Ann Oper Res (2012) 194:385–397
DOI 10.1007/s10479-010-0699-9

An XML format for benchmarks in High School Timetabling

Gerhard Post · Samad Ahmadi · Sophia Daskalaki · Jeffrey H. Kingston · Jari Kyngas · Cimmo Nurmi · David Ranson

Published online: 6 February 2010

Abstract The High School Timetabling Problem is amongst the most widely used timetabling problems. This problem has varying structures in different high schools even within the same country or educational system. Due to lack of standard benchmarks and data formats this problem has been studied less than other timetabling problems in the literature. In this paper we describe the High School Timetabling Problem in several countries in order to find a common set of constraints and objectives. Our main goal is to provide exchangeable benchmarks for this problem. To achieve this we propose a standard data format suitable for different countries and educational systems, defined by an XML schema. The schema and datasets are available online.

This research has been supported by BSIK grant 03018 (BRICKS: Basic Research in Informatics for Creating the Knowledge Society).

G. Post (✉)
Department of Applied Mathematics, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands
e-mail: g.f.post@math.utwente.nl

G. Post
ORTEC, Groningenweg 6k, 2803 PV Gouda, The Netherlands

S. Ahmadi
School of Computer Science, De Montfort University, The Gateway, Leicester, LE1 9BH, UK

S. Daskalaki
Engineering Sciences Department, University of Patras, 26500 Rio Patras, Greece

J.H. Kingston
School of Information Technologies, The University of Sydney, Sydney, Australia

J. Kyngas · C. Nurmi
Satakunta University of Applied Sciences, Tiedepuisto 3, 28600 Pori, Finland

D. Ranson
Department of Informatics, University of Sussex, Falmer, Brighton, BN1 9QH, UK

Keywords Timetabling · High school · Benchmark · Xml · Scheduling

1 Introduction

One of the best known timetabling problems is the High School Timetabling Problem; most people have some experience of timetables from their school days, with probably a not always positive opinion on it. There is an implicit belief that all High School Timetabling instances are similar, and that a computer program can always 'solve' it.

In reality, the research in this area is still very active and we are nowhere near solving all the instances of the High School Timetabling Problem to optimality. Moreover, continuous reforms in educational systems throughout the world generate new problems to tackle. On the literature side there are papers on general timetabling problems such as de Werra (1985), Cooper and Kingston (1993), Schaerf (1999), Carter and Laporte (1998), de Werra (1999), Burke and Petrovic (2002). Some of their concepts and/or methods can be used in real-life timetabling problems. On the other hand, there are also case studies of high schools in particular countries, some of which are mentioned in Sect. 2.

It is surprising that no standard format for exchanging datasets in the field of High School Timetabling has emerged until now. An accepted format would greatly facilitate the progress in the field. We try to fill in this gap by providing an XML format for existing and new High School Timetabling datasets. As a starting point, the authors of this paper have agreed to contribute at least one of their datasets in this XML format. These datasets will be made available on the internet at the web site devoted to this project (Post 2008).

XML has been chosen because it is extensively used nowadays in web services, or as a means of exchanging data between applications. The power of XML lies in the fact that the data is structured, and therefore it can easily be adjusted to changes, unlike plain text files. Programming languages have software libraries which make it easy to handle XML. Mathematically speaking, XML is a rooted tree, with information attached to the nodes. This information can contain cross references. Examples of XML are presented in Sect. 4; see also http://www.w3schools.com/xml/; http://en.wikipedia.org/wiki/XML.

The problem of exchanging timetabling data was discussed at the first International Conference on the Theory and Practice of Automated Timetabling (Cumming and Paechter 2005), where it became clear that the principal difficulty was the precise expression of the many different kinds of constraints. There may be ten or more kinds in any one instance, and they vary widely between institutions.

This complexity of specification has been addressed in several ways. Some papers have tried to generalize and unify the constraints (Chand 2004; Kitagawa and Ikeda 1988). Others have adapted existing technologies in which constraints may be expressed, such as XML and the semantic web (Custers et al. 2005; De Causmaecker et al. 2000; De Causmaecker et al. 2002; Özcan 2005), or object-oriented modeling and frameworks (Gröbner et al. 2003; Ranson and Ahmadi 2006). Others have expressed constraints as logic expressions within specifically designed specification languages (Burke et al. 1998; Kingston 2001; Monteiro da Mata et al. 1997). There has been at least one attempt to simply enumerate every possible constraint (Reis and Oliveira 2001). While much of this work is more general than our approach (for example, the use of MathML functions in Özcan (2005) allows essentially arbitrary constraints to be expressed in XML), none of it has led to significant data exchange.

Another approach is to restrict the problem domain to one particular kind of timetabling, then use judicious simplification to further reduce the specification burden while maintaining the essence of the problem. The Carter data sets for examination timetabling (Carter

et al. 1996) omit many details, notably all data related to rooms, and similar simplifications appear in the Traveling Tournament Problem (Easton et al. 2001) and the International Timetabling Competition (Paechter 2003). These are some of the most successful examples of timetabling data exchange so far. However, judicious simplification has been criticized for contributing to the gap between research and practice (Burke et al. 2006), at least in examination timetabling; and the data transfer has almost always been in one direction only.

2 The timetabling problem in different countries

Throughout the world, timetables are constructed for many different types of high schools. In the following subsections we describe the situation in the schools of Australia, England, Finland, Greece and the Netherlands. However, before getting into the details of their problems, we provide an overview of the problem and their principal differences. Not only the problems, but also the terminology varies among countries. For more details we refer to Sect. 3.1.

2.1 General aspects of the problem

The timetabling process in a given school is influenced by the school organization, and specifically the handling of the three groups of resources: students, teachers, and rooms. In the heart of the high school timetabling problem lie the students, who are usually organized in *base groups*, called 'Student divisions' or 'Student groups'. These base groups form a partitioning of the students, i.e. each student belongs to exactly one base group. We may consider two types of timetabling problems:

1. Problems for which the base groups never split up over different lessons, i.e. if two students of a base group have a lesson at the same time, they are in the same lesson.
2. Problems for which the students of a base group may attend different lessons for some times.

In practice not many schools will fall completely into the first category. Some schools, for example, will split two base groups of boys and girls for gym lessons. Similarly, other schools may split base groups for reasons such as religion, different levels of ability in a subject, etc.

Another important difference between countries is the compactness of the students' schedules. Compact schedules are schedules with no idle times. An *idle time* for a student is a time (slot) when the student has no lesson, between others on the same day where the student does have a lesson. In most cases compact schedules are required for the students. In some countries this is automatic, as a student has as many lessons as there are times. However, when a school has a large variety of elective subjects it is almost impossible to achieve compact schedules. Then the objective would be to minimize the number of idle times.

The teachers form the second most important group of resources affected by the timetable. They are normally preassigned to lessons by the school management. The assignments take into consideration all limiting constraints like the number of staff, their expertise and other aspects which may be difficult to model for an automated system. Examples of such difficult constraints would be assigning new teachers to lessons that take place in parallel, balancing teachers assigned to a base group of students (male/female, experienced/less experienced), or avoiding parent-child combinations in lessons.

Idle times in teachers' schedules also form a point of differentiation among different countries. If teachers are preassigned to lessons, it is extremely hard (except only in simple

cases) to find schedules with no idle times for both students and teachers. When idle times are not allowed for students, typically one can ask only to minimize the number of idles for teachers.

Another important issue is the spread of lessons for the teachers. Sometimes, as with students, this is almost automatic for full-time teachers. For part-time teachers, however, particular kinds of spreading may be important. For example, a teacher with an assignment of only 5 times per week will probably be unhappy with a schedule that spreads them over all days of the week.

The rooms of a school form the third major group of resources. Although most lessons will take place in a room, this does not necessarily imply that room assignment is an issue. In many schools, a room is preassigned to each base group of students, or to each teacher. It is also possible that the number of rooms is so large that assigning rooms is never a problem. While in some schools this may be the case for general-purpose rooms, specialized rooms may be scarce and in need of careful scheduling. Examples are specialized rooms for Physical Education, Music, Arts, or Labs for physics, biology and chemistry.

We now proceed with the descriptions of the problems that the different countries face in their schools.

2.2 Australia

Australian high schools have short school days, with correspondingly high teacher utilisations, and consequently no demand for compactness in their timetables. Most teachers (those with no special duties) teach for 75% of the times of the cycle, and the total workload of all teachers is almost 100% of the available workload permitted by the teachers' workload limits. These conditions commonly prevent teachers from being preassigned, except to a few key subjects (e.g. the senior ones), and force some courses to be shared between two teachers. Minimizing the number of these *split assignments* is a key goal. There is a very high utilization of specialized rooms, notably Science and Computer laboratories.

The above describes the situation in government funded high schools. Privately funded high schools are often better resourced, with lower teacher workloads and more laboratories. However, they typically have lower student numbers and a commitment to high student choice, leading to peculiarities such as *composite classes*, in which students of different ages study a common subject together under a single teacher, creating other difficulties.

Another Australian problem concerns the number and distribution of lessons for each subject. In the senior years this is very simple: for example, a senior student might attend six subjects, each occupying six times per week, spread across the five days, including a double period (i.e. two adjacent times not separated by a break) on one of the days. Student choice is catered for by means of *electives*: lists of subjects running simultaneously from which each student chooses one. In the junior years a desire to permit students to sample a wide range of subjects leads to a chaotic curriculum from the timetabling point of view. A few subjects (English, Mathematics, and Science) follow the senior pattern, but the remainder of the cycle is occupied by many small subjects with more or less random numbers of times.

For previous research and additional details we refer to Abramson (1991), Kingston (2005).

2.3 England

In English Secondary Schools students study a mixture of compulsory and optional subjects. For the 11- to 14-year-olds all subjects are compulsory and students study 10 main subjects

plus a modern foreign language, citizenship education, careers education and sex education. There are fewer compulsory subjects for 14- to 16-year-olds. In addition, schools must offer students some work-related learning during this phase.

Depending on the policy of the school, within each year group the students may be divided into base groups with a designated teacher who has organizational and pastoral care responsibilities.

Another common grouping structure used in British schools is a Band, which is a collection of several tutorial groups. Normally, all the base groups in a year are allocated to fewer bands. The mixing of base groups is done within a band. Subjects are grouped into *blocks*, which are equivalent to the Australian electives: a block consists of one or more subjects for different groups which will be scheduled at the same time. Blocks are constructed manually before the scheduling process starts.

Except in the Sixth Form, British school students receive compact schedules automatically, since they have as many lessons as there are times.

In general there is a large variety of different length of lessons, one or two weekly timetabling cycles, length of school day and hence different number of lessons in different schools. This requires any timetabling system for English schools to be highly parameterised to accommodate different patterns of delivery in different schools.

For previous research and additional details we refer to Wright (1996).

2.4 Finland

A basic goal in Finnish school timetabling is to construct a one-week schedule, which is repeated the whole season. A timetable consists of lessons of subjects, where a lesson is a predefined combination of a student group, a teacher, a room type and the duration of the lesson. Every student belongs to one base group and most of the lessons are scheduled based on this group. In addition, the student can belong to a number of optional groups, which are built according to the enrolment of students in optional courses. Every lesson is assigned to one and only one group, either base or optional. Base and optional groups define a compatibility matrix, which determines which groups can or cannot have lessons at the same time.

In all Finnish school levels teachers are preassigned to lessons. Rooms are preassigned to most lessons, based on the preferences of the teachers.

In a typical curriculum a student attends ten subjects, each taught for two to six hours each week. Lessons of a subject can take one, two, or (in exceptional cases) three hours. Hence, most subjects are taught either within two, three or four days each week.

Compact schedules for the students (student groups) are necessary and idle times are either strictly prohibited or highly inappropriate. Conversely, idle times for the teachers are allowed, but still are very unappreciated. Several teachers also prefer not to teach more than a given number of hours in a day.

Preassignment of teachers and rooms, a somewhat complicated structure of student groups and the demand for compact scheduling make Finnish School Timetabling Problem a challenge for both a manual solver and a computer software. A further description of the Finnish problem can be found in Nurmi and Kyngas (2007).

2.5 Greece

Secondary education in Greece is divided in two portions: *lyceum* (grades 7 to 9) and *gymnasium* (grades 10 to 12). The last two years in the lyceum are considered as preparatory for

the higher level of education and the students are asked to choose one of three directions. Similarly, in vocational lyceums the students are asked to choose one of several specializations.

Aspects common to the two school types are the five days of the week, the number of times per day (six to seven), the preassignment of lessons to teachers, who may be full-time or part-time, and the requirement for compact student schedules.

In the gymnasium all students of a given base group attend the same lessons during most of the time in their dedicated classroom. However, for a small portion of the weekly timetable some base groups split into sub-groups or reshuffle with some other base group and split again for attending certain 'special' subjects. For these subjects it is required that two teachers are teaching simultaneously to two different sub-groups of students, or alternatively that two teachers are collaboratively teaching to the same group of students.

In the lyceum the general practice is to keep students in their regular base groups for a portion of the day to attend the lessons that are common to all directions or specializations, then reshuffle and split again based on specializations, directions or elective courses until the end of the day.

Important objectives for the timetabling in the gymnasium are to schedule core courses evenly during the week, preferably during the prime times, and to maintain a minimal number of idle times for the teachers. For the lyceums, however, the objectives change because of all the scheduling considerations mentioned previously. Balanced distribution of the core courses during the week is still important; however, the other two objectives cannot be achieved.

For previous research and additional details we refer to Birbas et al. (1997), Valouxis and Housos (2003).

2.6 The Netherlands

In Dutch secondary schools different levels of education are offered within one school. Teachers are shared among these levels. In the lower years the students in a base group follow the same courses, while in the last years, the students choose a specialization, with some compulsory and some optional subjects. The timetable is usually weekly, and valid for a period of 6 weeks, a trimester, semester, or the whole year. In the lower levels compact schedules often are compulsory. In the higher levels compact schedules are impossible to realize for all students, and in the opinion of the school administration not necessary. However, avoiding idle times as much as possible for students as well as teachers is still important.

Usually the teachers are preassigned to the lessons by the school administration, to ensure a good spreading of the teachers. The majority of the teachers work part-time. According to collective labour agreements these teachers are entitled to one or more days off.

The rooms are generally not preassigned to lessons. A lesson requires a room of a certain type, and lessons should be assigned such that enough rooms of each type are available. High utilization occurs for specialized rooms, like the gyms.

Some schools have special arrangements for students good at sports, dance or ballet, allowing them to skip lessons at the beginning or the end of some days. Some schools experiment with large groups of students (50 to 60 students) in a learning studio. Effectively there are two teachers visiting them in a period, of which one acts as supervisor, and the other one teaches the subject. All these special constructions make the task of timetabling even more challenging.

For previous research and additional details we refer to de Gans (1981), Willemen (2002), de Haan et al. (2007).

3 Modeling the problem

The basic game we play in modeling is between abstraction and concreteness. In an abstract sense, we could only view events, and express all constraints in the events. For accessibility we decided to define Times and Resources as well. Groups of times, resources and events may be defined: TimeGroups, ResourceGroups, and EventGroups. This makes the formulation of constraints much more readable and compact. For example, an instance will usually contain the 'AvoidClashConstraint' as a hard constraint for all resources, i.e. for the resource group consisting of all resources.

3.1 Times and resources

We divide the time period into *times*. In many constraints it is essential to know the relation between groups of times, for example for idle times per day or week, or the number of lesson per day. Days and weeks are time groups, which are treated on the same basis as other time groups in the constraints. They can be introduced to make a dataset more understandable.

A *resource* is an entity which attends events. The most common resources in the timetabling problem are the students, teachers, and rooms. Other kinds of resources, such as equipment (video projectors, etc.) are possible but uncommon. Resources may be classified into subgroups, for example rooms of certain type or student groups of a certain form. It is a basic hard constraint of most timetabling problems that no resource may attend two events scheduled for the same time. A violation of this constraint is called a *clash*.

We attach no additional properties to resources; all additional requirements will be modeled with the help of the constraints, where the involved resources will be selected.

As stated before, we distinguish three main kinds of resources:

- *Students*. Usually a student group is preassigned to each events. Constraints that can be important for students are controlling the number of idle times, the number of lessons per day.
- *Teachers*. As mentioned earlier, in some countries teachers are preassigned to events, while in others they are not. In the latter case the choice of assignment will be restricted by teachers' qualifications and workload limits. Important issues for the teachers are the total number of idle times, the number of assigned times per day, the number of days with events, and unavailable days or times.
- *Rooms*. Most events take place in a room. If rooms abound, we can disregard them, if not they can be a bottleneck, especially for specialized rooms.

3.2 Events

Events are the basic scheduling objects. An event can represent either a single lesson, or a set of ('linked') lessons, that have to be taught at the same time.

We use the term *course* for a collection of events taught to a group of students in a subject. Events that refer to the same course, are called the *lessons* of the course. In other papers such a set of lessons might be called a class. The most common constraint related to courses concerns the distribution of the lessons of a given course in the week (or in different weeks). In situations where the teachers are not preassigned to courses, an important constraint is that all lessons of a given course should be assigned to the same teacher. In our format a course is a specialized event group, like the day and week are specialized time groups.

Other types of events, such as staff meetings, are also permitted. An event has the following properties.

- The *duration*, which is the number of times that have to be assigned to the event. These times must be consecutive.
- The *course* related to an event.
- The *time* that the event is scheduled to. If the duration is larger than one, the event will occupy some following times as well. Constructing a timetable involves assigning a time to all events.
- The *workload* that the event will contribute to a teacher's total workload. Often this is the same as the duration. The workload is only needed when teachers are not preassigned, and consequently total workloads must be calculated.
- The *resource groups* are preassigned.
- Any number of *resources*, either preassigned, or to be assigned. These each have a 'role', used to identify them. A role might have value 'room', 'teacher', 'senior teacher', and so on. A resource may be constrained to be from a particular resource group, for example a Science laboratory or an English teacher.

3.3 Constraints

Our XML format currently defines 13 constraint types. We expect that this number will grow in the future. A constraint can be hard ('Required') or soft. In both cases the constraint generates a cost, which either contributes to the infeasibility value, or to the objective value. The constraints describe all aspects of the scheduling, even the (elementary) assignment part. The advantage of this approach is that we can distinguish between levels of infeasibility: if not all events can be scheduled, we can give preferences among them. In Sect. 4.2 we give an example how a constraint contributes to the objective function.

The cost of a schedule consists of three parts: the cost of a resource, cost of an event group, cost of an event.

We group the constraints into three groups: constraints describing the basic scheduling problem, other constraints for events, and constraints for resources.

3.3.1 Basic scheduling constraints

- *AssignTimeConstraint* (Cost per event). Assign a time to each of the selected events.
- *AssignResourceConstraint* (Cost per event). Assign a resource to the role in each of the selected events.

Both constraints have a variant expressing the preference for the time, respectively the resource: *PreferTimesConstraint* and *PreferResourcesConstraint*.

3.3.2 Event constraints

- *LinkEventsConstraint* (Cost per event group). Schedule the selected event groups at the same (starting) time.
- *SpreadEventsConstraint*. (Cost per event group). Schedule the events of the selected event groups to the selected time groups between a minimum and a maximum number of times.
- *AvoidSplitAssignmentsConstraint*. (Cost per event group). For each selected event group, schedule the selected role of each event of this group to the same resource.

3.3.3 Resource constraints

Resource constraints describe the quality of the timetable of a single resource; the corresponding cost is attributed to the resource.

- *AvoidClashesConstraint*. Schedule the selected resources without clashes. This is one of the basic (hard) constraints.
- *AvoidUnavailableTimesConstraint*. Avoid that the selected resources are busy in the selected times.
- *LimitWorkloadConstraint*. Schedule workload to the selected resources between a minimum and a maximum.
- *LimitIdleTimesConstraint*. The number of idle times in the selected time groups should lie between a minimum and a maximum for each of the selected resources. Typically the time groups are a day or all days.
- *LimitBusyTimesConstraint*. The number of occupied times for the selected resources should lie between a minimum and a maximum for each of the selected time groups. Typically the time groups are the days.
- *ClusterBusyTimesConstraint*. The number of time *groups* with an assigned time should lie between a minimum and a maximum for the selected resources. Typically the time groups are days; for example a teacher requiring at most 3 days with lessons.

4 The XML format for benchmarks

XML (http://www.w3schools.com/xml/; http://en.wikipedia.org/wiki/XML) is a mark-up language similar to LaTeX and HTML. It is used extensively for exchanging data between applications. Most object-oriented languages, like Java, C++, C#, and Delphi, have extensive libraries to work with XML. For this reason XML seems to be very useful for benchmarking. For a comparable project in nurse rostering, see (Curtois 2006).

The XML format is defined by an XML schema, which is an XSD file (itself written in XML). The schema defines what elements must be and may be present in the XML file. The format is straightforward and to present it all would be tedious. Instead, we give three examples which should make the general flavour clear: the top-level format, one constraint, and the format of solutions. For a complete description, see (Post 2008), where a list with terms used can be found, as well as a detailed description of all constraints. As an introductory example the 4x4-Sudoku formulated as timetabling problem (without preassignments, and hence multiple solutions) is available at this website.

4.1 The top-level format

At the top level the XML file looks like this:

```
<HighSchoolTimetableArchive>
  <Instances>
    <Instance Id="Sudoku4x4">
      <Times>
         ...
      </Times>
      <Resources>
         ...
```

```
      </Resources>
      <Events>
         ...
      </Events>
      <Constraints>
         ...
      </Constraints>
    </Instance>
  </Instances>
  <SolutionGroups>
      ...
  </SolutionGroups>
</HighSchoolTimetableArchive>
```

The file contains one or more instances and any number of grouped solutions. The instance contains 'Times', 'Resources', 'Events', and 'Constraints' sections, as in the data model (Sect. 3).

4.2 Constraints

We give one example of a constraint, penalizing idle times. An idle time is relevant to a resource and to a group of times. If there is a time without event scheduled to the resource, but with scheduled events before and after in the same time group, we call it an idle time for the resource within this time group. Usually the time group will consist of all times of a day. In the constraints section of the XML document, we could have, among other constraints, the following:

```
<LimitIdleTimesConstraint Id="StudentIdleTimes">
  <Name>At most three idle times for students per week</Name>
  <Required>false</Required>
  <Weight>10</Weight>
  <CostFunction>SquareSum</CostFunction>
  <AppliesTo>
    <ResourceGroups>
      <ResourceGroup Reference="Students"/>
    <ResourceGroups>
  </AppliesTo>
  <TimeGroups>
    <TimeGroup Reference="Monday"/>
    <TimeGroup Reference="Tuesday"/>
    <TimeGroup Reference="Wednesday"/>
    <TimeGroup Reference="Thursday"/>
    <TimeGroup Reference="Friday"/>
  </TimeGroups>
  <Minimum>0</Minimum>
  <Maximum>3</Maximum>
</IdleTimesConstraint>
```

The attribute 'Id' and the first four fields are present in all constraints. They define a unique reference (Id) and a display name (Name), and give information on how to interpret violations this constraint. Required = "false" means that this constraint is a soft constraint, and

hence any cost will be added to the objective value. The Weight and CostFunction explain how to calculate the cost for the deviation of the violation. In this constraint a violation is that one of the students in the resource group 'Students' has more than 3 idle times in the week. The deviation is the surplus (per week), so the number of idle times minus 3. If a student has 1, 0, 2, 3, 1 idle times, on the five days of the week respectively, the total number of idle times is 7, and the deviation is 4. Since we use a quadratic term for this cost and multiply by the weight, we obtain the cost 160. Hence the (constraint, student) combination adds an amount of 160 to the objective function.

This example sums the idle times per day. If we only are interested in the idle times on Monday, we can omit the time groups for the other days.

4.3 Solutions

The solution in XML is presented as:

```
<Solution Id="Sudoku4x4">
  <Events>
    <Event Reference="Event1">
      <Time Reference="Day_1"/>
      <Resources>
        <Resource Reference="R1">
          <Role>Room</Role>
        </Resource>
      </Resources>
    </Event>
    <Event Reference="Event2">
        ...
    </Event>
      ...
  </Events>
</Solution>
```

The main thing a solution should do is tell the result of the two scheduling constraints. For this each Event section is filled with the time and resources of the corresponding assignment constraints. Once these aspects are known, all other information can be calculated, and a summary can be added. For debugging purposes, we introduce the possibility to give a full report on the violations. This report gives for each resource, event group, and event the constraints that are violated, and for each of the violated constraints the deviation, cost, and a text explaining the violation. At Kingston (2009) an evaluator is available which validates the XML in the dataset, and returns the report on the submitted solutions.

5 Conclusion

The XML format advocated here has been designed in order to better model the complete high school timetabling problem and facilitate data exchange between high school timetabling researchers. We used as input the situation in five different countries, and we think that the presented format can deal with these countries. This belief is based on several datasets we have in our 'old' formats. It is hoped that researchers and practitioners will consider adopting this format in their work and that this will lead to further discussions and

improvements to the model. As previously stated we fully expect the number of included constraints to expand. We actively will encourage others to contribute datasets in this format, and use the evaluator (Kingston 2009) to validate their datasets. This should not only lead to improvements of the model but also to the development of better and/or more general algorithms.

References

Abramson, D. (1991). Constructing school timetables using simulated annealing: sequential and parallel algorithms. *Management Science*, *37*, 98–113.

Birbas, T., Daskalaki, S., & Housos, E. (1997). Timetabling for Greek high schools. *Journal of the Operational Research Society*, *48*, 1191–1200.

Burke, E. K., & Petrovic, S. (2002). Recent research directions in automated timetabling. *European Journal of Operational Research*, *140*, 266–280.

Burke, E. K., Kingston, J. H., & Pepper, P. A. (1998). A standard data format for timetabling instances. In E. Burke & M. Carter (Eds.), *Lecture notes in computer science: Vol. 1408. Practice and theory of automated timetabling II* (pp. 213–222). Berlin: Springer.

Burke, E. K., McCollum, B., McMullan, P., & Qu, R. (2006). Examination timetabling: a new formulation. In: *Proceedings of the sixth international conference of the practice and theory of automated timetabling (PATAT 2006)*, Brno, 2006 (pp. 373–375).

Carter, M., Laporte, G., & Lee, S. T. (1996). Examination timetabling: algorithmic strategies and applications. *Journal of the Operational Research Society*, *47*, 373–383.

Carter, M. W., & Laporte, G. (1998). Recent developments in practical course timetabling. In E. Burke & M. Carter (Eds.), *Lecture notes in computer science: Vol. 1408. Practice and theory of automated timetabling II* (pp. 3–19). Berlin: Springer.

Chand, A. (2004). A constraint based generic model for representing complete university timetabling data. In: *Proceedings of the fifth international conference on the practice and theory of automated timetabling (PATAT 2004)*, Pittsburgh, 2004 (pp. 125–148).

Cooper, T. B., & Kingston, J. (1993). The solution of real instances of the timetabling problem. *The Computer Journal*, *36*, 645–653.

Cumming, A., & Paechter, B. (2005). *Standard formats for timetabling data*. Unpublished discussion session at the first international conference on the practice and theory of automated timetabling, Edinburgh, 2005.

Curtois, T. (2006). Nurse rostering web site. http://www.cs.nott.ac.uk/~tec/NRP/.

Custers, N., De Causmaecker, P., Demeester, P., & Vanden Berghe, G. (2005). Semantic components for timetabling. In E. Burke & M. Trick (Eds.), *Lecture notes in computer science: Vol. 3616. Practice and Theory of Automated Timetabling V'* (pp. 17–33). Berlin: Springer.

De Causmaecker, P., Demeester, P., De Pauw-Waterschoot, P., & Vanden Berghe, G. (2000). Ontology for timetabling. In: *Proceedings of the third international conference on the practice and theory of automated timetabling (PATAT 2000)*, Konstanz, 2000 (pp. 481–482).

De Causmaecker, P., Demeester, P., Lu, Y., & Vanden Berghe, G. (2002). Using web standards for timetabling. In: *Proceedings of the fourth international conference on the practice and theory of automated timetabling (PATAT 2002)*, Gent, 2002 (pp. 238–257).

de Gans, O. B. (1981). A computer timetabling system for secondary schools in the Netherlands. *European Journal of Operational Research*, *7*, 175–182.

de Haan, P., Landman, R., Post, G., & Ruizenaar, H. (2007). A case study for timetabling in a Dutch secondary school. In E. Burke & H. Rudová (Eds.), *Lecture notes in computer science: Vol. 3867. Practice and theory of automated timetabling VI* (pp. 267–279). Berlin: Springer.

de Werra, D. (1985). An introduction to timetabling. *European Journal of Operational Research*, *19*, 151–162.

de Werra, D. (1999). On a multiconstrained model for chromatic scheduling. *Discrete Applied Mathematics*, *94*, 171–180.

Easton, K., Nemhauser, G. L., & Trick, M. A. (2001). The travelling tournament problem: description and benchmarks. In *Lecture notes in computer science: Vol. 2239. Principles and practice of constraint programming (CP 2001)* (pp. 580–585). Berlin: Springer.

Gröbner, M., Wilke, P., & Büttcher, S. (2003). A standard framework for timetabling problems. In E. Burke & P. De Causmaecker (Eds.), *Lecture notes in computer science: Vol. 2740. Practice and theory of automated timetabling IV* (pp. 24–38). Berlin: Springer.

Kingston, J. H. (2001). Modelling timetabling problems with STTL. In E. K. Burke & W. Erben (Eds.), *Lecture notes in computer science: Vol. 2079. Practice and theory of automated rimetabling III* (pp. 309–321). Berlin: Springer.

Kingston, J. H. (2005). A tiling algorithm for high school timetabling. In E. Burke & M. Trick (Eds.), *Lecture notes in computer science: Vol. 3616. Practice and theory of automated timetabling V* (pp. 208–225). Berlin: Springer.

Kingston, J. H. (2009). The HSEval High School Timetable Evaluator. http://www.it.usyd.edu.au/~jeff/hseval.cgi.

Kitagawa, F., & Ikeda, H. (1988). An existential problem of a weight-controlled subset and its application to school timetable construction'. *Discrete Mathematics*, *72*, 195–211.

Monteiro da Mata, J., Luiz de Senna, A., & Augusto de Andrade, M. (1997). Towards a language for the specification of timetabling problems. In *Proceedings of the second international conference on the practice and theory of automated timetabling (PATAT'97)*, Toronto, 1997 (pp. 330–333).

Nurmi, K., & Kyngas, J. (2007). A framework for school timetabling problem. In: *Proceedings of the 3rd multidisciplinary international scheduling conference: theory and applications*, Paris, 2007 (pp. 386–393).

Özcan, E. (2003). Towards an XML-based standard for timetabling problems: TTML, multidisciplinary Scheduling: theory and applications. In *First international conference, MISTA '03*, Nottingham, Selected Papers (2005) (pp. 163–185).

Paechter, B. (2003). International timetabling competition. http://www.idsia.ch/Files/ttcomp2002/.

Post, G. (2008). High school timetabling web site. http://wwwhome.math.utwente.nl/~postgf/BenchmarkSchoolTimetabling/.

Ranson, D., & Ahmadi, S. (2006). An extensible modelling framework for the examination timetabling problem. In E. Burke & H. Rudová (Eds.) *Lecture notes in computer science: Vol. 3867. Practice and theory of automated timetabling VI* (pp. 383–393). Berlin: Springer.

Reis, L. P., & Oliveira, E. (2001). A language for specifying complete timetabling problems. In E. K. Burke & W. Erben (Eds.) *Lecture notes in computer science: Vol. 2079. Practice and theory of automated timetabling III* (pp. 322–341). Berlin: Springer.

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

Valouxis, C., & Housos, E. (2003). Constraint programming approach for school timetabling. *Computers & Operations Research*, *30*, 1555–1572.

Willemen, R. J. (2002). *School timetable construction; algorithms and complexity*. PhD thesis, Technical University Eindhoven, The Netherlands.

Wright, M. (1996). School timetabling using heuristic search. *Journal of Operational Research Society*, *47*, 347–357.