MISTA 2015

Modelling and Solving University Course Timetabling Problems Through XHSTT

George H.G. Fonseca · Haroldo G. Santos · Eduardo G. Carrano · Thomas J.R. Stidsen

Abstract The XHSTT was proposed as a standard format to express a wide range of School Timetabling problems. Although the format is powerful to represent different timetabling features, its application to University Course Timetabling (UCT) problems was not formally studied. The goal of this work is to present encodings from Curriculum-Based Course Timetabling (CB-CTT) and Post-Enrolment Course Timetabling (PE-CTT) to XHSTT and to evaluate how a state-of-art XHSTT solver performs on these problems. Computational experiments performed suggested that this approach is suitable for dealing with UTC: the XHSTT solver would be ranked as fourth in CB-CTT track of the Second International Timetabling Competition (ITC2007) and it achieved feasible solutions for most PE-CTT instances within one hour. Although the results do not outperform the best known approaches for these problems, XHSTT solvers were designed to handle a wide range of features and constraints beyond the

George H. G. Fonseca
Department of Computing and Information Systems
Federal University of Ouro Preto
St. Diogo de Vasconcelos 328, 35400-000
Ouro Preto, MG, Brazil
E-mail: george@decsi.ufop.br

Haroldo G. Santos
Department of Computing
Federal University of Ouro Preto
St. Diogo de Vasconcelos 328, 35400-000
Ouro Preto, MG, Brazil
E-mail: haroldo@iceb.ufop.br

Eduardo G. Carrano and George H. G. Fonseca
Graduate Program in Electrical Engineering
Federal University of Minas Gerais
Av. Antônio Carlos 6627, 31270-901
Belo Horizonte, MG, Brazil
E-mail: egcarrano@ufmg.br

Thomas R. J. Stidsen and George H. G. Fonseca
Department of Management Engineering
Technical University of Denmark
Building 426, Produktionstorvet, DK-2800
Kgs. Lyngby, Denmark
E-mail: thst@dtu.dk

ones present in these models, making it able to fit the specific requirements of several universities.

Keywords University Course Timetabling · Curriculum-Based Course Timetabling · Post-Enrolment Course Timetabling · XHSTT

1 Introduction

The University Course Timetabling (UCT) problem consists on scheduling a set of lectures for each course within a given number of rooms and timeslots. These assignments must comply several constraints, for example, (1) two lectures that the same student attends should not occur at the same time, (2) the room assigned to a lecture should be large enough to accommodate all enrolled students, and (3) unavailabilities of teachers should be respected.

The requirements of this problem vary largely from one university to another, specially when they are from different countries. These differences make it hard to perform fair comparisons of solvers for this problem. Considering this fact, two models of this problem were proposed in the Second International Timetabling Competition (ITC2007): (1) Post-Enrolment Course Timetabling and (2) Curriculum-Based Course Timetabling [19]. Although these models capture the main features of the problem, some university specific features cannot be modelled in these formats [9] [16].

The XHSTT (eXtended Markup Language for High School TimeTabling) format was proposed by Post *et al.* [24] aiming to be a standard format suitable for features from several High School Timetabling problems around the world. Due to its generality, it was adopted as the standard format for the Third International Timetabling Competition (ITC2011) [23].

Although the XHSTT format is powerful for expressing different features of timetabling problems, to the best of the authors' knowledge, it was never studied whether this format can be used to express and solve other timetabling problems besides High School Timetabling. Thus the main goal of this work is to investigate whether the XHSTT format is able to describe the University Course Timetabling problems from ITC2007 and how a state-of-art XHSTT solver performs at solving these problems.

The outline of this paper is as follows. Section 2 presents the description of the University Course Timetabling problems that were addressed in this work. Section 3 presents a short description of XHSTT format. Section 4 presents how the UCT problems can be encoded using XHSTT and the solver employed. Section 5 presents the computational experiments and the results. Finally, in Section 6, some concluding remarks are presented.

2 University Course Timetabling

Two models of the University Course Timetabling problem were considered in this work: Curriculum-Based Course Timetabling (CB-CTT) and Post-Enrolment Course Timetabling (PE-CTT). Both models were proposed in the ITC2007 and they are widely addressed in literature [6, 18, 2]. CB-CTT consists on the weekly scheduling of the lectures for several university courses within a given number of rooms and time periods. The conflicts between courses are set according to the curricula published by the University. PE-CTT assumes that the students have already enrolled the courses they want to follow and the timetabling should be built in such a way that there is no conflict for any student. In both cases, several other secondary objectives are considered as well. An extended explanation of each of these problems is given in the following subsections. Integer Programming formulations for CB-CTT and for PR-CTT can be found in [4] and [3], respectively.

2.1 Curriculum-Based Course Timetabling

The input of a Curriculum-Based Course Timetabling problem instance is given below [9]:

- A set of teaching days in a week $d \in D$ and a set of timeslots per day T_d. A period is a pair composed of a day and a timeslot.
- A set of rooms $r \in R$. Each room has a capacity $c_r \in \mathbb{N}$. Except for capacity, all rooms are equally suitable for all courses.
- A set of courses $c \in C$. Each course has a number of lectures (duration) $d_c \in \mathbb{N}$ to be scheduled. A course is attended by a number of students s_c and taught by a teacher. The lessons shout be spread into a minimum number $min_c \in \mathbb{N}$ of days and they should respect teacher unavailability.
- A set of curricula $q \in Q$. A curricula is a group of courses such that any pair of courses in the group have students in common. The set of courses of a curriculum is given by C_q.

The solution of the problem is an assignment of periods and rooms for all lectures of each course. This solution should satisfy several constraints, classified as hard or soft constraints. On the one hand, the compliance with hard constraints is mandatory, which means that a solution is said to be infeasible if hard constraint is violated. On the other hand, the attendance of soft constrains is desirable, but it is not mandatory. The penalties on soft constraints measure how nice a feasible scheduling is for its users.

The hard constraints of CB-CTT are given below:

H1: Two lectures cannot take place in the same room at the same time period.
H2: Lectures of courses in the same curriculum or taught by the same teacher must be scheduled in different periods.
H3: A lecture should not be scheduled to a period in which its teacher is not available.

The set of soft constraints is as follows:

S1: For each lecture, the number of students that attend the course should be less or equal to the number of seats of the assigned room ($s_c \leq c_r$). Each student exceeding this capacity increases total penalty in 1 point.
S2: The lectures of each course must be spread into at least min_c days. Each day below the minimum counts as 5 points for penalty.
S3: Lectures belonging to a curriculum should be adjacent to each other. Each isolated lecture in a curriculum counts as 2 penalty points.
S4: All lectures of a course should be given in the same room. Each distinct room used (except for the first one) counts as 1 point of penalty.

2.2 Post-Enrolment Course Timetabling

Usually, the goal of the Post-Enrolment Course Timetabling problem is to assign, for each event $e \in E$, a timeslot $t \in T$ and a room $r \in R$ respecting several a priori given constraints. The set of timeslots is composed by 45 times $T = \{1, \ldots, 45\}$ (grouped into 5 days of 9 hours each). Every room has a set of features F_r and a seating capacity $c_r \in \mathbb{N}$. Every event has a set of preferred times T_e, a set of required room features F_e and a number of enrolled students $s_e \in \mathbb{N}$. An event e can be assigned to a room r only if r satisfies the required features from e ($F_r \subseteq F_e$) and the number of seats in room r is sufficient for the number of students that attends e ($s_e \leq c_r$). The set of suitable rooms for each event e is R_e. There is also a set of students $s \in S$ having each a set of events that he/she attends S_e. Finally, a relation of precedences between events $r_{e_1,e_2} \in \{0, 1\}$ has the value 1 if event e_1 should be a predecessor of event e_2 [16].

Similarly to the CB-CTT problem, the constraints are split into hard constraints, whose compliance is mandatory, and soft constraints, whose compliance is desirable. The set of hard constraints is given below:

H1: No student can attend more than one event at a time.
H2: An event e can only be assigned to a suitable room $r \in Re$.
H3: No more than one event can be assigned to a room at any time slot.
H4: An event e can only be assigned to a preferred time slot $t \in T_e$.
H5: Events have to be scheduled in the prescribed order during the week.

The set of soft constraints is given below:

S1: Students should not be scheduled to attend an event in the last timeslot of a day (i.e. timeslots 9, 18, 27, 36, or 45). Each assignment of a student to attend an event in the last time of the say counts as 1 point for penalty.
S2: Students should not have to attend three (or more) events in successive timeslots of the same day. Each occurrence of a student having more than two classes consecutively in the same day increases the total penalty in 1 point.
S3: Students should not be required to attend only one event in a particular day. Each time a student has to attend only one event in a day counts as 1 point for penalty.

3 XHSTT Format

A XHSTT instance is composed of four entities:

- **Times**: it contains the possible timeslots for allocations. These timeslots may also be grouped in *TimeGroups*.
- **Resources**: it contains the available resources for assignments. Each resource has a specific *ResourceType*. Resources are also commonly grouped in *ResourceGroups*.
- **Events**: it represents the events to be scheduled. Each event has a duration, which represents he amount of times should be scheduled and a demand for a set of resources. Optionally events may have a workload which is considered to its assigned resources. Events are also commonly grouped into *EventGroups*.
- **Constraints**: it represents the set of constraints that should be attended. Table 1 presents the 16 constraint types available in this format. Each constraint may

Constraint	Description
Assign Resource	Event resource should be assigned a resource
Assign Time	Event should be assigned a time
Split Events	Event should split into a constrained number of sub-events
Distribute Split Events	Event should split into sub-events of constrained durations
Prefer Resources	Event resource assignment should come from resource group
Prefer Times	Event time assignment should come from time group
Avoid Split Assignments	Set of event resources should be assigned the same resource
Spread Events	Set of events should be spread evenly through the cycle
Link Events	Set of events should be assigned the same time
Order Events	Set of events should be ordered
Avoid Clashes	Resource's timetable should not have clashes
Avoid Unavailable Times	Resource should not be busy at unavailable times
Limit Idle Times	Resource's timetable should not have idle times
Cluster Busy Times	Resource should be busy on a limited number of days
Limit Busy Times	Resource should be busy a limited number of times each day
Limit Workload	Resource's total workload should be limited

Table 1 Different constraint types in the XHSTT format [22].

be hard, soft or absent. Hard constraints estimate feasibility, while soft constraints measure the quality of a solution (smaller values indicate better solutions). Each constraint has a cost that indicates the penalty for a single violation and a cost function, which defines how violations are penalized in the objective function. A deeper description of this format can be found in [24] and [14].

Eventually, times and resources may be preassigned to events. When they are not preassigned and Assign Times and/or Assign Resources constraints are present, a solver for XHSTT must do this assignment. It is also the job of XHSTT solvers to decide how (or if) events will be split into smaller sub-events.

4 Solution Approach

The solution approach for these problems is composed by two steps: (1) translate the problem to a XHSTT analogous and; (2) solve the resulting XHSST instance by a state-of-art solver. The translation from CB-CTT and PE-CTT to XHSTT and the solver applied to XHSTT instances are described in the next subsections.

4.1 CB-CTT Encoding

For each timeslot the corresponding $Time$ entity is created in XHSTT. For each curriculum, for each course, for each teacher, and for each room a *Resource* entity is created in XHSTT. Each course is directly converted to an *Event* having duration equals to the number of required lectures d_c. Each constraint from CB-CTT is encoded to XHSTT as follows:

H0: Assign Time Constraint and Assign Resource Constraint apply to all events, in order to ensure that a time and a room are assigned to each event. Weight: 1.
H1: Avoid Clashes Constraint applies to all rooms. Weight: 1.
H2: Avoid Clashes Constraint applies to all curricula and to all teachers. Weight: 1.

H3: Prefer Times Constraint applied to all courses, in order to ensure that all times except the unavailable ones are preferable. Weight: 1.
S1: Prefer Resources Constraint applies to each event e and to each room r in which the room capacity is lower than the event demand . Weight: $s_c - c_r$.
S2: Cluster Busy Times Constraint applies to each course c, stating that the minimum number of busy days is min_c. Weight: 5.
S3: Limit Busy Times Constraint applies to all students to ensure that a curricula should have at least 2 busy times in each day (this constraint does not applies to days without assignments), and; Limit Idle Times applies to all curricula stating that idle times are not desired in its schedule. Weight: 2.
S4: Avoid Split Assignments Constraint applies to all courses, stating that all events of a given class should be assigned to the same room. Weight: 1.

It is important to highlight that the encoding of constraint S3 for XHSTT does not fit the original constraint perfectly. The adapted constraint does not allow any idle time in curricula timetables whereas the original constraint states that isolated lectures are not desired. Unfortunately, it is possible to have an idle time without any isolated lecture (e.g. Times 1, 2, 4 and 5 active). The authors could not find a better fit for this constraint in XHSTT. Although this constraint does not model the original one exactly, it is suitable for real world problems.

4.2 PE-CTT Encoding

For each timeslot the corresponding *Time* entity is created in XHSTT. Each combination of three times in a row in the same day is converted to a *TimeGroup*, which contains the involved times. For each room and for each student a *Resource* entity is created in XHSTT. Each event is directly converted to an *Event* having duration 1 in XHSTT. Each constraint from PE-CTT is encoded to XHSTT as follows:

H0: Assign Time Constraint and Assign Resource Constraint apply to all events, in order to ensure that a time and a room are assigned to each event. Weight: 1.
H1: Avoid Clashes Constraint applies to all students. Weight: 1.
H2: Prefer Resources Constraint applies for each event, stating that only suitable rooms are preferred. Weight: 1.
H3: Avoid Clashes Constraint applies to all rooms. Weight: 1.
H4: Prefer Times Constraint applied to all events, stating that all the times except the unavailable ones are preferable. Weight: 1.
H5: Order Events Constraint applied to each pair of events that has to be ordered. Weight: 1.
S1: Avoid Unavailable Times Constraint applies to all students, in order to ensure that timeslots 9, 18, 27, 36 and 45 are unavailable. Weight: 1.
S2: Limit Busy Times Constraint applies to all students, stating that those students can have at most 2 assignments in any of the *TimeGroups* that represents three times in a row. Weight: 1.
S3: Limit Busy Times Constraint applies to all students, stating that students should have at least 2 busy times in each day (this constraint does not apply to days without assignments). Weight: 1.

This encoding fits perfectly the original PE-CTT problem.

4.3 XHSTT Solver

The adopted XHSTT solver employees a three step approach that involves graph algorithms, a metaheuristic and a matheuristic. This solver is an improved version of the ITC2011 winner [10] and it is refereed to as Hybrid Solver (HS). More detail about the solver can be found in [11].

In the first step, an initial solution is built by the KHE solver [13]. This solver, initially splits the events into sub-events according to Split Events and Distribute Split Events constraints and it makes connections between events connected by Link Events and Avoid Split Assignments constraints. In the next phase, it assigns times to these sub-events. The time assignment is made by solving weighted minimum matching problems between the times a resource is available and the events that this resource is preassigned to. Finally, in the resource assignment phase, a simple heuristic assigns resources, from the most constrained event to the least constrained one, prioritizing the assignments that leads to small (ideally 0) number of violations of constraints.

On the second step, a Variable Neighborhood Search (VNS) based algorithm is applied to the initial solution. The VNS algorithm has six neighborhood moves:

1. Event Swap (ES): two events e_1 and e_2 have their timeslots t_1 and t_2 swapped.
2. Event Move (EM): an event e_1 is moved from timeslot t_1 to another timeslot t_2.
3. Event Block Move (EBM): it works just like ES, but when moving events with different durations in contiguous timeslots, it keeps these events adjacent.
4. Resource Swap (RS): two events e_1 and e_2 have their assigned resources r_1 and r_2 swapped. Resources r_1 and r_2 should play the same role to allow the swap (e.g. both have to be teachers).
5. Resource Move (RM): an event e_1 has its assigned resource r_1 replaced by a new resource r_2.
6. Kempe Move (KM): two times t_1 and t_2 are fixed and one looks for the best path in ta bipartite conflict graph containing all events in t_1 and t_2. Arcs are built from conflicting events which are in different timeslots and their cost is the cost of swapping the timeslots of these two events.

The matheuristic is invoked when VNS reaches 10% of all available run time without get any improvement. The matheuristic loads the VNS solution to an Integer Programming model [15] and iteratively builds IP sub-problems having variables from a random set of resources freed to be optimized and the others fixed to their current value. The size of the sub-problem is auto-adaptive and the run time for each iteration (sub-problem) is 5% of the time limit set up to the solver. This algorithm runs until the time limit be reached and it returns the best solution found.

5 Computational Experiments

All experiments were ran on an Intel® i7 4510U 2.6 Ghz computer with 8GB of RAM portable computer, under Ubuntu 14.04 operating system. The software was coded in C++ and it was compiled with GCC 4.6.1. The Integer Programming solver adopted for the matheuristic phase was Gurobi 6.5. The obtained results were validated by both

XHSTT validator (HSEval[1]) and ITC2007 validators[2]. Our solver, along with our new instances, solutions and reports, can be found at GOAL-UFOP website[3].

5.1 Curriculum-Based Course Timetabling Results

Table 2 presents the obtained results of the Hybrid Solver for XHSTT applied to the Curriculum-Based Course Timetabling problem. For comparison purposes, the results from the finalists of ITC2007 and the best known solution ($\mathcal{UB}$) were also added to the table. The presented results are the average of ten executions of each solver. When the best known solution is optimal, it is marked with an asterisk (*). All solutions presented in this table are feasible, thus only the soft cost is presented. The time limit for the Hybrid Solver was adjusted to 273 seconds, according to the benchmark provided by the organizers of ITC2007[4].

Instances	$\mathcal{UB}$	Muller [20]	Lu [17]	Atsuta [1]	Geiger [12]	Clark [8]	HS [11]
comp01	5.00*	5.00	5.00	5.10	6.70	27.00	5.00
comp02	24.00	61.30	61.20	65.60	142.70	131.10	97.40
comp03	65.00	94.80	84.50	89.10	160.30	138.40	118.20
comp04	35.00*	42.80	46.90	39.20	82.00	90.20	50.00
comp05	284.00	343.50	326.00	334.50	525.40	811.50	1288.20
comp06	27.00*	56.80	69.40	74.10	110.80	149.30	108.20
comp07	6.00*	33.90	41.50	49.80	76.60	153.4	114.00
comp08	37.00*	46.50	52.60	46.00	81.70	96.50	63.80
comp09	96.00*	113.10	116.50	113.30	164.10	148.90	133.60
comp10	4.00*	21.30	34.80	36.90	81.30	101.30	89.60
comp11	0.00*	0.00	0.00	0.00	0.30	5.70	0.00
comp12	294.00	351.60	360.10	361.60	485.10	445.30	460.20
comp13	59.00*	73.90	79.20	76.10	110.40	122.90	109.00
comp14	51.00*	61.80	65.90	62.30	99.00	105.90	84.20
comp15	62.00	94.80	84.50	89.10	160.30	138.00	110.00
comp16	18.00*	41.20	49.10	50.20	92.60	107.30	69.00
comp17	56.00*	86.60	100.70	107.30	143.40	166.60	135.20
comp18	61.00*	91.70	80.70	73.30	129.40	126.80	107.00
comp19	57.00*	68.80	69.50	79.60	132.80	125.40	73.80
comp20	4.00*	34.30	60.90	65.00	97.50	179.30	276.00
comp21	74.00*	108.00	124.70	138.10	185.30	185.80	139.60
Avg. Rank		1.64	2.07	2.55	5.10	5.57	4.07
Avg. Gap		34.77	37.23	37.18	63.50	67.64	51.08

Table 2 Results of Hybrid Solver for Curriculum-Based Course Timetabling

It can be noticed from Table 2 that the Hybrid Solver for XHSTT is competitive on solving Curriculum-Based Course Timetabling problems from ITC2007. Considering the ordering procedure from the competition, the solver would be ranked fourth between the finalists. Although the obtained results were not better than the results from the winner, this performance shows that the Hybrid Solver is suitable for solving real world instances of CB-CTT. On average, our solver got solutions whose gap to the winner is around 15%. This result was expected since this solver was not designed to this specific problem.

[1] `http://sydney.edu.au/engineering/it/~jeff/hseval.cgi`

[2] `http://www.cs.qub.ac.uk/itc2007/curriculmcourse/validator.cc` and `http://www.cs.qub.ac.uk/itc2007/postenrolcourse/checksln3b.cpp`

[3] `http://www.goal.ufop.br/softwares/hstt`

[4] `http://www.cs.qub.ac.uk/itc2007/index_files/benchmarking.htm`

It is important to highlight that the XHSTT solver has an advantage when compared to solvers specifically designed to CB-CTT in real world problems: in XHSTT it is easy to add and to remove features and constraints according to the needs of each university. These adjustments for the specific needs of each university do not require any additional effort on coding new features and it is expected that the solver still works well on the modified problem. One example of feature that could be covered by XHSTT and its solvers is the requirement of double lessons (i.e. two contiguous assignments of the same course).

5.2 Post-Enrolment Course Timetabling Results

Table 3 presents the results for the Hybrid Solver applied to the Post-Enrolment Course Timetabling problems from ITC2007. When the obtained solution is not feasible, the results are expressed by the pair (H, S), in which H is the total penalty for hard constraints violation and S if the sum for the soft constraints. When the solutions have no hard constraint violation, only the penalty of soft constraints is presented. It is important to highlight that the Hybrid Solver took, at least 300 seconds to generate initial solutions. Thus, the time limit was extended to one hour (3600 seconds) for these experiments. Due to the long time of each run, only one execution was considered. Similarly to Table 2, the average results for 10 executions from the finalists of ITC2007 were also presented in the table. However, they should not be interpreted as a comparison but rather as an overview of how HS performs for this problem (taking longer processing times).

Although the Hybrid Solver was not able to compete with the finalists of ITC2007, the Hybrid Solver could produce feasible solutions for most of the instances from ITC2007. In fact, for some instances, the produced solutions were better than some of the finalists. In general the results show that the Hybrid Solver is not competitive with the best known approaches for this specific problem in University Course Timetabling but it still can be applied and achieve good solutions for most of the instances. Finally, it can easily accommodate new constraints or features from specific needs that vary from university to university.

6 Concluding Remarks

This work presented encodings from the two most studied models of University Course Timetabling to XHSTT and the results of XHSTT solvers on the resulting instances.

Computational experiments demonstrated that the Curriculum-Based Course Timetabling problem could be solved by a state of art XHSTT solver. The solver would be ranked as fourth on this track of ITC2007. In addition, it was also able to solve the Post-Enrolment Course Timetabling problem, however it took a considerably larger time to find good solutions when compared to the ITC2007 finalists.

In a practical perspective, the XHSTT solver is suitable for real world problems of UCT, since it is flexible enough for modelling a wide range of features according to university specific needs.

It is suggested as subject of further research: (1) to encode other timetabling problems into the XHSTT format and to evaluate the performance of XHSTT solvers on

Table 3: Results of Hybrid Solver for Post-Enrolment Course Timetabling

Instance	Atsuta [1]	Cambazard [5]	Chiarandini [7]	Nothegger [21]	Muller [20]	HS[a] [11]
comp-2007-2-1	(8.3, 647.6)	883.4	1730.5	(445.5, 1071.2)	(22.2, 1927)	(2.0, 1801.0)
comp-2007-2-2	(30.4, 884.6)	1565.9	1913.6	(335.8, 677.9)	(171.9, 2201.8)	(22.0, 2544.0)
comp-2007-2-3	529.9	237.3	389.7	732.6	333.9	694.0
comp-2007-2-4	683.2	370.0	480.2	727.5	559.7	915.0
comp-2007-2-5	21.0	6.8	679.9	128.3	20.9	797.0
comp-2007-2-6	64.5	4.2	977.4	391.9	266.6	865.0
comp-2007-2-7	97.7	7.5	354.1	3.8	183.6	(1.0, 1418.0)
comp-2007-2-8	24.1	0.0	1.3	80.6	24.5	430.0
comp-2007-2-9	(79.9, 832.9)	1868.6	2100.4	(684.3, 1080.5)	(346.5, 2407.7)	(4.0, 2432.0)
comp-2007-2-10	(8.1, 231.7)	1850.0	(37.1, 2272.3)	0.1	(577.5, 2319.0)	(20.0, 2581.0)
comp-2007-2-11	716.0	288.3	352.6	898.0	(5.4, 742.9)	832.0
comp-2007-2-12	1046.8	352.7	616.4	(112.8, 1275.3)	(14.1, 1293.1)	1079.0
comp-2007-2-13	102.6	128.3	911.1	478.6	475.6	945.0
comp-2007-2-14	0.4	4.1	983.5	97.0	407.9	889.0
comp-2007-2-15	460.6	93.1	310.6	142.7	268.2	624.0
comp-2007-2-16	251.8	17.1	5.8	131.2	178.4	390.0
comp-2007-2-17	19.0	4.9	9.8	116.4	106.2	102.0
comp-2007-2-18	(0.6, 39.4)	14.1	339.9	264.8	314.3	628.0
comp-2007-2-19	(348.7, 1838.2)	2027.0	2080.8	(89.7, 233.1)	(755.1, 2314)	(26.0, 2287.0)
comp-2007-2-20	(16.4, 1287.5)	505.0	640.5	(771.2, 2382.8)	919.3	879
comp-2007-2-21	3.6	27.1	876.3	(14.3, 326.6)	336.8	631
comp-2007-2-22	0.0	612.0	1839.2	82.7	(36.4, 1593.7)	(119, 2431)
comp-2007-2-23	(3.4, 573.9)	330.5	1043.4	(239.8, 1274.1)	701.3	(2, 3971)
comp-2007-2-24	(13.1, 911.1)	124.2	963.4	129.2	518.0	(2, 1591)

[a] Considering a larger time limit of 3600 seconds.

them, and; (2) to develop new neighborhood moves to the XHSTT solver so that it could exploit some features of these problems.

Acknowledgements This work was done with the support of the National Council for Scientific and Technological Development - CNPq. The authors would like also to thank the Brazilian agencies CAPES and FAPEMIG for the financial support.

References

1. Atsuta, M., Nonobe, K., Ibaraki, T.: Itc-2007 track2: an approach using general csp solver (2008)
2. Bettinelli, A., Cacchiani, V., Roberti, R., Toth, P.: An overview of curriculum-based course timetabling. TOP **23**(2), 313–349 (2015). DOI 10.1007/s11750-015-0366-z. URL http://dx.doi.org/10.1007/s11750-015-0366-z
3. Broek, J.J.J., Hurkens, C.A.J.: An ip-based heuristic for the post enrolment course timetabling problem of the itc2007. Annals of Operations Research **194**(1), 439–454 (2010). DOI 10.1007/s10479-010-0708-z. URL http://dx.doi.org/10.1007/s10479-010-0708-z
4. Burke, E.K., Mareek, J., Parkes, A.J., Rudov, H.: Decomposition, reformulation, and diving in university course timetabling. Computers & Operations Research **37**(3), 582 – 597 (2010). DOI http://dx.doi.org/10.1016/j.cor.2009.02.023. URL http://www.sciencedirect.com/science/article/pii/S0305054809000628. Hybrid Metaheuristics
5. Cambazard, H., Hebrard, E., O'Sullivan, B., Papadopoulos, A.: Local search and constraint programming for the post enrolment-based course timetabling problem. Annals of Operations Research **194**(1), 111–135 (2010). DOI 10.1007/s10479-010-0737-7. URL http://dx.doi.org/10.1007/s10479-010-0737-7
6. Ceschia, S., Di Gaspero, L., Schaerf, A.: Design, engineering, and experimental analysis of a simulated annealing approach to the post-enrolment course timetabling problem. Computers & Operations Research **39**(7), 1615–1624 (2012)
7. Chiarandini, M., Fawcett, C., Hoos, H.H.: A modular multiphase heuristic solver for post enrollment course timetabling. In: Proceedings of the 7th international conference on the practice and theory of automated timetabling (PATAT 2008) (2008)
8. Clark, M., Henz, M., Love, B.: Quikfixa repair-based timetable solver. In: Proceedings of the Seventh International Conference on the Practice and Theory of Automated Timetabling, http://www. comp. nus. edu. sg/˜ henz/publications/ps/PATAT2008. pdf (2008)
9. Di Gaspero, L., McCollum, B., Schaerf, A.: The second international timetabling competition (itc-2007): Curriculum-based course timetabling (track 3). Tech. rep., QUB/IEEE/Tech/ITC2007/CurriculumCTT/v1.0, Queens University, Belfast, United Kingdom (2007)
10. Fonseca, G., Santos, H., Toffolo, T., Brito, S., Souza, M.: Goal solver: a hybrid local search based solver for high school timetabling. Annals of Operations Research pp. 1–21 (2014). DOI 10.1007/s10479-014-1685-4. URL http://dx.doi.org/10.1007/s10479-014-1685-4
11. Fonseca, G.H., Santos, H.G.: Integrating metaheuristics and matheuristics for timetabling. Computers & Operations Research (submmited) (2016)
12. Geiger, M.J.: Evolutionary Multi-Criterion Optimization: 5th International Conference, EMO 2009, Nantes, France, April 7-10, 2009. Proceedings, chap. Multi-criteria Curriculum-Based Course Timetabling—A Comparison of a Weighted Sum and a Reference Point Based Approach, pp. 290–304. Springer Berlin Heidelberg, Berlin, Heidelberg (2009). DOI 10.1007/978-3-642-01020-0_25. URL http://dx.doi.org/10.1007/978-3-642-01020-0_25
13. Kingston, J.: Khe14: An algorithm for high school timetabling. In: Proceedings of the tenth international conference on the practice and theory of automated timetabling (PATAT 2014), pp. 269–291 (2014)
14. Kingston, J.H.: A software library for school timetabling (2012). Available at http://sydney.edu.au/engineering/it/~jeff/khe/, Accessed in December / 2012
15. Kristiansen, S., Srensen, M., Stidsen, T.: Integer programming for the generalized high school timetabling problem. Journal of Scheduling pp. 1–16 (2014). DOI 10.1007/s10951-014-0405-x. URL http://dx.doi.org/10.1007/s10951-014-0405-x
16. Lewis, R., Paechter, B., McCollum, B., et al.: Post enrolment based course timetabling: A description of the problem model used for track two of the second international timetabling competition. Cardiff Business School (2007)
17. L, Z., Hao, J.K.: A critical element-guided perturbation strategy for iterated local search. In: C. Cotta, P.I. Cowling (eds.) EvoCOP, *Lecture Notes in Computer Science*, vol. 5482, pp. 1–12. Springer (2009). URL http://dblp.uni-trier.de/db/conf/evoW/evocop2009.html#LuH09
18. Mansour, N., El-Jazzar, H.: Curriculum based course timetabling. In: Natural Computation (ICNC), 2013 Ninth International Conference on, pp. 787–792. IEEE (2013)
19. McCollum, B., Schaerf, A., Paechter, B., McMullan, P., Lewis, R., Parkes, A.J., Gaspero, L.D., Qu, R., Burke, E.K.: Setting the research agenda in automated timetabling: The

second international timetabling competition. INFORMS Journal on Computing **22**(1), 120–130 (2010)
20. Muller, T.: ITC2007 solver description: a hybrid approach. Annals OR **172**(1), 429–446 (2009). URL http://dblp.uni-trier.de/db/journals/anor/anor172.html#Muller09
21. Nothegger, C., Mayer, A., Chwatal, A., Raidl, G.R.: Solving the post enrolment course timetabling problem by ant colony optimization. Annals of Operations Research **194**(1), 325–339 (2012). DOI 10.1007/s10479-012-1078-5. URL http://dx.doi.org/10.1007/s10479-012-1078-5
22. Post, G., Ahmadi, S., Daskalaki, S., Kingston, J.H., Kyngas, J., Nurmi, C., Ranson, D.: An xml format for benchmarks in high school timetabling. In: Annals of Operations Research DOI 10.1007/s10479-010-0699-9., pp. 3867 : 267–279 (2010)
23. Post, G., Di Gaspero, L., Kingston, J., McCollum, B., Schaerf, A.: The third international timetabling competition. Annals of Operations Research pp. 1–7 (2013). DOI 10.1007/s10479-013-1340-5. URL http://dx.doi.org/10.1007/s10479-013-1340-5
24. Post, G., Kingston, J., Ahmadi, S., Daskalaki, S., Gogos, C., Kyngas, J., Nurmi, C., Musliu, N., Pillay, N., Santos, H., Schaerf, A.: XHSTT: an XML archive for high school timetabling problems in different countries. Annals of Operations Research p. 17 (2011). URL http://dx.doi.org/10.1007/s1047901110122. 10.1007/s1047901110122