

The Third International Timetabling Competition

Gerhard Post, Luca Di Gaspero,
Jeffrey H. Kingston, Barry McCollum, and
Andrea Schaerf

Received: date / Accepted: date

Abstract This paper is the organizers' report on the Third International Timetabling Competition (ITC2011), run during the first half of 2012. Its participants tackled 35 instances of the high school timetabling problem, taken from schools in 10 countries.

Keywords High school timetabling · International timetabling competition

1 Introduction

High school timetabling is a long-established area of timetabling, but it has received less scientific attention than some other areas, such as university timetabling. Its research community has been fragmented, and very little data has been shared.

Over the last few years, a group of researchers has worked on this problem. There is now an XML data format in which unsimplified instances and solutions of the problem can be specified precisely (Post et al. 2012, 2010b), and a web site for evaluating solutions (Kingston 2010). Instances have been widely sought. At the time of writing, 35 instances from 10 countries are available for download (Post 2011).

Building on this work and on two previous competitions (Paechter et al. 2002; McCollum et al. 2007), and supported financially by the PATAT conference series, EventMAP, and the Centre for Telematics and Information Technology at the University of Twente in the Netherlands, the Third International Timetabling Competition (ITC2011) was run by the authors in the first half of 2012.

This paper is the organizers' report on that competition.

Jeffrey H. Kingston
School of Information Technologies
The University of Sydney
E-mail: jeff@it.usyd.edu.au

2 Modelling high school timetabling

The competition used a data format called XHSTT in which high school timetabling problems and solutions can be expressed. This format has been described previously (Post et al. 2012, 2010b), and a full specification is available online (Kingston 2010). An overview, omitting syntactic details, is given here for completeness.

An XHSTT file is an XML file containing one *archive*, which consists of a set of instances of the high school timetabling problem, plus any number of *solution groups*. A solution group is a set of solutions to some or all of the archive's instances, typically produced by one solver. There may be several solutions to one instance in one solution group, for example solutions produced by the same stochastic solver, but using different random seeds.

Each instance has four parts. The first part defines the instance's *times*, that is, the individual intervals of time, of unknown duration, during which events run. Taken in chronological order these times form a sequence called the instance's *cycle*, which is usually one week. Arbitrary sets of times, called *time groups*, may be defined, such as the Monday times or the afternoon times. A *day* is a time group holding the times of one day, and a *week* is a time group holding the times of one week. For the convenience of display software, some time groups may be labelled as days or weeks.

The second part defines the instance's *resources*: the entities that attend events. Resources are partitioned into *resource types*. The usual resource types are a Teachers type whose resources represent teachers, a Rooms type of rooms, a Classes type of *classes* (sets of students who attend the same events), and a Students type of individual students. However, an instance may define any number of resource types. Arbitrary sets of resources of the same type, called *resource groups*, may be defined, such as the set of Science laboratories, or the set of senior classes.

The third part defines the instance's *events*: meetings between resources. An event has a *duration* (a positive integer), a *time*, and any number of *resources* (sometimes called *event resources*). The meaning is that the resources are occupied attending the meeting for *duration* consecutive times starting at *time*. The duration is a fixed constant. The time may be preassigned or left open to the solver to assign. Each resource may also be preassigned or left open to the solver to assign, although the type of resource to assign is fixed. Arbitrary sets of events, called *event groups*, may be defined. A *course* is an event group representing the events in which a particular class studies a particular subject. Some event groups may be labelled as courses.

For example, suppose class 7A meets teacher *Smith* in a Science laboratory for two consecutive times. This is represented by one event with duration 2 containing three resources: one preassigned Classes resource 7A, one preassigned Teachers resource *Smith*, and one open Rooms resource. Later, a constraint will specify that this room should be selected from the *ScienceLaboratories* resource group, and define the penalty imposed on solutions that do not satisfy that constraint.

If class 7A meets for Science several times each week, several events would be created and placed in an event group labelled as a course. However, it is common in high school timetabling for the total duration of the events of a course to be fixed, but for the way in which that duration is broken into events to be flexible. For example, class 7A might need to meet for Science for a total duration of 6 times per week,

Table 1 The 15 types of constraints, with informal explanations of their meaning.

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

in events of duration 1 or 2, with at least one event of duration 2 during which the students carry out experiments. One acceptable outcome would be five *sub-events*, as these fragments are called, of durations 2, 1, 1, 1, and 1; another would be three sub-events, of durations 2, 2, and 2. This is modelled by giving a single event of duration 6. Later, constraints specify how this event may be split into sub-events, and define the penalty imposed on solutions that do not satisfy those constraints.

The fourth and last part of an instance contains an arbitrary number of *constraints*, representing conditions that an ideal solution would satisfy. There are 15 types of constraints, stating that events should be assigned times, prohibiting clashes, and so on. The full list appears in Table 1. More types may be added in the future, if necessary.

Each type of constraint has its own specific attributes. For example, a Prefer Times constraint lists the events whose time it constrains, and the preferred times for those events. Each constraint also has attributes common to all constraints, including a Boolean value saying whether the constraint is hard or soft, and an integer weight.

The *infeasibility value* of a solution is the sum over the hard constraints of the number of violations of the constraint multiplied by its weight. The *objective value* of a solution is similar, only summed over the soft constraints. One solution is considered better than another if it has a smaller infeasibility value, or an equal infeasibility value and a smaller objective value.

As mentioned earlier, solutions are stored separately from instances, in solution groups within the archive file. A solution is a list of sub-events, each containing a duration, a time assignment, and some resource assignments. The HSEval web site (Kingston 2010) calculates the infeasibility and objective values of the solutions of an archive, and displays comparative tables, lists of violations, and so on.

3 The competition

The competition attracted 17 registrations, although only 5 teams submitted solutions in the end. This is many fewer than the over 100 registrations and about 40 active

Table 2 The source country, instance name, number of times, teachers, rooms, classes (groups of students), individual students, and events, of each of the 21 instances of Round 1 (archive XHSTT-2012.xml).

Country	Instance	Times	Teachers	Rooms	Classes	Students	Events
Australia	BGHS98	40	56	45	30		387
Australia	SAHS96	60	43	36	20		296
Australia	TES99	30	37	26	13		308
Brazil	Instance1	25	8		3		21
Brazil	Instance4	25	23		12		127
Brazil	Instance5	25	31		13		119
Brazil	Instance6	25	30		14		140
Brazil	Instance7	25	33		20		205
UK	StPaul	27	68	67	67		1227
Finland	Artificial	20	22	12	13		169
Finland	College	40	46	34	31		387
Finland	HighSchool	35	18	13	10		172
Finland	Secondary	35	25	25	14		280
Greece	HighSchool1	35	29		66		372
Greece	Patras2010	35	29		84		178
Greece	Preveza2008	35	29		68		164
Italy	Instance1	36	13		3		42
Netherlands	GEPRO	44	132	80	44	846	2675
Netherlands	Kottenpark2003	38	75	41	18	453	1156
Netherlands	Kottenpark2005	37	78	42	26	498	1235
South Africa	Lewitt2009	148	19	2	16		185

teams of ITC2007. Why fewer teams registered is not known, but it could be because of high school timetabling's lower profile, or because the 15 types of constraints make the instances awkward to handle in practice.

The competition had three independent parts, called Rounds 1, 2, and 3. In Round 1, participants were invited to submit solutions to 21 published instances. No restrictions were placed on how the solutions could be obtained. For each instance, a small prize was awarded to the participant who submitted the best solution to that instance, if it improved on the best solution previously known to the organizers. Such improved solutions were found to 15 of the 21 published instances during Round 1.

The 21 published instances are listed in Table 2. The table shows that they differ greatly in size. For example, the number of times varies between 25 and 148, and the number of classes (groups of students) varies between 3 and 84. They also differ in structure: some require most events to be split into sub-events, others do not; some require teachers to be assigned as well as rooms, although most do not; and so on.

Round 2 compared solvers under uniform conditions. Solvers were allowed to use only freely available software libraries, and a time limit was imposed: 1000 seconds of single-threaded execution per instance on the organizers' computer. Participants used a benchmark program to estimate how much time on their own computer was equivalent to the time limit on the organizers' computer.

For initial testing, the Round 1 instances were used, except that the Australian instances from Table 2 were omitted. This was because these are the only instances so far to use Avoid Split Assignments and Limit Workload constraints, and omitting them reduced the implementation burden for the participants.

The original plan was to select up to 5 finalists based on this initial testing, but since there were only 5 active teams, all were invited to become finalists. One team declined owing to problems with their solver. This left 4 finalists, who submitted their solvers, which the organizers ran on their own computer. For each of 18 hidden instances, and for each of 10 random seeds, each solver was run on that instance and seed for at most 1000 seconds, and the solutions for that instance and seed were ranked. Each solver's ranks were averaged. The solver with the lowest average rank was declared the winner.

The hidden instances were instances not previously published, although not very different from the published instances, and consisted of 14 completely new instances plus 4 small corrections or variants of the published instances. Although none of the organizers were participants, one finalist contributed some of these hidden instances. Their results on their own instances were excluded from the rankings.

The Round 2 finalists have described their solvers in invited short papers for the PATAT 2012 conference, written before results were announced (Domrös and Homberger 2012; Fonseca et al. 2012; Kheiri et al. 2012; Sørensen et al. 2012).

After Round 2, the hidden instances were published and Round 3 was conducted on them. As for Round 1, no restrictions were placed on how the solutions could be obtained. The participant with the lowest average rank over the hidden instances was declared the winner, again excluding participants' results on their own instances.

After the competition ended, the solutions were published on the competition web site, as XHSTT archive files. The results of Rounds 2 and 3 were calculated by passing these files to the appropriate HSEval operation, so can be publicly verified.

4 Conclusion

The aim of the competition, in brief, was to raise the profile of high school timetabling. This it has undoubtedly done.

Judging by previous competitions, the instances used in ITC2011 are likely to continue to attract researchers for years to come. These are unsimplified instances from real high schools around the world—a great foundation to build on.

References

- Jonathan Domrös and Jörg Homberger, An evolutionary algorithm for high school timetabling, Proceedings of the Ninth International Conference on the Practice and Theory of Automated Timetabling (PATAT 2012), Son, Norway, August 2012
- G.H.G. Fonseca, H.G. Santos, T.A.M. Toffolo, S.S. Brito, and M.J.F. Souza, A SA-ILS approach for the high school timetabling problem, Proceedings of the Ninth International Conference on the Practice and Theory of Automated Timetabling (PATAT 2012), Son, Norway, August 2012
- Ahmed Kheiri, Ender Özcan, and Andrew J. Parkes, HySST: hyper-heuristic search strategies and timetabling, Proceedings of the Ninth International Conference on the Practice and Theory of Automated Timetabling (PATAT 2012), Son, Norway, August 2012

- Jeffrey H. Kingston, The HSEval High School Timetable Evaluator, <http://www.it.usyd.edu.au/~jeff/hseval.cgi> (2010)
- B. McCollum, A. Schaerf, B. Paechter, P. McMullan, R. Lewis, A. Parkes, L. Di Gaspero, R. Qu, and E. Burke, Setting the research agenda in automated timetabling: the Second International Timetabling Competition, *INFORMS Journal on Computing* 22, 120–130 (2010)
- Ben Paechter, Luca Maria Gambardella, and Olivia Rossi-Doria, The First International Timetabling Competition, <http://www.idsia.ch/Files/ttcomp2002/> (2002)
- Gerhard Post, Jeffrey H. Kingston, Samad Ahmadi, Sophia Daskalaki, Christos Gogos, Jari Kyngäs, Cimmo Nurmi, Haroldo Santos, Ben Rorije and Andrea Schaerf, An XML format for benchmarks in high school timetabling II, *Annals of Operations Research* (online), <http://10.1007/s10479-011-1012-2>, 2011
- Gerhard Post, Samad Ahmadi, Sophia Daskalaki, Jeffrey H. Kingston, Jari Kyngäs, Cimmo Nurmi, and David Ranson, An XML format for benchmarks in high school timetabling, *Annals of Operations Research* 194, 385–397, 2012
- Gerhard Post, Benchmarking project for High School Timetabling, <http://www.utwente.nl/ctit/hst/> (2011)
- Matias Sørensen, Simon Kristiansen, and Thomas R. Stidsen, International timetabling competition 2011: an adaptive large neighborhood search algorithm, *Proceedings of the Ninth International Conference on the Practice and Theory of Automated Timetabling (PATAT 2012)*, Son, Norway, August 2012