The Second International Timetabling Competition: Examination Timetabling Track

The Second International Timetabling Competition: Examination Timetabling Track

Barry McCollum, Paul McMullan

School of Computer Science, Queen's University, Belfast,
University Road, N. Ireland, BT7 1NN, UK,
{ b.mccollum, p.p.mcmullan }@qub.ac.uk,

Edmund K. Burke, Andrew J. Parkes, Rong Qu

School of Computer Science, University of Nottingham,
Jubilee Campus, Nottingham, NG8 1BB, UK,
{ ekb, ajp, rxq }@cs.nott.ac.uk

Technical Report: QUB/IEEE/Tech/ITC2007/Exam/v4.0/17. September 20, 2007

Abstract

The 2nd International Timetabling Competition (ITC2007) is made up of three tracks; one on examination timetabling and two on course timetabling. This paper describes the examination timetabling track introduced as part of the competition. Both the model and the datasets are based on current real world instances introduced by EventMAP Limited. It is hoped that the interest generated as part of this competition will lead to the development, investigation and application of a host of novel and exciting techniques not previously trialed within this important real world search domain.

1 Introduction

Building on the success of the First International Timetabling Competition in 2002 [1], the second competition (ITC2007) is introduced with the overall aim of attracting researchers to develop and trial leading edge techniques within a competitive arena. It also aims to further generate interest in the research area by providing various formulations of the timetabling problems encountered within educational institutions based on 'real world' perspective. Particular emphasis is being placed on 'real world' scenarios with the objective of encouraging the production of techniques which have the potential to solve practical instances of the problem. The competition therefore has an important part to play in helping to bridge the current gap which exists between research and practice in this area.

To these ends three tracks are introduced along with a number of associated benchmark datasets. In this paper, we report on the Examination Timetabling Track (Track 1). The information presented here can be regarded as the official documentation for Track 1[1] and complements the content on the ITC2007 web site at

[1] Updated versions of this report may be made available as the competition progresses. Changes will only be made in an attempt to clarify the various issues discussed.

http://www.cs.qub.ac.uk/itc2007. Here, in addition to details on the examination track, some general information on the competition is provided in relation to background, motivation and rules.

The three tracks are examination timetabling (Exam TT), post-enrolment course timetabling (PostEnroll CTT) and curriculum-based course timetabling (Curriculum CTT). Although under the general 'umbrella' of educational timetabling, these three identified problem areas have significant differences which are discussed in detail at the Competition website. In addition, technical reports for all areas are available from the official website and can be found under each track. This paper details the examination track of the competition.

1.1 The Examination Timetabling Problem

As stated in the introduction, in order to tackle the main variations which exist within the practical area of educational timetabling, the current competition has been divided into three sections or tracks. From a research perspective this division is important in that it provides a framework to capture the main types of educational timetabling research currently taking place within the academic community.

Modeling the complexity of timetabling problems continue to represent important issues in the timetabling research area [2]. However these issues have not been widely discussed over the last ten years and there are still no universal complete models [3]. De Werra, Asratian and Durand [4] in 2002 presented a simple model and its possible extensions for class-teacher timetabling problems. The complexity of these problems was also studied, showing some variants of the problem as NP-complete. Further work to address these and related issues is needed to provide fundamental support for better understanding and development of exam timetabling research.

There is also research in the literature on building general timetabling languages and tools in an attempt to model real world instances of the problem. Tsang, Mills and Williams [5] developed a high level language to specify exam timetabling problems as constraint satisfaction problems. Di Gaspero and Schaerf [6] built a software tool called EASYLOCAL++ for easy implementation of local search algorithms on general timetabling problems. A general and reusable framework will further improve the current development and justify easy scientific comparisons.

The last ten years have seen a significant amount of research which addresses aspects from both theory and practice [3]. Meta-heuristics (i.e. Tabu Search [7, 8, 9], Simulated Annealing [10, 11, 12], Genetic Algorithms [13, 14], memetic algorithms [15], ant algorithms [16, 17], etc) represent the most effective state-of-the-art approaches on standard benchmarks. There is also a large amount of work where hybridizations between meta-heuristics are studied. This includes the effective integration of early timetabling techniques such as graph heuristics [18, 19, 20, 21] and constraint based techniques [11, 22, 23]. Along with these main themes of research there are also a number of new trends including more effective design of neighborhood structures (i.e. variable neighborhood search for timetabling [24, 25], etc). Flexibility of search is thus improved to tackle more complex problems with a wider range of constraints in exam timetabling. Some research motivated by the objective of raising the generality of timetabling approaches has also obtained promising results, hyper-heuristics [26] being one of the areas that is attracting much research attention [19, 20, 21, 24, 27, 28]. The last ten years of research on examination timetabling is discussed and reviewed in Qu et al [3].

The Examination Timetabling Problem addressed here introduces a practical formulation of the problem which, organisers believe, significantly adds to current research and provides a firm basis for future efforts in the area. In addition, it is hoped that the interest generated as part of this competition will lead to the development, investigation and application of a host of novel and exciting techniques not previously trialed within this important real world search domain. The problem model can be described as post enrollment. That is to say, students enrolled on particular courses which have associated exams are considered to be enrolled on or 'taking' those exams. Although other approaches to the problem are taken within institutions, this is by far the most common from a practical perspective as well as being the most widely reported model of the problem within the academic literature.

Recent Research has concentrated on a number of benchmark datasets introduced by Carter [29]. These benchmarks and the problems associated with them are discussed in more detail in Qu et al [3]. This particular track of the competition significantly adds to the research field by the introduction of a more 'real' model of the problem in terms of data, constraints and evaluation. All datasets used as part of this competition are taken from Institutions and have been anonymised for the purpose of competition use. At a future time, after the end of the competition, all 12 datasets will be released to the community.

Section 2 gives a non-mathematical description of the problem and motivations. Section 3 gives a mathematical programming formulation of the problem. For ease of exposition we have kept it separate and compact.

2 The Problem Model

The examination timetabling problem model presented here extends the current model of the problem commonly worked upon. The fundamental problem involves timetabling exams into a number of periods within a defined examination session while satisfying a number of hard constraints. Like other areas of timetabling, a feasible solution is one in which all hard constraints are satisfied. The quality of the solution is measured in terms of soft constraints satisfaction.

New and additional Information is provided on constraints (hard and soft), resources and the examination session. For example, in terms of hard constraints, room numbers and sizes are provided. In addition, information on the structure, length and number of individual periods is also presented. In terms of soft constraints, much more practical information is provided in terms of how an organisation measures the overall quality of a solution.

2.1 Problem Description

The problem consists of the following:

- An examination session is made of a number of periods over a specified length of time. Number and length of Periods are provided.
- A set of exams that are to be scheduled into periods. Exam codes are not provided. As with all entities, competitors should assume sequential numbering beginning with 0.
- A set of students enrolled on individual exams. Each student is enrolled on a number of exams. Students enrolled on an exam are considered to 'take' that examination. For each exam, the set of enrolled students is provided.
- A set of rooms with individual capacities are provided.
- Hard Constraints which must be satisfied
- Soft Constraints which contribute to a penalty if they are violated.
- Details including a 'weighting' of particular soft constraints.

A feasible timetable is one in which all examinations have been assigned to a period and room and all the following hard constraints are satisfied:

- No student sits more than one examination at the same time;
- The capacity of individual rooms is not exceeded at any time throughout the examination session;
- Period Lengths are not violated;
- Satisfaction of period related hard constraints e.g. Exam_A after Exam_B;
- Satisfaction of room related hard constraints e.g. Exam_A must use Room 101.

(Notice that, unlike course timetabling, exams are explicitly allowed to share rooms).

The soft constraints can be outlined as follows;

- Two exams in a row

  The number of occurrences when students have to sit two exams in a row on the same day.
- Two exams in a day

  The number of occurrences when students have to sit two exams on the same day. This constraint only becomes important when there are more than two examination periods in the one day. This is further explained later when the evaluation is described.

- Specified spread of examinations.

  The number of occurrences when students have to sit more than one exam in a time period specified by the institution. This is often used in an attempt to be as fair as possible to all students taking exams.

- Mixed duration of examinations within individual periods;

  The number of occurrences of exams timetabled in rooms along with other exams of differing time duration.

- Larger examinations appearing later in the timetable

  The number of 'large' exams appearing in the 'latter portion' of the timetable. Both 'large' and 'later portion' are user defined.

- Period related soft constraints

  The number of times a period is used which has an associated penalty. This is multiplied by the actual penalty as different periods may have different associated weightings.

- Room related soft constraints.

  The number of times a room is used which has an associated penalty. This is multiplied by the actual penalty as different rooms may have different associated weightings.

These can effectively be split into two groups i.e. those which are resource specific and those which can have a global setting. Period related and room related constraints are resource specific i.e. settings can be established for each period and each room. This allows control of how resources would be used in constructing a solution. Values for these can be found after the introduction of periods and rooms in the datasets. All other soft constraints can be set relative to each other. These are referred to as global Settings.

Institutions may weight these soft constraints differently relative to one another in an attempt to produce a solution which is appropriate for their particular needs. This is known as building the 'Institutional Model' and is defined here as the Institutional Model Index. This is a relative weighting of the soft constraints which effectively provides a quality measure of the solution to be built. Within the datasets provided a number of variables are given with values.

It should be noted that when formulating a solution, it is common place for an institution to 'play' with various settings of soft constraints in an attempt to produce solutions which they judge satisfactory to all the end users. Indeed, this is why we have provided the soft constraint weightings in the data as opposed to the problem definition. In addition, including the weights in the data rather than expecting them to be hard coded into the solver allows us to set different weightings for each dataset. 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. Once again, this is motivated by our experience that different institutions do indeed have different weights, and so no one set would be completely useful. The hidden instances will have weights that we believe are reasonable; but competitors should not assume that such weights are necessarily similar (or different!) to those of the public instances.

The details provided here significantly add to the model of the problem commonly used within the research arena. Of course, how individuals judge that particular solutions are 'satisfactory' is an interesting open research problem and is currently being tackled in a number of novel ways e.g. Asmuni et al [30].

2.2 The Evaluation Function

Generally, the quality of a timetable is reflected by two values: the number of hard constraint violations (Distance to Feasibility), and the weighted sum of soft constraint violations. In order to compare two solutions, first we will look at the Distance to Feasibility, and the solution with the lowest value for this will be the winner[2]. If the two solutions are tied, we will then look at the number of soft constraint violations. The winner will be the solution that has the lowest value here.

The *Distance to Feasibility* is the total of the following numbers;

Conflicts: The number of occurrences of conflicting exams in the same period.

RoomOccupancy: The number of occurrences of more seating being required in any individual period than that available.

PeriodUtilisation: The number of occurrences when more time is required in any individual period than that available.

PeriodRelated: The number of occurrences when ordering requirements are not obeyed.

RoomRelated: The number of occurrences when room requirements not obeyed.

The resulting system is a special case of "constraint hierarchies" [31, 32]. In our case, we effectively have three levels of constraints

1. **required**: constraints that absolutely cannot be broken (e.g. the constraint that exams are not split between rooms)
2. **hard**: constraints whose violation leads to non-zero "distance to feasibility" (which is essentially just the objective function relevant to this level). Feasibility corresponds to satisfaction of all hard and required constraints.
3. **soft**: the usual sets of constraints that we prefer to satisfy but expect that it will not be possible to satisfy them all

with the implication that improving the solution at any level takes precedence over improving it at lower levels. We remark that although we have made a particular

[2]For the competition datasets, all solutions which are deemed acceptable or 'legal' should have a zero value for this measure.

choice in the competition for which constraints are hard and which required, it is quite possible that other choices are also useful.

Although the nature of the practical problem described here usually leads to feasibility being found quite easily, this is not necessarily always the case in practice. It was felt essential that this measure was included here to allow solution evaluation to be consistent across all tracks of the competition and in order to establish an evaluation method that can be built upon for the future. In practice, the examination timetabling problem dictates that there must be no hard constraint violations. When the situation arises where this is not the case, the incumbent timetabler would normally introduce another period or indeed room and set a high associated penalty. It is clear that this issue required detailed discussion and as gaining feasibility is not seen as a major issue for the competition datasets, the organizers feel such a discussion is outside the remit of the current report.

Within the competition, the solution will be classified based on the satisfaction of the soft constraints. On the website, in order to explain the calculation of the penalty a simple example is used allowing individual components of the overall penalty to be explained. This may be added to as competitors report issues which require clarification etc.. Trial datasets will also be introduced which will illustrate the calculations. The following provides a description of how each soft constraint is calculated. (We also give references to the appropriate sections in the mathematical formulation of section 3).

2.2.1 Two Exams in a Row

This calculation considers the number of occurrences where two examinations are taken by students straight after one another, i.e. back to back. Once this has been established, the number of students are totaled and multiplied by the number provided in the 'two in a row' weighting within the 'Institutional Model Index'. Note that two exams in a row are not counted overnight e.g. if a student has an exam the last period of one day and another the first period the next day, this does not count as two in a row. (See section 3.9.1).

2.2.2 Two Exams in a Day

In the case where there are three periods or more in a day, the number of occurrences of students having two exams in a day which are not directly adjacent, i.e. not back to back, are calculated. The total number is subsequently multiplied by the 'two in a day' weighting provided within the 'Institutional Model Index'. Therefore, two exams in a day are considered as those which are not adjacent i.e. they have a free period between them. This is done to ensure a particular exam placing within a solution does not contribute twice to the overall penalty. For example if Exam A and Exam B were in adjacent periods in the same day the penalty would be counted as part of the 'Two exams in a row penalty'. It should be noted that where the examination session contains days with 2 periods, this component of the penalty, although present for continuity, becomes superfluous. When this is the case this portion of the penalty will always be equal to zero. (See section 3.9.2).

2.2.3 Period Spread

This constraint allows an organisation to 'spread' an individual's examinations over a specified number of periods. This can be thought of as an extension of the two constraints previously described. Within the 'Institutional Model Index', a figure is provided relating to how many periods the solution should be 'optimised' over. The higher this figure, potentially the better the spread of examinations for individual students. In many institutions constructing solutions while changing this setting has led to timetables with which the Institution is much more satisfied. If, for example, PERIODSPREAD within the Institutional Model Index is set at 7, for each exam we count all the occurrences of enrolled students who have to sit other exams afterwards but within 7 periods i.e. the desired period spread. This total is added to the overall penalty. It should be noted that the occurrences here will have contributed to the penalty calculated for the 'two exams in a row' and 'two exams in a day' penalties. Although, a single occurrence within the solution is effectively penalised twice, it is often necessary due to, as indicated above, many institutions requiring certain spreads to be minimised as an indication of solution quality. (See section 3.9.3).

2.2.4 Mixed Durations

This applies a penalty to a Room and Period (not Exam) where there are mixed durations. The intention here is to try and ensure that exams occur together which are of equal length. In calculating this portion of the penalty, the mixed duration component of the 'Institutional Model Index' is calculated by the number of violations detected. (See section 3.9.4).

2.2.5 Larger Exams towards the beginning of the examination session

It is desirable that examinations with the largest numbers of students are timetabled at the beginning of the examination session. In order to take account of this the FRONTLOAD expression is introduced. Within the 'Institutional Model Index' the FRONTLOAD expression has three parameters e.g., 100, 30, 5. The first of these is the number of largest exams that are to be considered. Largest exams are specified by class size. If there are ties by size then exams occurring first in the data file are chosen. The second parameter is the number of last periods to take into account which should be ideally avoided. The third parameter is the penalty or weighting that should be added each time the constraint is violated. This allows the Institution to attempt to ensure that larger exams occur earlier in the examination session. This is popular in practice as exams with more students enrolled take longer to mark. (See section 3.9.5).

2.2.6 Room Penalty

It is often the case that organisations want to keep certain room usage to a minimum. As with the 'Mixed Durations' component of the overall penalty, this part of the overall penalty should be calculated on a period by period basis. For each period, if a room used within the solution has an associated penalty, the penalty for that room for that period is calculated by multiplying the associated penalty by the number of times the room is used. (See section 3.9.6).

2.2.7 Period Penalty

It is often the case that organisations want to keep certain period usage to a minimum. As with the 'Mixed Durations' and the 'Room Penalty' components of the overall penalty, this part of the overall penalty should be calculated on a period by period basis. For each period, the penalty is calculated by multiplying the associated penalty by the number of times the exams timetabled within that period. (See section 3.9.7).

3 Mathematical Programming Formulation

The formulation given here is intended to provide a mathematical definition of the problem. Accordingly, it was designed for compactness and (relative) clarity, and so was allowed to be non-linear. This means, in our experience with it, that it is not fit for solving the problems, however, it has been used it to validate solutions. The formulation is backed by implementation in Ilog's OPL/CPLEX[3] cross-checked with the web-based validator.

Naturally, anyone intending to use mathematical programming methods should not be biased by the formulation given here. For example, we also have various linear formulation(s) but will report on these at a later date.[4]

In an effort to render the formulation more readable we will follow the following conventions:

- Sets (of exams,etc) are upper case
- Parameters (quantities whose value is known or easily derivable from the input files) are lower case
- Variables (quantities whose value is to be determined by the search) are upper case.

[3]http://www.ilog.fr

[4]And in case the reader is wondering, no we don't have a magic one that solves all the instances to optimality!

- When quantities such as size are associated with two different types, such as exam size and room size, then, rather than increase the usual plethora of symbols, we'll indicate the "type" with a superscript. For example, s^E and s^R are used for exam and room sizes respectively.

3.1 Sets and Parameters

The following sets and parameters are either directly present in the input file, or are straightforward to derive from it.

3.1.1 Exam Related

E : set of exams

s_i : size of exam $i \in E$

d_i^E : duration of exam $i \in E$

f_e^E : a boolean that is 1 iff exam e is subject to the Front-Load penalties, 0 otherwise

D : the set of durations used $\cup_i d_i^E$

u_{id}^D : a boolean that is 1 iff exam i has "duration type" d which an index for set D. We call them "types" because the duration values don't matter, only whether they are equal.

In the competition input file format, the "FRONTLOAD" entry specifies 3 parameters. The first parameter is *"number of largest exams. Largest exams are specified by class size"* and is used to select which exams are subject to the front load penalty, that is, the exams for which $f_e^E = 1$. To be precise, the exams should be sorted by largest-first, with a secondary sort by earliest-index-first in the case of equal sized exams. The specified number of exams are then taken from the front of this sorted list and given $f_e^E = 1$, the remaining exams (if any) are given $f_e^E = 0$.

The duration type, u_{id}^D, is used for the no-mixed-duration penalty. For example, suppose all exams might have durations of either 120 or 180 minutes, then there would be two duration types, and we could have $d \in \{0, 1\}$ with $u_{i0}^D = 1$ iff exam i has duration 120, $u_{i1}^D = 1$ iff exam i has duration 180.

3.1.2 Students

S : set of students

Student enrollments are encoded by:

t_{is} : 1 iff student s takes (is enrolled in) exam i, 0 otherwise

3.1.3 Room Related

R : set of rooms

s_r^R : size of room $r \in R$

w_r^R : a weight that specifies the penalty for using room r

3.1.4 Period Related

P : set of periods

d_p^P : duration of period $p \in P$

f_e^P : a boolean that is 1 iff period p is subject to the FrontLoad penalties. The second parameter of the FRONTLOAD line in the input file is used to fix this. Starting with the latest period, the required number of periods are given $f_p^P = 1$.

w_p^P : a weight that specifies the penalty for using period p

y_{pq} : a boolean that is 1 iff periods p and q are in the same day

3.2 Period Related Hard Constraints

3.2.1 AFTER

H^{aft} : a set of pairs of exams.

For every pair $(e_1, e_2) \in H^{aft}$ exam e_1 must occur strictly after exam e_2

3.2.2 EXAM_COINCIDENCE

H^{coin} : a set of pairs of exams

For every pair $(e_1, e_2) \in H^{coin}$ exams e_1 and e_2 must occur in the same period (though not necessarily the same room)

3.2.3 EXCLUSION

H^{excl} : a set of pairs of exams

For every pair $(e_1, e_2) \in H^{excl}$ exams e_1 and e_2 must *not* occur in the same period.

3.3 Room Related Hard Constraints

3.3.1 EXCLUSIVE

H^{sole} : a set of exams

For every exam $e \in H^{sole}$, if exam $e1$ is assigned to period p and room r then e must be the sole occupier, i.e. no other exam can be assigned to both p and r. (Unless specified by an EXCLUSIVE rule, then, as standard in exam timetabling, exams are allowed to share rooms.)

3.4 Institutional Weights and Parameters

w^{2R} : weight for "two in a row"

w^{2D} : weight for "two in a day"

w^{PS} : weight for period spread (defaults to one as not currently specified in the input format, but included here for completeness)

w^{NMD} : weight for "No mixed duration"

w^{FL} : weight for the Front load penalty

The PERIODSPREAD line of the input format itself just specifies

g : the period spread, the preferred minimal "gap" between exams for a student

3.5 Variables

3.5.1 Primary Decision Variables

The binary (boolean) decision variables that fix the assignment are simply

$$X_{ip}^{P} = \text{ 1 if exam } i \text{ is in period } p\text{, 0 otherwise} \tag{1}$$

$$X_{ir}^{R} = \text{ 1 if exam } i \text{ is in room } r\text{, 0 otherwise} \tag{2}$$

3.5.2 Secondary Variables

By secondary variables we mean those whose values will be directly forced given any legal assignment to primary variables. They are used to write the constraints and to compute the objective function.

The penalties for violations of the various soft constraints are encoded as non-negative variables as follows

C_s^{2R} = "two in a row" penalty for student s

C_s^{2D} = "two in a row" penalty for student s

C_s^{PS} = "period spread" penalty for student s

C^{NMD} = "No mixed duration" penalty

C^{FL} = "Front-Load" penalty

C^P = "soft period" penalty

C^P = "soft room" penalty

Note: since these are all secondary and will be forced by the constraints then many of these secondary variables need not be forced to be integer but can be relaxed to be floats if desired.

We also remark that many of these variable are not really necessary, as they will be constrained to be equal to expressions that could be included directly into the objective. However, we keep them separate here for the purposes of clarity, and because their values should correspond to values given by the validator on the web.

We also use:

$$U_{dpr}^D = \text{ 1 if duration type } d \text{ is used in period } p \text{ and room } r\text{, 0 otherwise} \tag{3}$$

3.6 Objective and Constraints

3.6.1 Objective

Minimise

$$\sum_{s \in S} \left(w^{2R} C_s^{2R} + w^{2D} C_s^{2D} + w^{PS} C_s^{PS} \right) + w^{NMD} C^{NMD} + w^{FL} C^{FL} + C^P + C^R \tag{4}$$

Notice that there are no separate weights for the room and period penalties C^R and C^P as the associated weights were already included in their definitions. Of course, the problem is inherently multi-objective, but this weighted sum approach is used for simplicity.

One can also see that the objective represents a compromise between the various interested parties or stakeholders. Roughly speaking:

- the desire of student s is for a good individual timetable is represented by $w^{2R} C_s^{2R} + w^{2D} C_s^{2D} + w^{PS} C_s^{PS}$. Notice, that it would also be straightforward to encourage assignments that are fair between students by using the standard technique of including nonlinear terms, such as $(C_s^{2R})^2$, to suppress penalties above the average.
- interests of exam invigilators (and students) are represented by C^{NMD}

- the front load C^{FL} represents the desire of the exam markers to receive the largest exams as soon as possible so as to give more time for marking
- the estate management has interests, represented by terms such as C^P and C^R, in avoiding the (presumably expensive) use of some rooms and periods

One can also define the penalties for the entire set of students:

$$C^{2R} = \sum_{s \in S} C_s^{2R} \quad (5)$$

$$C^{2D} = \sum_{s \in S} C_s^{2D} \quad (6)$$

$$C^{PS} = \sum_{s \in S} C_s^{PS} \quad (7)$$

It might help potential competitors to know that, for the sequence constraints, the current web validator reports

- two-in-a-row penalty = $w^{2R} C^{2R}$
- two-in-a-day penalty = $w^{2D} C^{2D}$
- period spread penalty = $w^{PS} C^{PS}$

rather than the individual components.

Minimisation is subject to the following required and hard constraints, and the constraints defining the soft penalties.

3.7 "Required" Constraints

Any solution that violates these is rejected outright, and if these are violated it does not even get a "distance to feasibility" score.

Every exam is allocated to at most one room (exams cannot be "split"):

$$\forall\, i \in E. \quad \sum_{r \in R} X_{ir}^{R} \leq 1 \quad (8)$$

Every exam is allocated to at most one period:

$$\forall\, i \in E. \quad \sum_{p \in P} X_{ip}^{P} \leq 1 \quad (9)$$

3.8 "Hard" Constraints

Remember that "hard" constraints are those that might be relaxed if no solution can be found that satisfies them all (with the drawback of getting a non-zero "distance to feasibility" score). However, for the competition instances it is expected that solvers will be able to satisfy all the hard constraints, and so here we will simply enforce them as constraints, and not allow relaxations. That is, we do not encode the "distance to feasibility" that measures the extent to which the hard constraints are violated, but just force it to be zero.

Every exam is allocated to at least one room, and to at least one period:

$$\forall\, i \in E. \quad \sum_{r \in R} X^R_{ir} \quad \geq \quad 1 \tag{10}$$

$$\forall\, i \in E. \quad \sum_{p \in P} X^P_{ip} \quad \geq \quad 1 \tag{11}$$

It can depend on the context whether these ought to be "required" instead of merely "hard"; but we make them potentially relaxable for consistency with the other tracks.

Room capacities are always respected:

$$\forall\, p \in P.\ \forall\, r \in R. \quad \sum_{i \in E} s^E_i X^P_{ip} X^R_{ir} \quad \leq \quad s^R_r \tag{12}$$

Notice that this is not linear: Recall we just giving a mathematical formulation to define the problem, not a formulation that we expect to be effective for solving it.

Period durations are respected:

$$\forall\, p \in P.\ \forall\, i \in E. \quad d^E_i X^P_{ip} \quad \leq \quad d^P_p \tag{13}$$

In any period, any student is taking at most one exam:

$$\forall\, p \in P.\ \forall\, s \in S. \quad \sum_{i \in E} t_{is} X^P_{ip} \quad \leq \quad 1 \tag{14}$$

(This enforces the usual conflict matrix between exams.)

The hard period constraints are enforced by:

$$\begin{aligned} &\forall (i,j) \in H^{aft} \ \ \forall p, q \in P, \text{ with } p \leq q \\ &\qquad X^P_{ip} + X^P_{jq} \quad \leq \quad 1 \end{aligned} \tag{15}$$

$$\begin{aligned} &\forall (i,j) \in H^{coin} \ \ \forall p \in P, \\ &\qquad X^P_{ip} = X^P_{jp} \end{aligned} \tag{16}$$

$$\begin{aligned} &\forall (i,j) \in H^{excl} \ \ \forall p \in P, \\ &\qquad X^P_{ip} + X^P_{jp} \ \leq \ 1 \end{aligned} \tag{17}$$

The hard room constraints are enforced by

$$\begin{aligned} &\forall i \in H^{sole} \ \ \forall j \in E, j \neq i \ \ \forall p \in P,\ \forall r \in R, \\ &X^P_{ip} + X^R_{ir} + X^P_{jp} + X^R_{jr} \leq 3 \end{aligned} \tag{18}$$

3.9 "Soft" Constraints

3.9.1 Two in a Row

If a student s is enrolled into two distinct exams i and j, and j occurs on the same day in the period immediately after the period used for i, then the penalty C_s^{2R} receives an increment of 1. Hence,

$$C_s^{2R} = \sum_{\substack{i,j \in E \\ j \neq i}} \sum_{\substack{p,q \in P \\ q=p+1 \ \& \ y_{pq}=1}} t_{is} t_{js} X_{ip}^P X_{jq}^P \qquad (19)$$

Notice that there is no double counting. The condition on the periods is $q = p + 1$ rather than $|p - q| = 1$ and so the condition $j \neq i$ is needed rather than $j > i$. That is, we separately capture the cases 'i is before j' and 'i is after j'.

3.9.2 Two in a Day

If a student s is enrolled into two distinct exams i and j and these exams occur in non-consecutive periods on the same day, then the penalty C_s^{2R} receives an increment of 1. Hence,

$$C_s^{2D} = \sum_{\substack{i,j \in E \\ j \neq i}} \sum_{\substack{p,q \in P \\ q>p+1 \ \& \ y_{pq}=1}} t_{is}\, t_{js}\, X_{ip}^P\, X_{jq}^P \qquad (20)$$

This is the same as for two-in-a-row except the $q = p + 1$ condition changed to $q > p + 1$.

3.9.3 Period Spread

If a student s is enrolled into two distinct exams i and j and these exams occur in distinct periods such that j is after i but is within the gap g, then the penalty C_s^{PS} receives an increment of 1. Again, double counting is prevented by putting a time order on the exams that contribute. Hence,

$$C_s^{PS} = \sum_{\substack{i,j \in E \\ j \neq i}} \sum_{\substack{p,q \in P \\ p<q \leq p+g}} t_{is} t_{js} X_{ip}^P X_{jq}^P \qquad (21)$$

This is the same as for two-in-a-row or day except that the conditions on the two periods p and q changed to $q \in [(p + 1), \ldots, (p + g)]$.

3.9.4 Non-Mixed Durations

We need to force U_{dpr}^D to be non-zero whenever some exam with duration type d uses period p and room r:

$$\forall d \in D.\ \forall i \in E, \text{ with } u_{id} = 1.\ \forall p \in P.\ \forall r \in R.$$

$$U_{dpr}^D \geq X_{ip}^P + X_{ir}^R - 1 \qquad (22)$$

The period-room pair *pr* receives a non-negative penalty, C^{NMD}_{pr}, which is the maximum of zero and the excess above one of the total number of durations types assigned to it:

$$\forall\, p \in P.\ \forall\, r \in R.$$
$$1 + C^{NMD}_{pr} \geq \sum_{d \in D} U^{D}_{dpr} \qquad (23)$$
$$C^{NMD}_{pr} \geq 0 \qquad (24)$$

The overall penalty is

$$C^{NMD} = \sum_{p \in P} \sum_{r \in R} C^{NMD}_{pr} \qquad (25)$$

The minimisation in the overall objective will force U^{D}_{dpr} and C^{NMD}_{pr} to be the intended minimal values consistent with the assignment.

3.9.5 Front Load

$$C^{FL} = \sum_{i \in E} \sum_{p \in P} f^{E}_{i} f^{P}_{p} X^{P}_{ip} \qquad (26)$$

3.9.6 Soft Period Penalties

$$C^{P} = \sum_{p \in P} \sum_{i \in E} w^{P}_{p} X^{P}_{ip} \qquad (27)$$

3.9.7 Soft Room Penalties

$$C^{R} = \sum_{r \in R} \sum_{i \in E} w^{R}_{r} X^{R}_{ir} \qquad (28)$$

4 Conclusion and Discussion

Information is presented here on a formulation of the examination timetabling problem that is common to many institutions. This track introduces a practical formulation of the problem which, organisers believe, significantly adds to current research and provides a firm basis for future efforts in the area. In relation to this, the following points are made.

We do not consider minimising the number of periods as part of this formulation as, in our experience, educational institutions manage the process by using set times for the examination session. That is not to say of course that this is not a major issue in relation to planning examination sessions. It is acknowledged that a full investigation and explanation of 'Distance of feasibility" is required if the formulation provided here is to be useful for such purposes.

From experience we have found that, in general, gaining feasibility is not as important an issue as in some cases of course timetabling. That is not to say, of course, that competitors may have difficulty satisfying all the hard constraints within the competition time limit requirement. If this is the case, and competitors experience difficulty in finding feasibility, we will decide how to deal with 'non feasible' solutions. It is pointed out here that a competition time limit is essential to allow comparison of the techniques used. In practice, it can be argued that the need for such a time limit is not required as organisations are often happy to allow for longer running times in search for 'better' solutions. That being said, it is often the case that due to many changes having to be made during solution construction [2], an individual within an institution requires the ability to generate many solutions quickly after making various amendments to the underlying data. This style of solution construction will be well served by the techniques developed as part of this track. Please also see the associated Curriculum CTT technical report for a discussion of this issue.

Although a 'weighted sum' evaluation function is not ideal e.g. it may have adverse side effects for certain individual students, it is the chosen method here due to the ease of implementation for purposes of comparison. It is hoped that the interest generated by efforts here will lead to true multi-objective evaluation of potential solutions. In particular, we specifically decided to include the weights in the data format itself rather than solvers having to hard code them. This at least ought to easily allow variations of the weights so as to explore multi-objective properties. Also, it is unlikely that every institution would have the same weights, and so fixing them in the solver seems inappropriate.

References

[1] Metaheuristics Network. International timetable competition 2002, 2003. http://www.idsia.ch/Files/ttcomp2002/ Organised by the Metaheuristics network, http://www.metaheuristics.net/ and PATAT 2002 http://www.asap.cs.nott.ac.uk/patat/patat02/patat02.shtml, accessed 2 July 2007.

[2] B. G. C. McCollum. (PLENARY) university timetabling: Bridging the gap in university timetabling. In *PATAT'06, Proceedings of the 6th International Conference on the Practice and Theory of Automated Timetabling*, pages 15–35, 2006. ISBN 80-210-3726-1. To be published in the forthcoming LNCS post conference proceedings.

[3] R. Qu, E. K. Burke, B. McCollum, L. T. G. Merlot, and S. Y. Lee. The state of the art of examination timetabling. Technical Report NOTTCS-TR-2006-4, School of CSiT, University of Nottingham., 2006. Technical Report.

[4] D. de Werra, A. S. Asratian, and S. Durand. Complexity of some special types of timetabling problems. *Journal of Scheduling*, 5:171–183, 2002.

[5] E. Tsang, P. Mills, and R. Williams. A computer aided constraint programming system. In *The 1st International Conference on the Practical Application of Constraint Technologies and Logic Programming (PACLP)*, pages 81–93, 1999.

[6] L. Di Gaspero and A. Schaerf. Tabu search techniques for examination timetabling. In E. K. Burke and W. Erben, editors, *Practice and Theory of Automated Timetabling: Selected Papers from the 3rd International Conference.*, 2001.

[7] L. Di Gaspero. Recolour, shake and kick: A recipe for the examination timetabling problem. In E. K. Burke and P. De Causmaecker, editors, *Proceedings of the 4th Interna- tional Conference on Practice and Theory of Automated Timetabling.*, pages 404–407, KaHo St.-Lieven, Gent, Belgium, 2002.

[8] G. M. White and B. S. Xie. Examination timetables and tabu search with longer-term memory. In E. K. Burke and W. Erben, editors, *Practice and Theory of Automated Timetabling: Selected Papers from the 3rd International Conference.*, 2001.

[9] L. Paquete and T. Stützle. Empirical analysis of tabu search for the lexicographic optimization of the examination timetabling problem. In E.K. Burke and P. De Causmaecker, editors, *Proceedings of the 4th International Conference on Practice and Theory of Automated Timetabling.*, 2002.

[10] Edmund K. Burke, Yuri Bykov, James Newall, and Sanja Petrovic. A time-predefined local search approach to exam timetabling problems. *IIE Transactions*, 36(6):509–528, Jun 2004.

[11] L. T. G. Merlot, N. Boland, B. D. Hughes, and P. J. Stuckey. A hybrid algorithm for the examination timetabling problem. In *Practice and Theory of Automated Timetabling: Selected Papers from the 4th International Conference.*, volume 2740 of *Springer Lecture Notes in Computer Science*, pages 207–231, 2003.

[12] J. Thompson and K. Dowsland. A robust simulated annealing based examination timetabling system. *Computers & Operations Research*, 25, 1998.

[13] Peter Ross, Dave Corne, and Hugo Terashima-Marn. The phase transition niche for evolutionary algorithms in timetabling. In Edmund K. Burke and Michael A. Trick, editors, *Selected papers from the First International Conference on the Theory and Practice of Automated Timetabling (PATAT 95)*, volume 1153, pages 309–324. Lecture Notes in Computer Science, Springer-Verlag, NY, 1996.

[14] T. Wong, P. Cote, and P. Gely. Final exam timetabling: A practical approach. In *IEEE Canadian Conference on Electrical and Computer Engineering (CCECE 2002).*, volume 2, pages 726– 731, 2002.

[15] Edmund K. Burke, James Newall, and Rupert F. Weare. A memetic algorithm for university exam timetabling. In Edmund K. Burke and Peter Ross, editors, *The Practice and Theory of Automated Timetabling*, volume 1153 of *Lecture Notes in Computer Science*, pages 241–250. Springer, 1996.

[16] K. A. Dowsland and J. Thompson. Ant colony optimization for the examination scheduling problem. *on scheduling problem.*, 56:426–438, 2005.

[17] Z. Naji Azimi. Hybrid heuristics for examination timetabling problem. *Applied Mathematics and Computation*, 163(2):705–733, 2005.

[18] Edmund K. Burke and James Newall. Solving examination timetabling problems through adaptation of heuristic orderings. *Annals of operations Research*, 129:107–134, 2004.

[19] Edmund K. Burke, Moshe Dror, Sanja Petrovic, and Rong Qu. *The Next Wave in Computing, Optimization, and Decision Technologies*, chapter Hybrid Graph Heuristics within a Hyper-heuristic Approach to Exam Timetabling Problems, pages 79–91. Springer, 2005.

[20] Edmund K. Burke, Sanja Petrovic, and Rong Qu. Case based heuristic selection for timetabling problems. *Journal of Scheduling*, 9(2):115–132, 2006.

[21] Edmund K. Burke, Barry McCollum, Amnon Meisels, Sanja Petrovic, and Rong Qu. A graph-based hyper-heuristic for educational timetabling problems. *European Journal of Operational Research*, 176:177–192, 2007.

[22] P. David. A constraint-based approach for examination timetabling using local repair techniques. In E. K. Burke and M. W. Carter, editors, *Practice and Theory of Automated Timetabling: Selected Papers from the 2nd International Conference.*, volume 1408 of *Springer Lecture Notes in Computer Science*, pages 169–186, 1998.

[23] T. A. Duong and K. H. Lam. Combining constraint programming and simulated annealing on university exam timetabling. In *Proceedings of the 2nd International Conference in Computer Sciences, Research, Innovation & Vision for the Future (RIVF2004)*, pages 205–210, Hanoi, Vietnam, February 2004.

[24] E. K. Burke, A. J. Eckersley, B. McCollum, S. Petrovic, and R. Qu. Hybrid variable neighbourhood approaches to university exam timetabling. Technical Report NOTTCS-TR- 2006-2, School of CSiT, University of Nottingham, 2006.

[25] S. Ahmadi, R. Barone, P. Cheng, P. Cowling, and B. McCollum. Perturbation based variable neighbourhood search in heuristic space for examination timetabling problem. In *Proceedings of Multidisciplinary International Scheduling: Theory and Applications (MISTA 2003)*, pages 155–171, Nottingham, August 13-16,, 2003. ISBN: 0-9545821-2-8.

[26] Edmund K. Burke, Emma Hart, Graham Kendall, James Newall, Peter Ross, and Sonia Schulenburg. *Handbook of Meta-Heuristics*, chapter Hyper-Heuristics:

An Emerging Direction in Modern Search Technology, pages 457–474. Kluwer, 2003.

[27] P. Ross, J.G. Marin-Blazquez, and E. Hart. Hyper-heuristics applied to class and exam timetabling problems. In *Proceedings of the 2004 Congress on Evolutionary Computation (CEC2004)*, pages 1691–1698, 2004.

[28] Edmund K. Burke, Sanja Petrovic, and Rong Qu. Case-based heuristic selection for examination timetabling. In *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution And Learning (SEAL 2002)*, pages 277–281, Singapore, Nov 18-22 2002.

[29] M. W. Carter, G. Laporte, and S. Y. Lee. Examination timetabling: Algorithmic strategies and applications. *Journal of Operational Research Society*, 47(3):373–383, 1996.

[30] Hishammuddin Asmuni, Edmund K. Burke, Jon M. Garibaldi, and Barry McCollum. A novel fuzzy approach to evaluate the quality of examination timetabling. In *Proceedings of the 6th International Conference on the Practice and Theory of Automated Timetabling (PATAT 2006)*, pages 82–102, Brno, Czech Republic, August 30th-September 1st 2006.

[31] A. Borning, R. Duisberg, B. Freeman-Benson, A. Kramer, and M. Woolf. Constraint hierarchies. In Norman Meyrowitz, editor, *Proceedings of the Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA)*, volume 22, pages 48–60, New York, NY, 1987. ACM Press.

[32] Alan Borning, Bjorn Freeman-Benson, and Molly Wilson. Constraint hierarchies. *Lisp and Symbolic Computation*, 5(3):223–270, Sep 1992.