, the CB-CTT formulation is a judicious simplification of the original problem. Constraints and objectives included in the formulation have been selected among the long list of real ones to be general and simple enough, but also representative of the various types of restrictions. For example, the objective on room stability for the lectures of a course is not particularly important in practice, but it represents a set of limitations that involve the use of rooms in different periods. Without this objective the management of the rooms could have been done independently for each period, which would have resulted in an oversimplification of the problem.

The comp instances are extracted from various departments of University of Udine (Italy), so that the values of the main features are quite diverse. The additional instances come also from different universities, so that they are yet broader in size and structure.

3.4 Examination Timetabling (ITC-2007-ETT)

The next formulation that we include in our study is the ETT proposed for ITC-2007 (Track 1). This formulation, even though it does not consider all practical features, is much more realistic than the uncapacitated version discussed in Section 3.1. Indeed, it includes several novel features collected from the activity of a commercial software in use in many British universities. The full specification is provided by McCollum et al. [2007].

Short specification Like other formulations, the time horizon is divided in a number of periods, each one belonging to a day. The novelty is that periods have a specific length (in minutes) and can have a penalty for scheduling exams in it.

As usual, rooms have a capacity and might be undesired, in the sense that there is a penalty for their use, like periods.

For each exam, a length of execution is given, so that it is compatible only with periods of sufficient duration. In addition, an exam might require to be scheduled in a dedicated room, otherwise it can share the room with other exams. For each exam, it is also given the set of students enrolled for it.

For some pairs of exams a precedence rule is specified, stating that one exam must be scheduled after, at the same time, or at a different time with respect to the other one.

The objective function is composed by the following components (soft constraints): (i) a student taking two exams in consecutive periods in the same day, (ii) a student taking two exams in the same day (iii) a student taking two exams within a fixed number of periods (spread), (iv) exams in the same room with mixed durations, (v) an exam with many students scheduled towards the end of the planning horizon, (vi) an exam scheduled in an undesired period or an undesired room.

The weights of the soft constraints vary from case to case, and are included in the input file of each instance.

Benchmarks A dataset composed of 12 instances from British universities was released for ITC-2007. The main features of these instances[2] are summarized in Table 5 [adapted from Battistutta et al., 2017].

File formats and repositories The file format is a single-file text-only one, created ad-hoc for the ITC-2007 competition. Although the format is better engineered than the simple one of previous competitions, still there are some fragilities, as it is witnessed by the presence of incoherent data in two of the competition instances[3]. Fortunately, these inconsistencies do not affect the significance of those instances but the meaning of the two soft constraint types involved is irrelevant since they are unsatisfiable.

The files are available from the ITC-2007 website and also from OPTHUB, where there are also a few solutions listed.

[2]Note that the number of exams reported by Burke and Bykov [2016, Table 2] is overestimated as they consider the largest student identifier, but some numbers are missing in the file.

[3]Instances 6 and 8 have a number of Front Period (both instances) and Period Spread (instance 6 only) larger than the number of periods (highlighted in boldface). This implies that the corresponding soft constraints are always violated, independently of the solution.

Inst.	**E**	**S**	**P**	**R**	$\mathbf{P}_{HC}$	$\mathbf{R}_{HC}$	**FP/FE**	**PS**	**Cd**	**ER**	**SE**	**S/Cap**	**PC**
1	607	7883	54	7	12	0	30/100	5	0.05	1.61	53.3	0.75	15.9
2	870	12484	40	49	8	2	30/250	1	0.01	0.44	43.0	0.23	26.5
3	934	16365	36	48	82	15	20/200	4	0.03	0.54	65.5	0.33	34.1
4	273	4421	21	1	20	0	10/50	2	0.15	13.00	79.6	0.86	19.1
5	1018	8719	42	3	27	0	30/250	5	0.01	8.08	33.6	0.34	72.6
6	242	7909	16	8	22	0	**30**/25	**20**	0.06	1.9	76.31	0.56	35.0
7	1096	13795	80	15	28	0	30/250	10	0.02	0.91	41.5	0.22	43.6
8	598	7718	80	8	20	1	**100**/250	15	0.05	0.93	52.5	0.43	177.0
9	169	624	25	3	10	0	10/100	5	0.08	2.25	15.0	0.60	32.8
10	214	1415	32	48	58	0	10/100	20	0.05	0.14	36.7	0.13	89.4
11	934	16365	26	40	83	15	20/400	4	0.03	0.90	65.5	0.48	51.1
12	78	1653	12	50	9	7	5/25	5	0.18	0.13	47.2	0.20	36.7

Table 5: Features of the ITC-2007 benchmark instances. Symbol definition: **E** (Exams), **S** (Students), **P** (Periods), **R** (Rooms), $\mathbf{P}_{HC}$ (Periods Hard Constraints), $\mathbf{R}_{HC}$ (Rooms Hard Constraints), **FP / FE** (Frontload Periods / Frontload Exams), **PS** (Periods Spread), **Cd** (Conflict density), **ER** (Exams per Room), **SE** (Students per Exam), **S/Cap** (Students to Capacity Ratio), **PC** (Period Conflict). The values highlighted in boldface are incoherent with the number of available periods.

Other datasets and generators A set of 8 instance coming from Yeditepe University and proposed by Özcan and Ersoy [2005] have been translated from their original format to the ITC-2007 one by Parkes and Özcan [2010]. They are available at `http://www.cs.nott.ac.uk/~pszajp/timetabling/exam/`.

These instance are relatively small, with a maximum size of 210 exams. In addition, they are obtained from a simpler formulation, so that many features of the ITC-2007-ETT formulation are unused. Up to our knowledge, no other real-world instances have been introduced later on for the problem.

An instance generator has been developed by Battistutta et al. [2017] for tuning purposes, and a set of 50 challenging artificial instance has been made public on OPTHUB.

The original solution checker provided for ITC-2007 is not available anymore, but solutions can be validated from OPTHUB.

Discussion As mentioned by McCollum et al. [2007], this formulation is a significant step forward the use of complete formulations for standard problems. Indeed, with respect to its predecessors it includes many novel real-world features, in particular of British universities, even though, for the aim of simplicity, some aspects are still left out.

This is also the first formulation of our list that has the weights of the different objectives written in the instance, rather than fixed for all scenarios. As written by McCollum et al. [2007]: "this is motivated by our experience that different institutions do indeed have different weights, and so no one set would be completely useful". Still from McCollum et al. [2007]: "We hope that this will encourage the development of solvers that are robust rather than potentially over-tuned to one particular set of weights for a dataset."

3.5 High-School Timetabling (XHSTT)

Our next formulation was introduced by Post et al. [2012] as an attempt to create a unified formulation and data format for the HTT problem. The proposed formulation, called XHSTT, is extremely rich and the intent is to avoid, differently from the previous formulations, any concession to judicious simplifications. In fact, the proposing team is composed by researchers from various countries, with the aim of including the features coming from as many different situations as possible all around the world.

XHSTT has also been used as the subject of the ITC-2011 competition, which has led to a boost for its spread in the scientific community. In fact, XHSTT is the most popular formulation compared to other ones among the community, and it has drawn the attention of many researchers, in particular after ITC-2011. In addition, the dataset is diverse and quite challenging, also compared to previous ones.

Over the years, several versions of the archive have been collected in the XHSTT project, each one mainly based on the previous one with some improvements on the current instances (name change, format simplification, error correction, redundancy removal, ...) and some new instances. As a consequence, in some cases authors have competed on slightly different versions of the same instances, so that a comprehensive and fair comparison has been made not possible. Indeed, Kristiansen et al. [2015] wrote: "such updates to the format make it hard for researchers to compare computational results with those previously reported".

For the full problem specification, we refer to the work by Post et al. [2012] and to the XHSTT website `https://www.utwente.nl/hstt/`, which contains also some updates with respect to the original formulation.

Short specification As mentioned above, the formulation is complex, so that it is quite difficult to discuss it in brief. Basically, it includes three types of entities: times, resources (students, classes, teachers, and rooms), and events. For each of these three, it is possible to define sets of atomic elements, and use these sets to express complex constraints and objectives.

In XHSTT there are 15 different types of constraints, which range from spreading lectures in the week, to student idle times, to preferences and unavailabilities. For brevity, we do not list them here and refer again to the XHSTT website for their comprehensive specification. Each individual constraint can be declared either hard (Required) or soft (non-Required).

Benchmarks As benchmarks we consider the current version of the archive at the XHSTT website, called XHSTT-2014. As mentioned in the website: "XHSTT-2014 contains a carefully selected subset of the instances collected during this project, in their most up-to-date form".

Inst.	**T**	**Te**	**R**	**C**	**S**	**E**	**D**
AU-BG-98	40	56	45	30	—	387	1564
AU-SA-96	60	43	36	20	—	296	1876
AU-TE-99	30	37	26	13	—	308	806
BR-SA-00	25	14	—	6	—	63	150
BR-SM-00	25	23	—	12	—	127	300
BR-SN-00	25	30	—	14	—	140	350
DK-FG-12	50	90	69	—	279	1077	1077
DK-HG-12	50	100	71	—	523	1235	1235
DK-VG-09	60	46	53	—	163	918	918
ES-SS-08	35	66	4	21	—	225	439
FI-PB-98	40	46	34	31	—	387	854
FI-WP-06	35	18	13	10	—	172	297
FI-MP-06	35	25	25	14	—	280	306
GR-PA-08	35	19	—	12	—	262	262
IT-I4-96	36	61	—	38	—	748	1101
KS-PR-11	62	101	—	63	—	809	1912
NL-KP-03	38	75	41	18	453	1156	1203
NL-KP-05	37	78	42	26	498	1235	1272
NL-KP-09	38	93	53	48	—	1148	1274
UK-SP-06	25	68	67	67	—	1227	1227
US-WS-09	100	134	108	—	—	628	6354
ZA-LW-09	148	19	2	16	—	185	838
ZA-WD-09	42	40	—	30	—	278	1353

Table 6: Features of the XHSTT benchmark instances. Symbol definition: **T** (Times), **Te** (Teachers), **R** (Rooms), **C** (Classes), **S** (Students), **E** (Events), **D** (Duration).

This archive is composed of 25 instances. We removed two of them, namely GR-H1-97 and GR-P3-10, for which a perfect solution (i.e., having zero cost) can be easily obtained. The main features of the remaining 23 instances are shown in Table 6 taken from the XHSTT website. The symbol "—" means that the corresponding resource group is omitted in the particular instance.

File formats and repositories Instances are written in an XML file format, which includes also a metadata part. Thanks to the flexibility of XML, many instances and solutions can be inserted in a single file.

All instances, lower bounds, and best solutions are available at the XHSTT website, including a checker that validates a solution and writes a report of the corresponding violations.

Other datasets and generators Other instances have been contributed from the community over the years and they have been included in the XHSTT website, but they are currently considered less interesting. There are also a few artificial ones obtained by translating instances from other formulations, which do not use most of the constraint types. All the instances are available from the XHSTT website. Up to our knowledge, no artificial instance generator is available.

Discussion As mentioned above, XHSTT is a full-fledged real-world problem with all possible constraints and objectives included. The spirit of this effort is to consider all possible constraints in use somewhere in the globe, allowing the possibility to produce instances that use only a subset of the constraints for its specification. The formulation is still evolving, with student sectioning and different campuses as candidate new features.

A drawback of this choice is that is it rather labor demanding to implement an effective solver for the complete specification of XHSTT. However, a solver could also be developed to deal with only a subset of the possible constraint types.

A limit of the benchmark dataset is that nowadays many instances are solved to proven optimality, so that the competition is moved mainly to the performance of solvers under specific timeouts. An alternative standard formulation, which has recently gained some attention, is the Brazilian one introduced by Saviniec and Constantino [2017], mentioned in Section 3.7, and described in the survey by Tan et al. [2021].

3.6 University Course Timetabling (ITC-2019)

Our last standard formulation is the one of the CTT problem proposed by Müller et al. [2018] for the ITC-2019 competition, that we call ITC-2019. This formulation actually represents a combination of CTT with the student sectioning problem. The formulation is indeed rather rich and structured, and it represents a big step forward bridging the gap between theory and practice of timetabling research. Nonetheless, it still cannot be considered a totally complete problem, as the authors themselves write "to make the problems more attractive, we remove some of the less important aspects of the real-life data while retaining the computational complexity of the problems".

Short specification ITC-2019 consists in sectioning students into classes based on courses enrollments, and then assigning classes to available periods and rooms, respecting various constraints and preferences.

The main novelty is that courses may have a complex structure of classes, with one or more configurations, further divided in subparts, and parent-child relationship between classes. For each class, it is also specified the list of possible periods and rooms for meetings.

The other remarkable feature is that the timetable may differ from week to week, differently from CB-CTT and PE-CTT where the very same weekly timetable is replicated for the whole semester. This feature is present in many practical situations as it allows the institution to gain flexibility in the organization.

Lastly, there are many *distribution* constraints that are evaluated between individual pairs of classes, or all classes as a whole. Distribution constraints may affect the time of the day, the days of the week, the week of the semester, or the room assigned [see Müller et al., 2018, §3.5].

A penalty is associated to the selection of a room and a period for a class, so that the objective function is composed by four main components: (i) class/period penalization, (ii) class/room penalization, (iii) violations of distribution constraints, and (iv) student conflicts.

Benchmarks Instances come from ten institutions, including Purdue University (USA), Masaryk University (Czech Republic), AGH University of Science and Technology (Poland), and Istanbul Kültuür University (Turkey). The real-life data was properly anonymized and simplified as discussed below.

The dataset is composed by 30 instances from ITC-2019 (10 early, 10 middle, 10 late) with very different features in terms of size of the problem (number of classes, students and rooms), room utilization, student course demand, course structure, time patterns, travel times and distribution constraints. Such diversity reflects the different sources of the data, both for the type of institution (school/faculty/entire university) and geographical position. Table 7 reports a selection of the instance features available from a more comprehensive list published on the competition website.

File formats and repositories Instances are written in XML format and available from the competition website (`https://www.itc2019.org`) after registering. In addition, the winners of ITC-2019 have implemented a preprocessing procedure for the ITC-2019 datasets [Holm et al., 2019] that reduce instances to a simplified, though still complete, form. The reduced ITC-2019 datasets are available at `https://dsumsoftware.com/itc2019/`.

Other datasets and generators Up to our knowledge, there are no other instances available apart from the six test instances provided in the competition website.

Discussion Although the formulation mostly adhere to reality, some aspects of real-life data have been neglected or transformed into existing constraints in order to make the formulation easier to model and to work on it. The most

Inst.	Co	Cl	R	S	W	CoS	ClS	TCl	RCl
agh-fis-spr17	340	1239	80	1641	16	8.17	16.2	117.73	15.92
agh-ggis-spr17	272	1852	44	2116	16	6.98	29.92	25.2	7.28
bet-fal17	353	983	62	3018	16	6.24	9.08	23.77	25.43
iku-fal17	1206	2641	214	0	14	—	—	35.36	30.76
mary-spr17	544	882	90	3666	16	2.88	2.9	13.98	13.57
muni-fi-spr16	228	575	35	1543	15	6.24	10.06	16.62	4.82
muni-fsps-spr17	226	561	44	865	19	7.76	11.6	20.24	3.15
muni-pdf-spr16c	1089	2526	70	2938	13	8.72	17.35	59.61	11.82
pu-llr-spr17	697	1001	75	27018	16	3.03	3.4	9.28	15.23
tg-fal17	36	711	15	0	14	—	—	25.86	4.41
agh-ggos-spr17	406	1144	84	2254	16	7.01	13.94	93.58	10.92
agh-h-spr17	234	460	39	1988	16	2.6	4.18	236.35	25.47
lums-spr18	313	487	73	0	20	—	—	43.86	27.19
muni-fi-spr17	186	516	35	1469	14	6.22	10.3	18.92	5.25
muni-fsps-spr17c	116	650	29	395	14	6.98	32.94	124.74	5.06
muni-pdf-spr16	881	1515	83	3443	13	9.2	10.04	32.76	17.47
nbi-spr18	404	782	67	2293	15	6.03	12.46	38.09	4.83
pu-d5-spr17	212	1061	84	13497	15	1.45	2.46	11.79	8.77
pu-proj-fal19	2839	8813	768	38437	17	4.71	6.95	13.43	9.83
yach-fal17	91	417	28	821	16	5.07	13.14	43.98	4.61
agh-fal17	1363	5081	327	6925	18	8.7	20.91	75.55	10.52
bet-spr18	357	1083	63	2921	16	6.52	10.46	23.17	25.15
iku-fal18	1290	2782	208	0	13	—	—	32.72	27.72
lums-fal17	328	502	73	0	20	—	—	43.5	26.54
mary-fal18	540	951	93	5051	16	4.16	4.17	11.37	15.11
muni-fi-fal17	188	535	36	1685	13	6.59	10.43	16.3	4.94
muni-fspsx-fal17	515	1623	33	1152	21	8.87	21.82	67.85	4.42
muni-pdfx-fal17	1635	3717	86	5651	13	9.84	15.94	66.74	18.48
pu-d9-fal19	1154	2798	224	35213	15	3.51	4.37	13.89	14.24
tg-spr18	44	676	18	0	16	—	—	23.37	5.67

Table 7: Features of the ITC-2019 benchmark instances. Symbol definition: **Co** (Course), **Cl** (Classes), **R** (Rooms), **S** (Students), **W** (Weeks), **CoS** (average Courses for Student), **ClS** (average Classes for Student), **TCl** (average Times of a Class), **RCl** (average Rooms of a Class).

important changes involve the computation of the list of rooms available for a class and their individual penalties, travel times, translation of distribution constraints, and student reservation.

3.7 Other Formulations

We now review the additional problem formulations that provide real-world instances that are publicly available. Table 8 shows the list of available ones, up to our knowledge, along with the information whether the solutions and a solution checker are available.

It is worth mentioning that there are many papers claiming that the search method has been applied to real-world cases, but then they do not provide the corresponding files (mainly for privacy issues). There are also many cases in which the link for retrieving the instances is not working anymore, typically due to authors changing affiliation. The latter phenomenon clearly show that the strategy of posting data in author's website does not work in the long run. In some cases, the link has been restored by the authors upon our specific request.

4 State-of-the-Art Results

In this section, we report the results for each of the formulations introduced in Sections 3.1 — 3.6. For each formulation, among all results in the literature, we select and report the ones that we consider "state-of-the-art", intending with this term those that have the best scores for some instances. However, in this selection we take into account also the running time, thus including also results that are worse than others but obtained with significantly shorter time.

For each contribution, we show, if available, the average and the best scores for each instance, along with the running time (when relevant). Further details, such as the computing speed and the number of threads of the machines are neglected, and can be retrieved (if reported) in the corresponding articles.

Reference	Prob	#Inst	Sol	Check	Format	Source	link
Beligiannis et al. [2008]	HTT	11	×	×	text	Greece	https://www.dropbox.com/s/rolhmd31bmrea4a/Input%20instances.zip
Rudová et al. [2011]	CTT	50	√	√	XML	Purdue (US)	https://www.unitime.org
Müller [2016]	ETT	9	√	√	XML	Purdue (US)	https://www.unitime.org
Woumans et al. [2016]	ETT	1	√	×	Excel	Belgium	https://www.kuleuven.be/cv/personallinks/u0038694e.htm
Saviniec and Constantino [2017]	HTT	34	×	×	XML	Brazil	https://www.gpea.uem.br/benchmark.html
Lemos et al. [2019]	CTT	8	√	√	XML	Lisbon (PT)	https://github.com/ADDALemos/MPPTimetables
Battistutta et al. [2020]	ETT	40	√	√	JSON	Italy	https://bitbucket.org/satt/examtimetablinguniuddata
Güler et al. [2021]	CTT	1	×	×	Excel	Yıldız (TR)	https://sites.google.com/view/mgguler/datasets

Table 8: Other formulations and datasets. #Inst: number of instances, Sol: solutions available (√ = Yes, × = No), Check: checker available, Source: single institution or country in case of many institutions.

The tables include also, when available, the best lower bound and the best known result (upper bound), specifying also the researchers that have found them. In addition, the lowest best values are in italics and the proven optimal solutions are underlined (except for perfect solutions). Finally, top average results in the table are in boldface.

4.1 Results on UETT

The state-of-the-art results on Toronto benchmarks described in Section 3.1 are shown in Table 9. The last two columns report the LBs and the best UBs. The UBs are obtained by several authors, whereas the LBs are all obtained by Gogos et al. [2021]. The letter beside each UB value indicates who are the authors: "B" stands for Burke et al. [2010b], "BB" for Burke and Bykov [2016], "L" for Leite et al. [2018] and "BC" for Bellio et al. [2021].

We remark that there are some early results for which it is not clear whether they were obtained on the original input data (see discussion on Section 3.1). Therefore, we decided to remove them and to bound to fully trustworthy results only.

The proposed methods have different running times, reported in the right-most three columns of Table 9. Therefore a completely fair comparison is not possible, given that UETT is particularly sensible to the running time. In fact, longer runs consistently produce results better than shorter ones. As a consequence, highlighted values do not identify univocally the "best" contributions, as they compete with different timeouts. For this reason, for Bellio et al. [2021] we report the results of both the short and long runs, even though the short ones are clearly inferior to the long ones, but can be considered as competitive for the allotted time.

We also notice that all methods are metaheuristics, and there are no approaches such as mathematical and constraint programming among the most successful ones. As we will see in the next sections, this is not the case for some of the other formulations (see Sections 4.3, 4.5, and 4.6).

We can see that, unfortunately, the LBs [Gogos et al., 2021] are not particularly tight, leaving room for improvements.

4.2 Results on PE-CTT

For the PE-CTT formulation, the results that we consider state-of-the-art are shown in Tables 10, 11 and 12, for the two datasets identified as benchmarks in Section 3. The second dataset is split into two tables because the first set of 16 instances and the second one of 8 instances have been considered by different authors, as the latter have been released at a later stage.

All results are obtained from 31 runs, using the time limit allowed by the competition benchmark program (about 300s). For each instance, the top average result is shown in boldface, whereas the lowest best value is shown in italic. For the ITC-2002 benchmarks, the column UB reports the best known value, which in this case is the lowest value in the table, except for instance 1 for which it has been obtained by Goh et al. [2017] with longer (five times) timeout, and instances 10 and 11 obtained by Nagata [2018] using a method different from the most performing one reported here.

Table 9: Results on Toronto benchmarks of UETT.

	Bellio et al. 2021			Burke and Bykov 2008			Mandal et al. 2020			Burke and Bykov 2016		Leite et al. 2018		Bellio et al. 2021		Gogos et al. 2021	
Inst.	avg	best		avg	best		avg	best		avg	best	avg	best	avg	best	LB	UB
car91	4.44	4.38	s	4.68	4.58	s	4.72	4.58	h	4.34	4.32	4.39	4.31	**4.27**	*4.24*	0.0059	4.237932^{BC}
car92	3.8	3.75	s	3.92	3.81	s	3.93	3.82	h	3.7	3.67	3.72	3.68	**3.68**	*3.64*	0.0079	3.642109^{BC}
ear83	32.89	32.61	s	32.91	32.65	s	34.49	33.23	h	32.66	32.62	32.61	32.48	**32.60**	*32.42*	18.2596	32.420444^{BC}
hec92	10.16	10.05	s	10.22	10.06	s	11.09	10.32	h	10.12	10.06	**10.05**	*10.03*	**10.05**	*10.03*	3.8162	$10.033652^{L.BC}$
kfu93	13.06	12.87	s	13.02	12.81	s	13.97	13.34	h	12.85	*12.8*	**12.83**	*12.81*	12.88	*12.81*	5.736	12.799028^{BC}
lse91	10.09	9.92	s	10.14	9.86	s	10.62	10.24	h	9.84	*9.78*	9.81	*9.78*	**9.80**	*9.78*	3.3555	9.773661^{BC}
pur93	4.32	4.22	s	4.71	4.53	s				**3.91**	*3.88*	4.18	4.14	4.02	4	0.0014	3.88^{BB}
rye93	8.1	7.99	s	8.06	7.93	s	10.29	9.79	h	7.94	7.91	7.93	7.89	**7.91**	*7.84*	3.7868	7.837586^{BC}
sta83	157.05	*157.03*	s	157.05	*157.03*	s	157.64	157.14	h	157.04	*157.03*	**157.03**	*157.03*	**157.03**	*157.03*	152.0458	156.86^{B}
tre92	7.85	7.72	s	7.89	7.72	s	8.03	7.74	h	7.68	7.64	7.7	7.66	**7.66**	*7.59*	0.8601	7.590367^{BC}
uta92	3.13	3.05	s	3.26	3.16	s	3.22	3.13	h	3.01	2.98	3.04	3.01	**2.97**	*2.95*	0.0022	2.947193^{BC}
ute92	24.82	*24.76*	s	24.82	24.79	s	26.04	25.28	h	24.82	24.78	24.83	24.8	**24.79**	*24.76*	21.5993	24.76^{BC}
yor83	34.93	34.56	s	36.16	34.78	s	36.79	35.68	h	34.79	34.71	34.63	34.45	**34.57**	*34.40*	19.1435	34.404888^{BC}
Time																	
Min	130.8		s	450.0		s				4.6	h	24	h	26.2	h		
Max	1382.0		s	901.0		s				5.7	h	48	h	52.2	h		
Avg	413.1		s	654.6		s	1		h	5.1	h	31.4	h	34.7	h		

Table 10: Results on ITC-2002 benchmarks of PE-CTT.

	Kostuch 2005	Goh et al. 2017		Nagata 2018		Goh et al. 2019		Goh et al. 2020		
Inst.	best	avg	best	avg	best	avg	best	avg	best	UB
01	*16*	32.6	23	**30.2**	*16*	37	26	36.8	29	10
02	*2*	13.7	7	**11.4**	*2*	16.3	6	16.2	*2*	2
03	*17*	36.4	26	**31**	*17*	38.2	27	34.3	24	17
04	*34*	63.1	50	**60.8**	*34*	69	47	70.7	46	34
05	42	58.6	38	72.1	42	**51.8**	*36*	55	43	36
06	0	0.8	0	2.4	0	0.8	0	**0.4**	0	0
07	2	2.6	0	8.9	2	**2.4**	0	**2.4**	0	0
08	0	**1.4**	0	2	0	1.5	0	2.2	0	0
09	1	**4.6**	0	5.8	2	6.4	0	6.5	0	0
10	*21*	40.9	28	**35**	*21*	40.4	22	39.2	26	18
11	*5*	17.7	10	**12.9**	*5*	19	10	19.7	9	4
12	55	64.5	53	76.3	55	64.1	47	**63.9**	*46*	46
13	*31*	53.3	38	**47.1**	*31*	51	33	51.2	40	31
14	11	12.9	5	22.3	11	13.6	*4*	**12.1**	*4*	4
15	2	**4.0**	0	8.4	2	4.8	0	4.4	0	0
16	0	**0.5**	0	3.4	0	2.2	0	1.6	0	0
17	37	41.6	26	54	37	**36.8**	25	38.7	*24*	24
18	4	9.7	*2*	**9.4**	4	12.5	3	11.7	4	2
19	*7*	24.7	11	**16.4**	*7*	25.6	15	23.6	9	7
20	0	**0**	0	0.5	0	**0**	0	**0**	0	0
Avg		24.18		25.52		24.67		24.53		

Table 11: Results on ITC-2007 public benchmarks of PE-CTT.

	Mayer et al. 2008		Goh et al. 2017		Nagata 2018		Goh et al. 2019		Goh et al. 2020	
Inst.	avg	best	avg	best	avg	best	avg	best	avg	best
01	613	0	307.6	0	**81.7**	0	209.4	0	191.8	0
02	556	0	63.4	0	48	0	10.1	0	**1.7**	0
03	680	110	199.4	163	**155**	*55*	188.6	141	189.8	137
04	580	53	328.8	242	**254.1**	*10*	320.9	192	315.5	24
05	92	13	2.7	0	**0**	0	2.9	0	2.9	0
06	212	0	33.2	0	**0**	0	54.7	0	37.6	0
07	4	0	18	5	**3.6**	0	14.5	4	16.2	5
08	61	0	**0**	0	**0**	0	1.6	0	5.7	0
09	202	0	100.7	0	**58.9**	0	15.2	0	2.6	0
10	**4**	0	65.3	0	6.4	0	30.5	0	16.3	0
11	774	143	244.3	161	**140.4**	*3*	201.6	136	199.6	21
12	538	0	318.2	0	**33.1**	0	303.5	0	258.1	0
13	360	5	99.5	0	**0**	0	90.4	0	85.9	0
14	41	0	0.2	0	**0**	0	25.6	0	17.8	0
15	29	0	192	0	**0**	0	12.5	0	9.3	0
16	101	0	105.8	10	**1.5**	0	45.8	0	40.2	0
Avg	302.9		129.9		48.9		95.5		86.9	

Table 12: Results on ITC-2007 hidden benchmarks of PE-CTT.

	Cambazard et al. 2010		Ceschia et al. 2012		Lewis and Thompson 2015		Goh et al. 2017		Nagata 2018		Goh et al. 2019		Goh et al. 2020	
Inst.	avg	best	avg	best	avg	best	avg	best	avg	best	avg	best	avg	best
17	4.9	0	**0.0**	0	0.07	0	0.8	0	0.2	0	0.5	0	0.1	0
18	14.1	0	41.1	0	2.16	0	12.5	0	**0.5**	0	7.7	0	15.5	0
19	2027.0	1824	951.5	0	346.08	0	516.7	0	616.8	0	**11**	0	79.6	0
20	505.0	445	700.2	543	724.54	557	650.7	586	**482**	*438*	664	555	661.5	579
21	27.1	0	35.9	5	32.09	1	12.5	0	**0.1**	0	25.7	0	14.8	0
22	550.8	29	19.9	5	1790.08	4	136	1	35	0	**5.8**	0	22.6	0
23	**330.5**	238	1707.7	1292	514.13	0	504.4	11	1083.5	777	713.6	56	531.7	0
24	124.2	21	105.3	0	328.18	18	192.6	5	1	0	**77.5**	0	102.1	0
Avg	448.0		445.2		467.2		253.3		277.4		188.2		178.5	

We do not report the UBs in Tables 11 and 12 as most of them are equal to 0 (for instance 11 by Lewis and Thompson 2015, not in the table). The only distinctive instances are 3, 4, and 20 with UB values 55, 10 (reported in the corresponding table), and 150 (found by Nagata 2018 with another method), respectively. For ITC-2002, conversely, for many instances, the perfect solution is still to be found.

The first comment on these tables is that all best results have been found by local search methods, namely Tabu Search [Nagata, 2018] and Simulated Annealing [Goh et al., 2017, 2019, 2020]. In general, best results are obtained by Nagata [2018], that uses a composite neighborhood and *elite candidate* rules to reduce the computational cost of the full neighborhood exploration prescribed by Tabu Search. Good results are obtained also by Goh et al., mainly using random move selection.

Goh et al. [2019] report also the results for longer running times (i.e., five times longer), which are not shown here. Unsurprisingly, both the best and average cost are remarkably improved when the execution time is extended.

It is worth noticing that the fact that all instances have a perfect (zero cost) solution might bias the search methods toward certain specific strategies. For example, the objective that penalizes all lectures in the last period of the day might be exploited, by removing such periods completely from the search space.

4.3 Results on CB-CTT

Table 13: Results on ITC-2007 benchmarks of CB-CTT.

	Abdullah and Turabieh 2012		Kiefer et al. 2017		Lindahl et al. 2018		
	avg	best	avg	best	avg	LB	UB
comp01	**5.00**	*5*	**5.0**	*5*	12.0	$5^{C,B1}$	5^{*}
comp02	**36.36**	*26*	41.5	34	49.5	24^{B2}	24^{A}
comp03	74.36	70	**71.7**	*68*	74.5	58^{B3}	64^{K}
comp04	38.45	*35*	**35.1**	*35*	38.5	35^{*}	35^{*}
comp05	314.45	295	**305.2**	*294*	373.5	247^{B3}	284^{K}
comp06	**45.27**	*30*	47.8	41	58.3	27^{A}	27^{A}
comp07	**12.00**	*7*	14.5	10	35.0	6^{*}	6^{A}
comp08	**40.82**	*37*	41.0	39	49.7	37^{*}	37^{A}
comp09	108.36	102	102.8	*100*	**100.5**	96^{B2}	96^{L1}
comp10	**8.36**	*5*	14.3	7	25.7	4^{*}	4^{A}
comp11	0.00	*0*	0.0	*0*	6.5	0^{*}	0^{*}
comp12	320.27	315	**319.4**	*306*	360.7	248^{B3}	294^{K}
comp13	64.27	*59*	**60.7**	*59*	69.0	59^{*}	59^{A}
comp14	64.36	61	**54.1**	*51*	56.9	51^{*}	$51^{A,L1}$
comp15	72.73	69	**72.1**	*66*	74.5	58^{B3}	62^{K}
comp16	**23.73**	*18*	33.8	26	37.1	$18^{A,B2}$	18^{A}
comp17	76.36	*60*	**75.7**	67	86.1	$56^{A,B3}$	56^{A}
comp18	75.64	69	**66.9**	*64*	72.9	61^{L2}	61^{K}
comp19	66.82	*57*	**62.6**	59	64.8	57^{B2}	57^{L1}
comp20	**13.45**	*7*	27.2	19	34.3	4^{*}	4^{A}
comp21	100.73	*86*	**97.0**	93	103.8	$74^{B2,L2}$	74^{P}
Avg	74.37	67.29	**73.73**	68.71	84.94		

Table 13 shows the best results for CB-CTTbenchmarks obtained using the timeout fixed for the ITC-2007 dataset (300-500 seconds depending on the CPU). Longer runs, which unsurprisingly obtain better results, are not considered here [see Lü and Hao, 2009, Asín Achá and Nieuwenhuis, 2014, Song et al., 2021]. We take them into account only for establishing the LBs and UBs, which are shown in the last two columns of the table. In particular, the LBs are obtained with a running time up to 40 times the ITC-2007 timeout.

Besides each best-known lower and upper bound values, we report a letter that indicates who are the authors[4]: "A" stands for Asín Achá and Nieuwenhuis [2014], "B1" for Burke et al. [2010a], "B2" for Bagger et al. [2019b], "B3" for Bagger et al. [2019a], "C" for Cacchiani et al. [2013], "K" for Kiefer et al. [2017], "L1" for Lü and Hao [2009], "L2" for Gerard Lach, "P" for Phillips [2015]. If the same value was found by many different authors, we marked it with the symbol $*$.

[4]We note that the UBs and LBs in Table 4 of Lindahl et al. [2018] (column Best, including the numbers in parentheses) are actually wrong, as they refer to the formulation UD1 instead of UD2 considered in that paper (and here)

We see from Table 13 that almost all current best results are obtained by two contributions, namely Abdullah and Turabieh [2012] and Kiefer et al. [2017], who both proposed metaheuristic methods using Adaptive Large Neighborhood operators. Abdullah and Turabieh implemented a Genetic Algorithm hybridized with Tabu Search employing large neighborhood operators, whose sequence of employment follows a "best" selection strategy, based on previous knowledge about the successful percentage of each neighborhood structure on each instance. Kiefer et al. [2017] presented an Adaptive Large Neighborhood Search algorithm embedded in a Simulated Annealing framework, incorporating several destroy and repair operators, whose selection probability is dynamically biased towards the best-performing ones. Quite a few other papers have produced results that were state-of-the-art at the time of their publication, including [Müller, 2008, Lü and Hao, 2009, Abdullah et al., 2012, Bellio et al., 2016].

As remarked by Bagger et al. [2019a, Table 6], all benchmark instances but 3 are currently solved to optimality[5]. In our opinion, the fact that the optimal value has been found does not undermine the benchmarking role of these instances, which are still challenging for medium-short timeouts. Nonetheless, there are other public instances that are already available (on OPTHUB) that could come up beside the current ones, in order to create a larger, more comprehensive benchmark set (see Section 3.3).

It is worth noticing that this is the only one among our six standard formulations for which there has been a lot of research for finding the best lower bounds.

4.4 Results on ITC-2007-ETT

The state-of-the-art results for ITC-2007-ETT using the ITC-2007 timeout are shown in Table 14. First of all, we notice that the best results are obtained mainly by Bikov and coworkers. They use innovative local search algorithms, such as Late Acceptance and Step Counting Hill Climbing, applied to complex neighborhood structures (such as Kempe chains).

Research on this problem is still active and more recent results are available [see, e.g., Battistutta et al., 2017, Leite et al., 2019, 2021]; however, they do not outperform the previous ones shown in Table 14.

Table 14: Results on ITC-2007 benchmarks of ITC-2007-ETT.

	Burke and Bykov 2016		Bykov and Petrovic 2016	Burke and Bykov 2017	Gogos et al. 2010	Arbaoui et al. 2019
Inst.	avg	best	best	avg	best	LB
1	3792.5	3691	*3647*	**3787**	4128	—
2	**393.1**	385	385	402	*380*	10
3	7611.8	*7359*	7487	**7378**	7769	670
4	**12100.4**	*11329*	11779	13278	13103	1620
5	2512.9	2482	*2447*	**2491**	2513	—
6	25491.5	25265	*25210*	**25461**	25330	22875*
7	3755.1	3608	*3563*	**3589**	*3537*	—
8	6949.9	6818	*6614*	**6701**	7087	1250*
9	**930**	*902*	924	997	913	—
10	**12975.7**	*12900*	12931	13013	13053	0
11	23931.7	*22875*	23784	**22959**	24369	3970
12	**5176.3**	5107	5097	5234	*5095*	2030

The lower bounds are obtained by Arbaoui et al. [2019] by considering only a subset of the soft constraints. In detail, they consider the spacing soft constraints, namely (i) and (ii) mentioned in Section 3.4, and compute the number of violations induced by the largest clique in the corresponding graph. As shown in Table 14, for some instances the method does not produce any result as the largest clique is not big enough to contribute any violation. For instances 6 and 8, marked with an *, we add to the LB computed by Arbaoui et al. (whose original values were 2600 and 0, respectively) the fixed cost of constraints (iii) and (v) due to the fact that there are not enough periods to satisfy them (see Section 3.4 for the detailed explanation).

4.5 Results on XHSTT

Table 15 reports the state-of-the-art results for the benchmark instances of XHSTT. As mentioned in Section 3.5, two instances have been eliminated due to the fact that they are too easy.

[5] In the paper they are actually 4, but as mentioned above, `comp03` and `comp15` are identical in this formulation.

Table 15: Results on the XHSTT-2014 benchmarks of XHSTT.

Inst.	Demirović and Musliu 2017 avg	Demirović and Stuckey 2018 avg	Teixeira et al. 2018 avg	Fonseca et al. 2017 z	Kheiri and Keedwell 2017 best	LB	UB
AU-BG-98			(3, 514)		493	0	128^{G}
AU-SA-96			(16, 91)	*0*	2	0	0^{G}
AU-TE-99			(7, 13)	*20*	61	20^{G}	20^{G}
BR-SA-00	*5*	*5*			10	$5^{L,D}$	5^{L}
BR-SM-00	61.4	88	100		(2, 117)	$51^{L,D}$	51^{L}
BR-SN-00	50.6	66	170		101	35^{D}	35^{D}
DK-FG-12				1300	1522	412^{G}	1263^{G}
DK-HG-12				(12, 2356)	(12, 2628)	7^{G}	$(12, 2330)^{G}$
DK-VG-09				(2, 2329)	(2, 2720)	$(2, 0)^{G}$	$(2, 2323)^{G}$
ES-SS-08				*335*	517	$334^{L,G}$	335^{L}
FI-PB-98	54.6	9			8	0	0
FI-WP-06	9.8	4	1		7	0	0^{G}
FI-MP-06	95.2	90	93		89	$77^{L,V2}$	77^{G}
GR-PA-08	5	7			4	3^{L}	3^{G}
IT-I4-96	35				34	27^{L}	27^{G}
KS-PR-11					3	0	0^{D2}
NL-KP-03			1383	*199*	466	0	199^{G}
NL-KP-05			1056	433	811	$89^{V2,G}$	425^{G}
NL-KP-09				*1620*	(2, 7495)	180^{G}	1620^{G}
UK-SP-06				(5, 4014)	(19, 1294)	0	$(4, 1708)^{S}$
US-WS-09				103	512	101^{G}	101^{G}
ZA-LW-09		*0*			52	0	0^{V}
ZA-WD-09					(9, 0)	0	0^{L}

The first three columns report average results obtained within the competition timeout (1000 secs), whereas Fonseca et al. and Kheiri and Keedwell did not impose a time limit. Besides each best-known lower and upper bound values, we report a letter that indicates who are the authors: "G" stands for the UFOP-GOAL team (Fonseca, Santos, and Carrano), "L" for the Lectio team (Kristiansen, Sørensen, and Stidsen), "V" for the VAGO team (Valouxis, Gogos, Daskalaki, Alefragis, Goulas, and Housos), "D" for Á. P. Dorneles, "V2" for M. de Vos, "D2" for Demirović and Musliu, and "S" for Skolaris (M. Klemsa).

We first notice that most authors have not considered all instances. For example, Fonseca et al. [2017] omit instances whose optimal solution is already known and proven.

Although the ITC-2011 competition was dominated by metaheuristic methods, recently exact methods based on integer programming [Kristiansen et al., 2015, Fonseca et al., 2017, Dorneles et al., 2017], maxSAT [Demirović and Musliu, 2017] and constraint programming [Demirović and Stuckey, 2018] have proven to be very effective for XHSTT. Indeed, differently from the formulations of Sections 3.1–3.4, for XHSTT it ended up being customary to use IP techniques and to evaluate the performance of a solution methods without time limit. Indeed, its best known solutions and lower bounds are updated/improved by the community on the XHSTT website. An up-to-date categorization of the different solution methods applied to HTT (including XHSTT) is presented by Tan et al. [2021].

As mentioned in Section 3.5, the previous versions of the XHSTT archive are deprecated, and thus we do not include them in the benchmarks. However, one of them, namely the hidden dataset of ITC-2011, due to its popularity given by the competition, has been used as testbed by many authors. In particular, there are interesting results by Kristiansen et al. [2015], Fonseca et al. [2016], Demirović and Musliu [2017], and Teixeira et al. [2018]. In addition, LBs have been found by Dorneles et al. [2017].

4.6 Results on ITC-2019

The competition finished in 2020, so the problem is rather new, and the only published results are those of the competition. Differently from previous competitions, the goal of ITC-2019 was to find all-time-best solutions to all competition instances, without time limits or technology restrictions. As a consequence, in Table 16 we re-

Table 16: Results on ITC-2019 benchmarks of ITC-2019.

Inst.	DSU Team LB	UB
agh-fis-spr17	1336	3039^{D}
agh-ggis-spr17	23164	34285^{D}
bet-fal17	89278	289965^{D}
iku-fal17	18001	18968^{D}
mary-spr17	14359	14910^{D}
muni-fi-spr16	3556	3756^{D}
muni-fsps-spr17	868	$\underline{868}^{D}$
muni-pdf-spr16c	14279	33724^{D}
pu-llr-spr17	10038	$\underline{10038}^{D}$
tg-fal17	4215	$\underline{4215}^{U}$
agh-ggos-spr17	1844	2864^{D}
agh-h-spr17	8945	21559^{D}
lums-spr18	24	95^{D}
muni-fi-spr17	2500	3825^{D}
muni-fsps-spr17c	1361	2596^{D}
muni-pdf-spr16	13008	17208^{D}
nbi-spr18	18014	$\underline{18014}^{D}$
pu-d5-spr17	6981	15204^{M}
pu-proj-fal19	54972	117425^{M}
yach-fal17	516	1074^{M}
agh-fal17	5728	118038^{M}
bet-spr18	63444	348524^{D}
iku-spr18	25781	25863^{D}
lums-fal17	254	349^{D}
mary-fal18	3496	4331^{D}
muni-fi-fal17	1890	2999^{D}
muni-fspsx-fal17	7747	10123^{M}
muni-pdfx-fal17	26711	98373^{M}
pu-d9-fal19	28000	39942^{D}
tg-spr18	12704	$\underline{12704}^{D}$

ported only UBs, whose solutions are collected (and continuously updated) on the competition website (`https://www.itc2019.org/`).

The competition was won by the DSU team [Holm et al., 2019] who devised a Fix-and-Optimize matheuristic, which was able to find all best solutions except for one instance (agh-fal17). In addition, the DSU teams maintains a website (`https://dsumsoftware.com/itc2019/`) reporting their current best results and the lower bounds (showed on Table 16). The second place was obtained by Rappos et al. [2019] who modeled the problem as MIP enhanced with some preprocessing techniques that improve its efficiency. The third place was occupied by Gashi and Sylejmani [2019] who presented a Simulated Annealing algorithm[6].

The letter beside each UB value in Table 16 indicate the authors: "D" stands for the DSUM team (Holm, Mikkelsen, Sørensen, and Stidsen), "U" for the UFOP team (M, A. Pires, H. Gambini Santos, T. A.M. Toffolo), and "M" for Müller [2020].

5 Conclusions and Future Directions

The quest for formulations and benchmarks carried out for this survey has brought out various aspects of the current practice in timetabling research. We summarize here our observations, and we split them in three groups regarding the standard formulations, the specific formulations, and the solution techniques, respectively. In our opinion, these observation can serve as starting points for future research directions.

Key observations about standard formulations:

[6]Their source code is available at `https://github.com/edongashi/itc-2019`

A. Most of the standard formulations arose from timetabling competitions, which have given the necessary initial boost in terms of infrastructure and promotion.

B. For some of the standard formulations, the benchmark instances are not challenging anymore, as they are too easily solved to optimality. Others, on the contrary, are still very challenging after more than 20 years from their publication.

C. There is a clear trend in the timetabling community to move toward rich formulations, getting rid of strong simplifications. In our opinion, this is a positive trend, but should be paired with the maintenance and renewal of the simple formulations, that could still serve as better testbeds for comparisons.

Moving to the contributions introducing specific formulations, we have the following observations:

D. Many of the papers discussing original formulations do not provide publicly available data. For others, the original repository has become inaccessible after some time from the publication of the paper. Finally, in other cases, the file formats are too cumbersome and not sufficiently documented, to be easily usable for other researchers.

E. Most formulations are too specific for the particular case at hand without consideration of wider application, so that it is difficult to gain general insights from the papers. In addition, in some cases the precise formulation is not completely explained, so that it is not possible for other researchers to replicate the same model and to obtain comparable results.

F. For most formulations, the solutions are not made available, and thus the results in the papers could not be validated. In addition, the source code of the search method is very rarely available, so that the experiments cannot be replicated.

Regarding the comparison of solution techniques, we make the following observations:

G. There is a need for the clear definition of the competition grounds, in terms of running time, statistical significance, computing architecture, usable technology, commercial licenses, and other issues. In the formulations coming form the competitions, the ground has been set by the official competition rules, which however might need to be refined and extended in order to do not harness future research.

H. The results of Section 4 clearly show that both exact and (meta)heuristic techniques have their role and their chance to emerge, depending on the specific formulation and the competition ground.

I. There is a need for new benchmarks that could take over for the ones that turned out to be too easy for state-of-the-art techniques.

J. There is also need for more instances that could be used for the statistically-principled tuning of the solution methods, letting the benchmarks to be used only for the validation phase (avoiding overtuning). To this aim, the use of high quality generators could also help, as them could provide an unlimited number of instances.

All above points together highlight the need for the development of research infrastructures in terms of common formulations, robust file formats, long-term web repositories with instances and solutions, generators, and solution checkers. The implementation of a wholesome and robust infrastructure of this type is clearly too expensive in terms of human effort to be left to the initiative of single research groups. Therefore, there is the need for coordinated community-level actions, in order to develop an infrastructure and, at the same time, create the necessary consensus upon its adoption. In our opinion, to this aim, the organization of future timetabling competitions could still be the right key to pursuit this task.

Another point that emerged from our analysis is the issue of the reproducibility and trustworthiness of results. In fact, the risk of reporting false results has emerged significantly, though mainly in the early times of the timetabling research. In any case, it is still important that data is available for both inspection and future comparisons. This is indeed a general issue that is ubiquitous in many research areas, as journals currently push for publication of data along with the papers.

We are trying to give our contribution for solving these issues by the development of the web application OPTHUB, which provides a common platform able to host new problems with their instances and solutions. Solutions in OPTHUB are immediately validated and made available to the community.

OPTHUB is an ongoing project, and hopefully will be extended significantly in future releases. The main future feature will be include in a new version is the possibility to upload the software and to run it (also on behalf of other researchers). Hopefully, this option will allow the community to make fairer comparisons and statistical analyses on the behavior of the solution code.

Acknowledgments

We wish to thank all the people from the timetabling community that have helped us by pointing out to us articles, datasets, and results. We will name all of them in the acknowledgments of the final version of this article.

References

Salwani Abdullah and Hamza Turabieh. On the use of multi neighbourhood structures within a tabu-based memetic approach to university timetabling problems. *Information Sciences*, 191:146–168, 2012. ISSN 0020-0255. Data Mining for Software Trustworthiness.

Salwani Abdullah, Hamza Turabieh, Barry McCollum, and Paul McMullan. A hybrid metaheuristic approach to the university course timetabling problem. *Journal of Heuristics*, 18(1):1–23, 2012. ISSN 1381-1231. doi: 10.1007/s10732-010-9154-y.

Babak Akbarzadeh and Broos Maenhout. A decomposition-based heuristic procedure for the medical student scheduling problem. *European Journal of Operational Research*, 288(1):63–79, 2021.

Can Akkan and Ayla Gülcü. A bi-criteria hybrid genetic algorithm with robustness objective for the course timetabling problem. *Computers and Operations Research*, 90:22–32, 2018.

Panayiotis Alefragis, Christos Gogos, Christos Valouxis, and Efthymios Housos. A multiple metaheuristic variable neighborhood search framework for the uncapacitated examination timetabling problem. In *Proc. of the 13th Int. Conf. on the Practice and Theory of Automated Timetabling (PATAT-2021, Volume I)*, pages 159–171, 2021.

Taha Arbaoui, Jean-Paul Boufflet, and Aziz Moukrim. Lower bounds and compact mathematical formulations for spacing soft constraints for university examination timetabling problems. *Computers and Operations Research*, 106:133–142, 2019. ISSN 0305-0548. doi: https://doi.org/10.1016/j.cor.2019.02.013.

Roberto Asín Achá and Robert Nieuwenhuis. Curriculum-based course timetabling with SAT and MaxSAT. *Annals of Operations Research*, 218:71–91, February 2014. ISSN 15729338.

N.-C.F. Bagger, M. Sørensen, and T.R. Stidsen. Dantzig–wolfe decomposition of the daily course pattern formulation for curriculum-based course timetabling. *European Journal of Operational Research*, 272(2):430–446, 2019a.

Niels-Christian Fink Bagger, Guy Desaulniers, and Jacques Desrosiers. Daily course pattern formulation and valid inequalities for the curriculum-based course timetabling problem. *Journal of Scheduling*, 22(2):155–172, 2019b.

Michele Battistutta, Andrea Schaerf, and Tommaso Urli. Feature-based tuning of single-stage simulated annealing for examination timetabling. *Annals of Operations Research*, 252(2):239–254, 2017. ISSN 0254-5330.

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

Michele Battistutta, Sara Ceschia, Fabio De Cesco, Luca Di Gaspero, Andrea Schaerf, and Elena Topan. Local search and constraint programming for a real-world examination timetabling problem. In Emmanuel Hebrard and Nysret Musliu, editors, *17th International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pages 69–81. Springer International Publishing, 2020.

Grigorios N Beligiannis, Charalampos N Moschopoulos, Georgios P Kaperonis, and Spiridon D Likothanassis. Applying evolutionary computation to the school timetabling problem: The greek case. *Computers and Operations Research*, 35(4):1265–1280, 2008.

Ruggero Bellio, Sara Ceschia, Luca Di Gaspero, Andrea Schaerf, and Tommaso Urli. Feature-based tuning of simulated annealing applied to the curriculum-based course timetabling problem. *Computers & Operations Research*, 65:83–92, 2016.

Ruggero Bellio, Sara Ceschia, Luca Di Gaspero, and Andrea Schaerf. Two-stage multi-neighborhood simulated annealing for uncapacitated examination timetabling. *Computers and Operations Research*, 132:105300, 2021. ISSN 0305-0548.

Andrea Bettinelli, Valentina Cacchiani, Roberto Roberti, and Paolo Toth. An overview of curriculum-based course timetabling. *TOP*, pages 1–37, 2015.

Alex Bonutti, Fabio De Cesco, Luca Di Gaspero, and Andrea Schaerf. Benchmarking curriculum-based course timetabling: formulations, data formats, instances, validation, visualization, and results. *Annals of Operations Research*, 194(1):59–70, 2012.

E. K. Burke and Y. Bykov. A late acceptance strategy in hill-climbing for exam timetabling problem. In *Proceedings of the 7th international conference on the practice and theory of automated timetabling (PATAT 2008)*, pages 1–7, 2008.

E. K. Burke, J. Mareček, A. J. Parkes, and H. Rudová. Decomposition, reformulation, and diving in university course timetabling. *Computers and Operations Research*, 37(3):582–597, 2010a. ISSN 0305-0548.

Edmund K Burke and Yuri Bykov. An adaptive flex-deluge approach to university exam timetabling. *INFORMS Journal of Computing*, 28(4):781–794, 2016.

Edmund K. Burke and Yuri Bykov. The late acceptance hill-climbing heuristic. *European Journal of Operational Research*, 258(1):70–78, 2017. ISSN 0377-2217.

Edmund K. Burke and Sanja Petrovic. Recent research directions in automated timetabling. *European Journal of Operational Research*, 140(2):266–280, 2002.

Edmund K. Burke, Jakub Mareček, Andrew J. Parkes, and Hana Rudová. Penalising patterns in timetables: Novel integer programming formulations. In Stefan Nickel and Jörg Kalcsics, editors, *Operations Research Proceedings 2007*, Operations Research Proceedings, pages 409–414, Berlin, 2008. Springer.

E.K. Burke, A.J. Eckersley, B. McCollum, S. Petrovic, and R. Qu. Hybrid variable neighbourhood approaches to university exam timetabling. *European Journal of Operational Research*, 206(1):46–53, 2010b. ISSN 0377-2217.

Yuri Bykov and Sanja Petrovic. A step counting hill climbing algorithm applied to university examination timetabling. *Journal of Scheduling*, 19(4):479–492, 2016.

V. Cacchiani, A. Caprara, R. Roberti, and P. Toth. A new lower bound for curriculum-based course timetabling. *Computers and Operations Research*, 40(10):2466–2477, February 2013. ISSN 03050548.

Valentina Cacchiani and Paolo Toth. Nominal and robust train timetabling problems. *European Journal of Operational Research*, 219(3):727–737, 2012.

Hadrien Cambazard, Emmanuel Hebrard, Barry O'Sullivan, and Alexandre Papadopoulos. Local search and constraint programming for the post enrolment-based course timetabling problem. *Annal of Operations Research*, pages 1–25, 2010. ISSN 0254-5330.

M. W. Carter, G. Laporte, and S. Y. Lee. Examination timetabling: Algorithmic strategies and applications. *Journal of the Operational Research Society*, 74:373–383, 1996.

Sara Ceschia, Luca Di Gaspero, and Andrea Schaerf. Design, engineering, and experimental analysis of a simulated annealing approach to the post-enrolment course timetabling problem. *Computers and Operations Research*, 39: 1615–1624, 2012. ISSN 0305-0548.

Mei Ching Chen, San Nah Sze, Say Leng Goh, Nasser R. Sabar, and Graham Kendall. A survey of university course timetabling problem: Perspectives, trends and opportunities. *IEEE Access*, 9:106515–106529, 2021. doi: 10.1109/ACCESS.2021.3100613.

Arnaud De Coster, Nysret Musliu, Andrea Schaerf, Johannes Schoisswohl, and Kate Smith-Miles. Algorithm selection and instance space analysis for curriculum-based course timetabling. *Journal of scheduling*, 2021. online first.

Emir Demirović and Nysret Musliu. Maxsat-based large neighborhood search for high school timetabling. *Computers and Operations Research*, 78:172–180, 2017.

Emir Demirović and Peter J Stuckey. Constraint programming for high school timetabling: a scheduling-based model with hot starts. In *International conference on the integration of constraint programming, artificial intelligence, and operations research*, pages 135–152. Springer, 2018.

Luca Di Gaspero and Andrea Schaerf. Multi-neighbourhood local search with application to course timetabling. In Edmund Burke and Patrick De Causmaecker, editors, *Proc. of the 4th Int. Conf. on the Practice and Theory of Automated Timetabling (PATAT-2002), selected papers*, volume 2740 of *Lecture Notes in Computer Science*, pages 262–275. Springer, 2003.

Luca Di Gaspero, Barry McCollum, and Andrea Schaerf. The second international timetabling competition (ITC-2007): Curriculum-based course timetabling (track 3). Technical report, Queen's University, Belfast (UK), August 2007.

Árton P Dorneles, Olinto CB de Araújo, and Luciana S Buriol. A column generation approach to high school timetabling modeled as a multicommodity flow problem. *European Journal of Operational Research*, 256(3): 685–695, 2017.

George H.G. Fonseca, Haroldo G. Santos, and Eduardo G. Carrano. Integrating matheuristics and metaheuristics for timetabling. *Computers and Operations Research*, 74:108–117, 2016. ISSN 0305-0548.

George H.G. Fonseca, Haroldo G. Santos, Eduardo G. Carrano, and Thomas J.R. Stidsen. Integer programming techniques for educational timetabling. *European Journal of Operational Research*, 262(1):28–39, 2017. ISSN 0377-2217.

E Gashi and K Sylejmani. Simulated annealing with penalization for university course timetabling. In *Proceedings of the International Timetabling Competition 2019*, 2019.

Christos Gogos, George Goulas, Panayiotis Alefragis, Vasilios Kolonias, and Efthymios Housos. Distributed scatter search for the examination timetabling problem. In *PATAT 2010 Proceedings of the 8th International Conference on the Practice and Theory of Automated Timetabling*, pages 211–223, 2010.

Christos Gogos, Angelos Dimitsas, Vasileios Nastos, and Christos Valouxis. Some insights about the uncapacitated examination timetabling problem. In *2021 6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*, pages 1–7. IEEE, 2021.

Say Leng Goh, Graham Kendall, and Nasser R Sabar. Improved local search approaches to solve the post enrolment course timetabling problem. *European Journal of Operational Research*, 261(1):17–29, 2017.

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

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

Mehmet Güray Güler, Ebru Geçici, Tuğçe Köroğlu, and Emre Becit. A web-based decision support system for examination timetabling. *Expert Systems with Applications*, 183:1–11, 2021. ISSN 0957-4174.

Dennis S Holm, Rasmus Ø Mikkelsen, Matias Sørensen, and Thomas R Stidsen. A mip based approach for international timetabling competation 2019. In *Proceedings of the International Timetabling Competition 2019*, 2019.

D. S. Johnson. A theoretician's guide to the experimental analysis of algorithms. In M. H. Goldwasser, D. S. Johnson, and C. C. McGeoch, editors, *Data Structures, Near Neighbor Searches, and Methodology: Fifth and Sixth DIMACS Implementation Challenges*, pages 215–250. American Mathematical Society, 2002.

Ahmed Kheiri and Ed Keedwell. A hidden markov model approach to the problem of heuristic selection in hyper-heuristics with a case study in high school timetabling problems. *Evolutionary computation*, 25(3):473–501, 2017.

Alexander Kiefer, Richard F Hartl, and Alexander Schnell. Adaptive large neighborhood search for the curriculum-based course timetabling problem. *Annals of Operations Research*, 252(2):255–282, 2017.

Jeffrey H. Kingston. Educational timetabling. In A. Şima. Etaner-Uyar, Ender Özcan, and Neil Urquhart, editors, *Automated Scheduling and Planning*, volume 505 of *Studies in Computational Intelligence*, pages 91–108. Springer Berlin Heidelberg, 2013.

Philipp Kostuch. The university course timetabling problem with a three-phase approach. In Edmund Burke and Michael Trick, editors, *Proceedings of the 5th international conference on the practice and theory of automated timetabling (PATAT- 2004), selected papers*, volume 3616 of *Lecture notes in computer science*, pages 109–125. Springer-Verlag, 2005.

Simon Kristiansen, Matias Sørensen, and Thomas R Stidsen. Integer programming for the generalized high school timetabling problem. *Journal of Scheduling*, 18(4):377–392, 2015.

Nuno Leite, Carlos Fernandes, Fernando Melício, and Agostinho Rosa. A cellular memetic algorithm for the examination timetabling problem. *Computers and Operations Research*, 94:118–138, 2018.

Nuno Leite, Fernando Melício, and Agostinho Rosa. A fast simulated annealing algorithm for the examination timetabling problem. *Expert Systems with Applications*, 122:137–151, 2019.

Nuno Leite, Fernando Melício, and Agostinho C Rosa. A fast threshold acceptance algorithm for the examination timetabling problem. In *Handbook of Operations Research and Management Science in Higher Education*, pages 323–363. Springer, 2021.

Alexandre Lemos, Francisco S Melo, Pedro T Monteiro, and Inês Lynce. Room usage optimization in timetabling: A case study at Universidade de Lisboa. *Operations Research Perspectives*, 6:100092, 2019.

R. Lewis and B. Paechter. Finding feasible timetables using group-based operators. *IEEE Transactions on Evolutionary Computation*, 11(3):397–413, 2007.

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*, 240(3):637–648, 2015. ISSN 0377-2217.

Rhyd Lewis, Ben Paechter, and Barry McCollum. Post enrolment based course timetabling: A description of the problem model used for track two of the second international timetabling competition. Technical report, Cardiff University, Wales, UK, 2007.

Rhydian Lewis. A survey of metaheuristic-based techniques for university timetabling problems. *OR Spectrum*, 30 (1):167–190, 2008.

M. Lindahl, M. Sørensen, and T.R. Stidsen. A fix-and-optimize matheuristic for university timetabling. *Journal of Heuristics*, 24(4):645–665, 2018.

Leo Lopes and Kate Smith-Miles. Pitfalls in instance generation for Udine timetabling. In *Learning and Intelligent Optimization (LION4)*, pages 299–302. Springer, 2010.

Leo Lopes and Kate Smith-Miles. Generating applicable synthetic instances for branch problems. *Operations Research*, 61(3):563–577, 2013.

Zhipeng Lü and Jin-Kao Hao. Adaptive tabu search for course timetabling. *European Journal of Operational Research*, 200(1):235 – 244, 2009. ISSN 0377-2217.

Ashis Kumar Mandal, Mohd Nizam Mohmad Kahar, and Graham Kendall. Addressing examination timetabling problem using a partial exams approach in constructive and improvement. *Computation*, 8(2):46, 2020.

Alfred Mayer, Clemens Nothegger, Andreas Chwatal, and Günther Raidl. Solving the post enrolment course timetabling problem by ant colony optimization. In E. Burke and M. Gendreau, editors, *Proceedings of the 7th international conference on the practice and theory of automated timetabling (PATAT-2008)*, pages 1–13, 2008.

Barry McCollum, Paul McMullan, Edmund K. Burke, Andrew J. Parkes, and Rong Qu. The second international timetabling competition: Examination timetabling track. Technical Report QUB/IEEE/Tech/ITC2007/Exam/v4.0/17, Queen's University, Belfast (UK), September 2007.

Barry McCollum, Andrea Schaerf, Ben Paechter, Paul McMullan, Rhyd Lewis, Andrew J. Parkes, Luca Di Gaspero, Rong Qu, and Edmund K. Burke. Setting the research agenda in automated timetabling: The second international timetabling competition. *INFORMS Journal on Computing*, 22(1):120–130, 2010.

Amnon Meisels and Andrea Schaerf. Modelling and solving employee timetabling problems. *Annals of Mathematics and Artificial Intelligence*, 39(1-2):41–59, 2003.

Seyyed Ali MirHassani and Farhang Habibi. Solution approaches to the course timetabling problem. *Artificial Intelligence Review*, 39(2):133–149, 2013.

Moritz Mühlenthaler and Rolf Wanka. Fairness in academic course timetabling. *Annals of Operations Research*, 239 (1):171–188, 2016.

Tomáš Müller. Real-life examination timetabling. *Journal of Scheduling*, 19(3):257–270, 2016.

Tomáš Müller. Itc 2019: Preliminary results using the unitime solver. In *Proceedings of the 13th International Conference on the Practice and Theory of Automated Timetabling (PATAT) Volume*, 2020.

Tomáš Müller and Keith Murray. Comprehensive approach to student sectioning. *Annals of Operations Research*, 181 (1):249–269, 2010.

Tomáš Müller. ITC2007 solver description: A hybrid approach. In E. Burke and M. Gendreau, editors, *"Proc. of the 7th Int. Conf. on the Practice and Theory of Automated Timetabling (PATAT-2008)"*, pages 429–446, 2008.

Tomáš Müller, Hana Rudová, and Zuzana Müllerová. University course timetabling and International Timetabling Competition 2019. In *Proceedings of the 12th International Conference on the Practice and Theory of Automated Timetabling (PATAT-2018)*, pages 5–31, 2018.

Yuichi Nagata. Random partial neighborhood search for the post-enrollment course timetabling problem. *Computers and Operations Research*, 90:84–96, 2018. ISSN 0305-0548.

Andrew J Parkes and Ender Özcan. Properties of Yeditepe examination timetabling benchmark instances. In *Proceedings of the 8th International Conference on the Practice and Theory of Automated Timetabling*, pages 531–534, 2010.

Antony Phillips. *Mathematical Programming-based Models and Methods for University Course Timetabling*. PhD thesis, University of Auckland, 2015.

Nelishia Pillay. A survey of school timetabling research. *Annals of Operations Research*, 218(1):261–293, 2014.

Gerhard Post, Samad Ahmadi, Sophia Daskalaki, Jeffrey H Kingston, Jari Kyngas, Cimmo Nurmi, and David Ranson. An xml format for benchmarks in high school timetabling. *Annals of Operations Research*, 194(1):385–397, 2012.

Gerhard Post, Luca Di Gaspero, Jeffrey H. Kingston, Barry McCollum, and Andrea Schaerf. The third international timetabling competition. *Annals of Operations Research*, 239(1):69–75, 2016. doi:10.1007/s10479-013-1340-5.

R. Qu, E. Burke, B. McCollum, L. Merlot, and S.Y. Lee. A survey of search methodologies and automated system development for examination timetabling. *Journal of Scheduling*, 12(1):55–89, 2009.

E Rappos, E Thiémard, S Robert, and JF Hecĥe. A mip based approach for international timetabling competation 2019. In *Proceedings of the International Timetabling Competition 2019*, 2019.

Olivia Rossi-Doria, Michael Sampels, Mauro Birattari, Marco Chiarandini, Marco Dorigo, Luca M. Gambardella, Joshua Knowles, Max Manfrin, Monaldo Mastrolilli, Ben Paechter, Luis Paquete, and Thomas Stützle. A comparison of the performance of different metaherustic on the timetabling problem. In Edmund Burke and Patrick De Causmaecker, editors, *Proc. of the 4th Int. Conf. on the Practice and Theory of Automated Timetabling (PATAT-2002), selected papers*, volume 2740 of *Lecture Notes in Computer Science*, pages 329–351, Berlin-Heidelberg, 2003. Springer-Verlag.

Hana Rudová, Tomáš Müller, and Keith Murray. Complex university course timetabling. *Journal of Scheduling*, 14 (2):187–207, 2011.

Landir Saviniec and Ademir Aparecido Constantino. Effective local search algorithms for high school timetabling problems. *Applied Soft Computing*, 60:363–373, 2017.

Andrea Schaerf. A survey of automated timetabling. *Artificial Intelligence Review*, 13(2):87–127, 1999.

J Dario Landa Silva, Edmund K Burke, and Sanja Petrovic. An introduction to multiobjective metaheuristics for scheduling and timetabling. In *Metaheuristics for multiobjective optimisation*, pages 91–129. Springer, 2004.

T. Song, M. Chen, Y. Xu, D. Wang, X. Song, and X. Tang. Competition-guided multi-neighborhood local search algorithm for the university course timetabling problem. *Applied Soft Computing*, 110, 2021.

Thomas Stidsen, David Pisinger, and Daniele Vigo. Scheduling EURO-k conferences. *European Journal of Operational Research*, 270(3):1138–1147, 2018.

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

Ulisses Rezende Teixeira, Marcone Jamilson Freitas Souza, Sérgio Ricardo de Souza, and Vitor Nazário Coelho. An adaptive vns and skewed gvns approaches for school timetabling problems. In *International Conference on Variable Neighborhood Search*, pages 101–113. Springer, 2018.

David Van Bulck, Dries Goossens, Jörn Schönberger, and Mario Guajardo. Robinx: A three-field classification and unified data format for round-robin sports timetabling. *European Journal of Operational Research*, 280(2):568–580, 2020.

David Van Bulck, Dries Goossens, Jeroen Belien, and Morteza Davari. The fifth international timetabling competition (itc 2021): Sports timetabling. In *MathSport International 2021*, pages 117–122. University of Reading, 2021.

Gert Woumans, Liesje De Boeck, Jeroen Beliën, and Stefan Creemers. A column generation approach for solving the examination-timetabling problem. *European Journal of Operational Research*, 253(1):178–194, 2016.

Ender Özcan and Ersan Ersoy. Final exam scheduler-fes. In *2005 IEEE Congress on Evolutionary Computation*, volume 2, pages 1356–1363. IEEE, 2005.