

Setting the Research Agenda in Automated Timetabling: The Second International Timetabling Competition

B. McCollum, P. McMullan

School of Electronics, Electrical Engineering and Computer Science, Queen's University, Belfast, N. Ireland, BT7 1NN, b.mccollum@qub.ac.uk

B. Paechter

Centre for Emergent Computing,
Napier University, Edinburgh, EH10 5DT, Scotland, b.paechter@napier.ac.uk

R. Lewis

Cardiff Business School, Prifysgol Caerdydd / Cardiff University, Cardiff, Wales CF10 3EU
LewisR9@cardiff.ac.uk

A. Schaerf, L. Di Gaspero

Department of Electrical, Management and Mechanical Engineering, University of Udine,
33100 Udine, Italy, schaerf@uniud.it

A. J. Parkes, R. Qu, E. Burke

School of Computer Science, University of Nottingham, Jubilee Campus
Nottingham, NG8 1BB, UK, a.parkes@cs.qub.ac.uk

The 2nd International Timetabling Competition (ITC2007) was announced on the 1st August 2007. Building on the success of the first, this competition aimed to further develop interest in the area of educational timetabling while providing researchers with models of the problems faced which incorporate an increased number of real world constraints. A main objective of the competition was that conclusions drawn would further stimulate debate within the widening timetabling research community. The overall aim of the competition was to create better understanding between researchers and practitioners by allowing emerging techniques to be trialed and tested on real world models of timetabling problems. The competition was divided into three tracks to reflect the important variations which exist within educational timetabling within Higher Education. As these formulations incorporate an increased number of 'real world' issues, it is anticipated that the competition will set the research agenda within the field. After finishing on the 25th January 2008, final results of the competition are to be made available in May 2008. Along with background to the competition, the tracks are described here together with initial results for the datasets released.

Key words: artificial intelligence; timetabling; competition; benchmarks; results

1 Introduction

Timetabling within a university context has long been recognised as difficult from both a theoretical and practical perspective [Schaerf (1999)]. Whether it be timetabling courses or examinations, much effort is spent producing solutions which are both workable and of a sufficiently high quality. In recent years, timetabling has been the subject of a competition. In 2002, the European Metaheuristic Network organized the First International Timetabling Competition (ITC2002). Based on a specific problem model, the formulation presented contained characteristics of the course timetabling problem found in many Universities. Details on applied techniques and results can be found at the webpage <http://www.idsia.ch/Files/ttcomp2002/>. More recently, this formulation has become somewhat of a standard within the research area with many researchers using it and the associated generated datasets within their work [Abdullah et al. (2007), Chiarandini et al (2006), Di Gaspero and Schaerf (2006), Kostuch (2005), Lewis et al. (2007b)]. ITC2002 has therefore been successful in generating common ground for cross-fertilisation of ideas for research groups within the timetabling community.

The Second International Timetabling Competition (ITC2007), which opened on August 1st 2007, followed the main ethos of the first in further providing a basis on which research in timetabling can progress. Detailed information can be found at the competition website (<http://www.cs.qub.ac.uk/itc2007>). An important aim of this competition was the generation of new approaches for timetabling problems described by attracting users from all areas of research, specifically encouraging multi-disciplinary approaches. An additional important aim was to narrow the gap which exists between research and practice within this important area at the intersection between operational research and artificial intelligence [McCollum (2007a)]. To this end, various formulations of the timetabling problems encountered within educational institutions based on a ‘real world’ perspective were introduced. In providing these formulations, it was considered important to balance the inclusion of all known associated specific details with the required competitive element. Importantly, the success of the first international timetabling competition was built upon by introducing significantly more depth and complexity in not one but three distinct variations of the timetabling problem, called tracks. Competitors were encouraged to enter one, two or all of the tracks.

Although inevitably overlap exists, these tracks represented distinct problems within the area of educational timetabling both from a research and practical perspective. From a research perspective, it was felt by the organisers that this division was important because it provided a framework for capturing the main types of educational timetabling research currently taking place within the academic community. From a practical perspective, the tracks also provided more details of the models experienced in real world situations.

The three tracks considered in the competition cover the main formulations of both examination (Exam TT) and course timetabling (CTT) problems. As for the course timetabling problem, this area has been subdivided into two tracks. Both of these tracks are distinct and represent methods of course timetable construction which are used in various forms within many institutions. Namely post enrolment course timetabling (PostEnroll CTT) and curriculum based course timetabling (Curriculum CTT).

Technical reports (McCollum et al (2007b), Lewis et al (2007), Di Gaspero et al. (2007)) are available on the competition web site for each track detailing all aspects of the formulation offered. In this work, following a description of the main aspects of the competition, an overview of each track is provided along with results on the datasets released as part of the competition.

2 Competition Rules

The Competition Rules were divided into three headings; General (rules 1-4), Solution Creation (rules 5-13) and Adjudication (rules 14-18). For all Tracks the following rules were stipulated¹;

Rule 1: This competition seeks to encourage research into automated timetabling methods, and to offer prizes to the most successful methods in particular tracks. It is the spirit of these rules that is important, not the letter. With any set of rules for any competition it is possible to work within the letter of the rules but outside the spirit. The organisers ask that you please don't do this. It's not fair, it's not good science, and it will result in disqualification.

Rule 2: The organisers reserve the right to disqualify any participant from the competition at any time if the participant is determined by the organisers to have worked outside the

¹ In order to preserve the exact meaning, the rules are quoted directly from the website and therefore are in present tense.

spirit of the competition rules. The organisers' decision is final in any matter. Decisions will be made democratically by the organisers with the Chair having the casting vote in the case of ties.

Rule 3: Specific rules regarding individual tracks can be found at the appropriate pages on the official web site. Where there is any conflict, this set of general rules takes precedence.

Rule 4: The organisers reserve the right to change the rules at any time and without notice. Any change of rules will be accompanied by a general email to all participants.

Rule 5: The competition has an opening day and a deadline when all submissions must be uploaded. These deadlines are absolute and no extensions will be given under any circumstances, as to do so would be unfair to other participants.

Rule 6: Participants have to implement an algorithm to tackle the problem on a single processor machine; they can use any programming language.

Rule 7: The goal is to produce timetables in which a number of hard constraints are satisfied (i.e. feasible timetables) and to minimize the number of broken soft constraints. Where feasibility cannot be reached, infeasible solutions can be provided according to the rules of the specific track. Competitors should refer to the information associated with each track for further specifics.

Rule 8: Instances of different size and type will appear on the web site from the opening day. Two weeks before the deadline more instances will be placed on the web. A third set of datasets will be used to internally rank the top competitors. The datasets are therefore classified as Early Instances, Late Instances and Hidden Instances. Competitors should refer to the information associated with each track for further specifics on datasets. The Hidden Instances will be released after the competition is closed.

Rule 9: Participants have to benchmark their machine with the program provided in order to know how much time they have available to run their program on their machines.

Rule 10: The algorithms should take as input a problem file in the format described, and produce as output a solution in the allowed CPU time. Obviously the algorithm should not take account of additional hard coded knowledge about the instance (e.g. introducing instance specific heuristics). The same version of the algorithm must be used for all instances. That is, the algorithm should not "know" which instance it is solving – while your particular algorithm might analyse the problem instance and set parameters accordingly, it

should not “recognise” the particular instance. The programmer should not set different parameters for different instances although if the program is doing this automatically then this is acceptable.

Rule 11: The algorithm can be either deterministic or stochastic. In both cases, participants must be prepared to show that these results are repeatable in the given computer time. In particular, the participants that use a stochastic algorithm should code their program in such a way that the exact run that produced each solution submitted can be repeated (by recording the random seed, etc.). They can try several runs to produce each submitted solution (each with the allowed computer time), but they must be able to repeat the run for any solution submitted.

Rule 12: Participants should submit for each instance (Early and Late) the best score found by their algorithm in the specified computer time, by uploading it onto the web site. Competitors should refer to the information associated with each track for further specifics.

Rule 13: Participants should also submit a concise and clear description of their algorithm, so that in principle others can implement it. A template will be made available one month before the end date for this purpose. This is a fundamental part of a competitors' submission.

Rule 14: For each track, a set of 5 finalists will be chosen after the competition deadline. Ordering of competitors will be based on the scores provided on the early and late instances. The actual list will be based on the ranks of solvers on each single instance. The mean average of the ranks will produce the final place list.

Rule 15: Based on the place list a set of top solvers, the finalists, will be asked to provide the executable that will be run and tested by the organisers. The finalists' solver will be rerun by the organisers on all instances (including the Hidden ones). It is the responsibility of the competitor to ensure all information is provided to enable the organisers to recreate the solution. The solver submitted by the finalist should require as command-line arguments, input and output file names and, for stochastic solvers only, the random seed. If appropriate information is not received or indeed the submitted solutions cannot be recreated, another finalist will be chosen from the original competitors.

Rule 16: Finalists' eventual place listings will be based on the ranks on each single instance for a set of trials on all instances (including the hidden ones).

Rule 17: In some circumstances, finalists may be required to show source code to the organisers. This is simply to check that they have stuck to the rules and will be treated in the strictest confidence.

Rule 18: Entries from participating organising partners will not be permitted. However, results from participants who choose to work on the problems will be presented for comparison.

3 Differences from ITC2002

The first International Timetabling Competition ran in 2002. An overview of the techniques submitted is given in the following (Abdullah et al.(2007), Lewis et al. (2007)). The following sections highlights the main areas of difference between the two competitions.

As detailed earlier, the first innovation introduced by ITC2007 was the subdivision of three tracks in order to better cover the main formulations of the field of educational timetabling problems. The second innovation aimed at bridging the gap between research and practice: the competition introduced a significant degree of complexity in all tracks meaning the new formulations employed are closer (in more aspects, although not all) to those of ‘real world’ problems [McCollum07a]. The third main innovation was with regards to feasibility of solutions. It is common in the area of timetabling research to regard a solution is as feasible if all the hard constraints are satisfied. Often, in the real world, cases arise where solutions are judged as good even though feasibility is not attained. In ITC2002, in order to avoid the problems of measuring degrees of infeasibility and comparing these with number of broken soft constraints, it was decided that only feasible solutions were accepted. Consequently, all problem instances were constructed so that feasibility was not overly difficult to obtain.

Participants of ITC2007 who reached only infeasible solutions for some instances were able to submit their solution, although it was pointed out that all instances are guaranteed to have at least a feasible one. In order to compare different solvers in cases of infeasible solutions for some instances, the organisers used an evaluation based on ranking of solutions on each instance, rather than on the actual scores (see Section 5). Due to this scoring based on rankings, an infeasible solution on one instance did not necessarily prejudice the overall performance of the participant. In addition, instances for the competition were selected from a larger set of cases, without the limitation of ‘easy feasibility’.

Finally, in ITC2002 the ranking was based on the solution provided by the participants with checks made on unseen instances on the organiser's machines. This meant that the competition was open to the criticism that participants could take advantage of the Mongolian Horde approach [Schaerf and Di Gaspero (2007)]: "Run as many trials as you can and report only the best of all of them" – although in practice the checks on unseen instances showed that this was not a problem.. In ITC2007, the re-running of finalist solvers on organisers' machine (with new seeds) and the use of hidden instances in the rankings were introduced to improve the situation. Although not used for the place-list, the organisers plan to use principled statistical tools to analyse in more detail the performance of the solvers, especially for the stochastic ones.

4 Benchmarking

As with the first competition, the winner of each track was chosen based on the quality of the solutions produced by the proposed technique within a specific, pre-imposed time limit, measured in elapsed time (see Section 5 for further detail on winner designation). Note that although conforming to a time limit might not always be an important constraint in real world timetabling, using one does allow us to introduce a fair competitive element to the competition (which, it was hoped, would help to attract more researchers to enter).

In order to allocate a time limit to each of the competitors, a benchmarking program was created and distributed. This program was only suitable for individual, single processor machines, not for specialist parallel machines or clusters. Competitors would execute this on their own machines, whereupon the program would set about performing a number of computational operations of the sort involved in timetabling. When the program halted, the program then considered how long it had taken to execute, and used this figure in order to calculate an appropriate time limit for the machine at hand. Obviously, the speed of the benchmark program (and resultant time limit) on an individual machine depends on a number of factors including the memory, the type of processor, the clock speed, and the operating system. Note that in providing this benchmark program, it was not possible to provide perfectly equitable benchmarks across the various platforms, types of processor, and so on, and we acknowledge that the benchmark may well have been "kinder" to some people than others. However, one way that we attempted to counter this potential discrepancy, was by running all of the eventual finalists' algorithms on our own "benchmark ma-

chine” therefore creating more of a level playing field in the final stages of the competition.

The reason why it was decided to have a fixed running time was mainly to remove one degree of variability from the scoring system. We anticipate that future competitions will take into account in some principled way the trade-off between solution quality and running times. For the selection of the fixed amount of running time, the key question is concerned with establishing a realistically feasible running time for the actual timetabling. Given that the timetabling from both a course and exam perspective is performed usually a few times a year, one might think that a running time much longer than the 5-10 minutes granted for the competition would also be reasonable. In practical cases however, as many researchers and practitioners have pointed out, the solution of a real case is an interactive process, during which it is necessary to solve a large number of instances. In fact, constraints and objectives are manually adjusted between runs of a working session for one single case (for various reasons: what-if scenarios, last minute changes, etc.). As a rule of thumb, a running time longer than a few minutes makes the process very tiresome and difficult for the human operator.

5 Designation of the Winners

The algorithms of the various competitors were judged on the basis of the results of their algorithms on a set of benchmark instances. Three sets of data were used for each track: the ‘Early’ set, the ‘Late’ set, and the ‘Hidden’ set. Early datasets were released at the beginning of the competition with a further Late set, being released two weeks before the deadline. Hidden datasets were used by the organisers for each track to make judgments relating to the finalists and for choosing eventual winners of each track.

The adjudication process was divided into two phases. For each track the five best performing algorithms were classified as the ‘Finalists’ for that track. These were chosen based on the results provided by the competitors and in accordance with the competition rules and ranking criteria. Finalists for each track were asked to provide details and executable versions of their techniques to allow the organisers to run and test them on all datasets, including the hidden datasets.

The following illustrates how the ‘Finalists’ for each track were chosen. Let m be the total number of early and late instances and k the number of participants that produce a

solution for all m instances. Let X_{ij} be the result supplied (and verified) by participant i for instance j . An important feature to note here is that in all three competition tracks, the quality of a timetable is indicated by two separate values: (1) the “Distance to Feasibility”, and (2) the “Soft Cost”; which indicate the level at which the hard and soft constraints, respectively, are adhered to (see [the tech reps] for specific details of how these values are calculated in each case). Therefore, each result X_{ij} is a pair (d,s) composed of the distance to feasibility d and the score of the objective function s . Candidate solutions are then compared by examining these values in the following way. First, the distance to feasibility d is considered, and the solution that is seen to have the lowest value for this is considered the winner. However, if two or more solutions are equal here, then the winner is then chosen as the solution that has the lowest soft cost amongst these. Note that this method of using a pair of values means that solution quality is a type of ordinal data, meaning that we are able to rank solutions, but we cannot calculate a precise distance between them (except, of course, when we have solutions that have an equal distance to feasibility).

To compare the algorithms, the matrix X of results is transformed into a matrix of ranks assigning to each X_{ij} a value R_{ij} from 1 to k . That is, for each instance j , the supplied results $X_{1j}, X_{2j}, \dots, X_{kj}$ are compared with each other and the rank 1 is assigned to the best observed timetable, the rank 2 to the second smallest, and so on to the rank k , which is assigned to the worst timetable for instance i . In case of ties average ranks are assigned.

The organisers checked the runs of the candidate finalist with the submitted seed to make sure that the submitted runs were repeatable. In ascertaining the final winner, the same evaluation process was repeated for the finalists with the following differences: All instances, including hidden ones, were used and the solvers run by the organizers. In addition, for each instance, the organisers ran ten independent trials with seeds chosen at random. For each trial, the organisers computed the ranks and averaged them for all trials on all instances. The winner of each track was the one with the lowest mean rank.

6 Results and Rational

In this section we will now outline the three problem models used in the competition. We will also present results gained by our own solvers for each of the released problem

instances. (Specific details of these solvers are given below). The purpose of running the organisers' solvers on the instances and releasing these results is twofold: On the one hand, the organisers had to ensure that a feasible solution could be reached for all instances, thus the results are a proof of existence of a feasible solution. On the other hand, from the optimisation point of view, these results form a sort of baseline (upper-bound) that can be exploited in future comparisons.

In all three cases these algorithms are stochastic, meaning that different runs are generated with different random seeds. In the following sections, statistics calculated from multiple runs are therefore presented in order to provide information on the types of solutions these algorithms are able to produce in general. Using our solvers we performed 51 runs on each instance using a different seed in each case and, for each run, calculated the distance to feasibility and soft cost of the best solution found. From these, we then identified the worst, median, and best solutions that were obtained in the 51 runs, together with the upper and lower quartiles. Note that by using 51 runs here (instead of, say, 50), we are able to calculate the quartiles and median without the need for interpolation, which would be inappropriate here.

7 The Examination Timetabling Problem

From a practical perspective, much work is required in establishing a generic examination timetable model which is applicable across a wide range of scenarios. The problem formulation proposed here significantly adds to current models used within research and provides a basis for further real world constraints to be described. 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 some 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 [Carter et al. (1996)]. These benchmarks and the problems associated with them are discussed in more detail elsewhere [Qu et al. (2007)]. 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.

7.1 The Problem Model

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. For detailed information the interested reader is referred to the examination technical report (McCollum et al 2007a). Importantly, 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.

From experience we have found that, in general, gaining feasibility within examination timetabling is not as important an issue as with some cases of course timetabling. That is not to say, that in relation to the competition datasets researchers 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 (though this is not anticipated as the datasets were chosen from those institutions which had a relatively low clash matrix conflict density) a method must be used to decide how to deal with these ‘non feasible’ solutions. It is pointed out here that a competition time limit is essential to allow comparison of the techniques used. How to implement a timetable solution when feasibility is not found is always, in our experience, decided in by the institution. Actions to be taken include, extending the session, introducing another room, allowing more capacity within particular rooms, holding students over the lunch break etc.. The current competition does not deal with these institutional heuristics and therefore expected researchers to gain feasibility. Indeed, datasets were chosen on this basis.

The following describes the problem formulation. An examination session is made of a number of periods over a specified length of time, i.e., examination session. Period lengths within which a set of examinations can be allocated are provided. A set of students are enrolled on individual examinations, where each individual student may be enrolled on a

number of exams. A set of rooms with individual capacities are provided. Hard and Soft Constraints are provided, the Soft Constraints having an associated value contributing to a penalty, if violated. Details including a 'weighting' of particular soft constraints are provided within the Institutional Model Index.

A feasible timetable is one in which all examinations have been assigned to a period and room so that 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;
 - Period related hard constraints e.g. Exam_A After Exam_B;
 - Room related hard constraints e.g. Exam_A must use Room 101.

A candidate timetable is penalised for each occurrence of the following soft constraints:

- Two exams in a row;
- Two exams in a day;
- Specified spread of examinations;
- Mixed duration of examinations within individual periods;
- Larger examinations appearing later in the timetable;
- Period related soft constraints;
- Room related soft constraints;

These constraints can effectively be split into two groups; those which are resource specific and those which can have a global setting. The following resource specific constraints can be set for each period and each room i.e. period related soft constraints and room related soft constraints. This allows control of how resources would be used in constructing a solution. The following Global Setting constraints can be set relative to each other: Two exams in a row, Two exams in a day, Specified spread of examinations, Mixed duration of examinations within individual periods, larger examinations appearing later in the timetable. 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 Insti-

tutional 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. More information can be found in the Input Format section on the competition web site.

7.2 Description of Algorithm

The method used to solve the examination scheduling problem involves a two-phase construction and improvement technique. Construction is implemented with an Adaptive (Squeaky-Wheel) ordering heuristic [Burke et al. (2004)], which utilises a weighted order list of the events to be scheduled based on individual soft penalties and ‘difficulty to schedule’ penalties. This heuristic terminates when a feasible solution has been achieved, at which point the improvement phase begins. Improvement is an extension to the Great Deluge algorithm which has been employed as an improvement to Simulated Annealing in scheduling problems [Dueck (1990)]. The extended technique employs a reheat mechanism with guided parameters to avoid local optimum and attempt to provide a much wider search of the solution neighbourhood. Two basic heuristics are employed within the local search, i.e. random move and swap, while maintaining feasibility and attempting to continually improve the solution or keep the evaluation within a given boundary limit. The application of these simple heuristics allow the algorithm to explore more neighbourhoods efficiently in a limited time environment (such as with a time-limited competition), as computation time is not generally spent on making specific choices for moves or swaps. This technique has achieved considerably better results than those currently published for the standard course timetabling benchmarks [Abdullah et al.(2007)]. A full description of the algorithm along with the results achieved is described in [McMullan (2007)].

Table 1.0 lists the main characteristics of the examination competition data sets:

Instance	Conflict Density (%)	Exams	Students	Periods	Rooms	Period HC	Room HC
Exam_1	5.05	607	7891	54	7	12	0
Exam_2	1.17	870	12743	40	49	12	2
Exam_3	2.62	934	16439	36	48	170	15
Exam_4	15.0	273	5045	21	1	40	0
Exam_5	0.87	1018	9253	42	3	27	0
Exam_6	6.16	242	7909	16	8	23	0

Exam_7	1.93	1096	14676	80	15	28	0
Exam_8	4.55	598	7718	80	8	20	1
Exam_9	---	---	---	---	---	---	---
Exam_10	---	---	---	---	---	---	---
Exam_11	---	---	---	---	---	---	---
Exam_12	---	---	---	---	---	---	---

Table 1

At the time of writing, the competition is underway. It should be noted that all information relating to the hidden datasets will be released at the appropriate time.

The initial observation is that the conflict density for most of the data sets is quite low (generally around 5%). This is reflective of the amount of choice allowed to students within the modern curriculum, with a large variation in module/subject choices between each student. Theoretically this should mean that achieving feasibility will not be a major problem in terms of the clash matrix as with the first competition, although the extra hard constraints introduced may balance this out somewhat. The measurable problem ‘size’ (number of exams and students) varies to a certain extent across the set of problems, the largest of which could be argued as either exam_3 or exam_7 and the smallest as exam_4. The periods and rooms available will also have a measurable effect on the difficulty of achieving feasibility and/or a quality solution. Exam_4 may appear quite limited in terms of these resources, however the capacity of the single room available is 1200 (again, an effect of the ‘real-world’ nature of the data sets), and with exams able to share rooms, this still provides a certain degree of flexibility. The Period and Room Hard Constraints will also add to the measurable difficulty of each problem set, although it can be seen that Room Hard Constraints are rarely enforced, and when used, to a limited extent. The amount and type of Period Hard Constraints were reasonably similar across the data sets, although exam_3 clearly was much more constrained in this regard. It could be argued that this, combined with the large ‘size’ of exam_3, would place it as clearly the most difficult data set to schedule, both in terms of achieving feasibility and in obtaining a competitive evaluation score.

7.3 Results

Table 2.0 details the results using the method described.

Instance	Best	Q ₁	Median	Q ₃	Worst
----------	------	----------------	--------	----------------	-------

Exam_1	4633	4750	4799	4852	4889
Exam_2	405	405	425	430	441
Exam_3	9064	9214	9251	9388	9440
Exam_4	15663	15764	15821	15991	16365
Exam_5	3042	3062	3072	3104	3149
Exam_6	25880	25915	25935	26000	26080
Exam_7	4037	4091	4185	4257	4268
Exam_8	7461	7563	7599	7689	7827
Exam_9	---	---	---	---	---
Exam_10	---	---	---	---	---
Exam_11	---	---	---	---	---
Exam_12	---	---	---	---	---

Table 2.0

7.4 Discussion of results

Table 3 provides a summary of the results achieved using the method described in Section 7.2. The first point to note is that feasibility was gained on all of the data sets. Some of the data sets took a little longer to achieve feasibility during the construction phase than others, but all achieved a feasible solution within one minute of construction. In general, the most difficult solutions proved to be exam_1, exam_5 and exam_6, with the improvement getting stuck in local optima more often than the others. As expected, the larger data sets took more computation time per solution generation and evaluation, therefore less iteration time could be spent on these within the imposed competition time limit. Exam_2 was an interesting case, as the evaluation function based on the soft constraints imposed gave a much smaller value than with any of the others. A useful exercise in further analysis may be to attempt an approximation of the bounds for this problem (possibly using Linear Programming) in order to assess how close any solver can come to optimality.

7.5 Limitation of Formulation

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 to feasibility” is required if the formulation provided here is to be useful for such purposes.

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.

8 Post Enrolment based Course Timetabling

The first course timetabling track is the Post Enrolment-based Course Timetabling problem (Post Enroll CTT), which simulates the situation where students are given a choice of lectures they want to attend. The timetable is then constructed so that all students can attend their various chosen options. This track is actually an extension of the problem model used in the first competition, though in this case it also has extra hard constraints in order to move research further in the direction of real-world timetabling. These extra hard constraints are intended to make finding feasibility more difficult; thus there is a shift of emphasis from soft constraints to hard constraints.

8.1 The Problem Model

The Post Enrolment-based course timetabling problem involves assigning a set of “events” (lectures, tutorials, and so on) to a fixed number of timeslots and rooms in accordance with a set of constraints. The hard constraints for the problem are as follows. First, for each event there is a set of students who have enrolled to attend; thus events should be assigned to timeslots in such a way that no student is required to attend more than one event in any one timeslot. Next, each event also requires a set of room features (e.g. a certain number of seats, specialist teaching equipment, etc.), which will only be provided by certain rooms; thus each event needs to be assigned to a suitable room that exhibits the room features that it requires. The double booking of rooms is also prohibited. Hard constraints are also imposed stating that some events cannot be taught in certain timeslots (in a real world

situation, perhaps the lecturer might be unavailable to teach here, or perhaps some school policy needs to be adhered to). Finally, certain precedence constraints – stating that some events need to be scheduled before or after others – are also imposed on the problem

As mentioned earlier, this problem model is based on the problem-version used in ITC2002, though in this case the latter two hard constraints are additions to the model. The three soft constraints for this problem are also the same as the original competition, and are as follows:

Students should not be required to attend events in timeslots that occur at the end of a working day;

Students should not have to attend events in three or more consecutive timeslots in the same day;

Students should not be required to attend just one event in a day.

Note that a fuller description of this problem, including the precise methods of calculating solution quality, and also a detailed description of the rationale behind this problem are all given in the official technical report of this problem [Lewis et al(2007a)], available on the ITC2007 website.

The following lists the main features of the sixteen problem instances released during the competition

Instance	Events	Rooms	Students	Timeslots	Place Occupancy rate (%)
comp-2007-2-1	400	10	500	45	88.9
comp-2007-2-2	400	10	500	45	88.9
comp-2007-2-3	200	20	1000	45	22.2
comp-2007-2-4	200	20	1000	45	22.2
comp-2007-2-5	400	20	300	45	44.4
comp-2007-2-6	400	20	300	45	44.4
comp-2007-2-7	200	20	500	45	22.2
comp-2007-2-8	200	20	500	45	22.2
comp-2007-2-9	400	10	500	45	88.9
comp-2007-2-10	400	10	500	45	88.9
comp-2007-2-11	200	10	1000	45	44.4
comp-2007-2-12	200	10	1000	45	44.4
comp-2007-2-13	400	20	300	45	44.4

comp-2007-2-14	400	20	300	45	44.4
comp-2007-2-15	200	10	500	45	44.4
comp-2007-2-16	200	10	500	45	44.4

Table 3

8.2 Algorithm Description

The method we used for solving this problem is a time-dependent, metaheuristic-based algorithm, of which a full description can be found in [Lewis et al (2008)]. The basic approach is to split the solving process into three stages, with each stage being allocated exactly one third of the available run time. If a stage completes its objectives before reaching its time limit, then the remaining time is passed on to the following stage. In the first stage, the aim is to insert as many events as possible into the timetable whilst obeying all hard constraints except for the precedence constraints which are, for the time being, not considered. This is achieved using specialized constructive heuristics, together with an iterative improvement mechanism. At the end of this stage, any events that have not been inserted are left to one side (and are effectively removed from the problem). In the second stage, the algorithm then sets about trying to remove any violations of the precedence constraints, whilst not re-violating any of the other hard constraints in the process. This is achieved using a specialized, restricted neighborhood operator that forbids any moves causing a violation of these hard constraints. At the end of this stage, any events that are still seen to be causing a violation of the precedence constraints are also removed from the timetable. Finally, in the third stage, the aim is to try and eliminate any soft constraint violations occurring in the timetable. Rather like Stage-2, this is again achieved using a restricted neighborhood operator that does not allow any moves to be made that cause a violation of any of the hard constraints. Note that in the second and third stages of this method the neighborhood operators are used in conjunction with simulated annealing. In order to maximize the potential of this metaheuristic, in both cases special on-the-fly techniques are used in order to determine a cooling rate that allows the slowest cooling possible within the available time limits, thus maximizing the potential of the algorithm to produce good solutions.

8.3 Results

Table X summarizes the results gained on the competition benchmarks using the above method. It can be seen that we were able to gain feasibility to all instances at least once, and were also able to find feasibility in all runs with eight of the sixteen instances. In one case we were also able to obtain a perfect solution, in this case with the eighth instance. In general, instances 1, 2, 9, and 10 proved the hardest to solve – our success rates for finding feasibility were 45%, 22%, 2%, and 2% respectively. Note that all other instances had a success rate of $\geq 95\%$ in this respect.

Instance	Best	Q ₁	Median	Q ₃	Worst
comp-2007-2-1	0 (1294)	0 (1600)	17 (1492)	32 (1693)	105 (1944)
comp-2007-2-2	0 (1599)	18 (1718)	46 (1826)	80 (2016)	213 (2176)
comp-2007-2-3	0 (278)	0 (416)	0 (457)	0 (523)	0 (664)
comp-2007-2-4	0 (388)	0 (538)	0 (589)	0 (644)	0 (761)
comp-2007-2-5	0 (22)	0 (123)	0 (193)	0 (268)	0 (638)
comp-2007-2-6	0 (369)	0 (606)	0 (696)	0 (767)	20 (708)
comp-2007-2-7	0 (74)	0 (300)	0 (421)	0 (529)	0 (890)
comp-2007-2-8	0 (0)	0 (162)	0 (206)	0 (256)	0 (366)
comp-2007-2-9	0 (1582)	59 (1829)	80 (2312)	120 (1864)	214 (1609)
comp-2007-2-10	0 (2380)	83 (2339)	126 (2262)	194 (2303)	372 (2159)
comp-2007-2-11	0 (344)	0 (456)	0 (541)	0 (605)	0 (800)
comp-2007-2-12	0 (486)	0 (660)	0 (741)	0 (852)	125 (710)
comp-2007-2-13	0 (365)	0 (538)	0 (631)	0 (707)	19 (766)
comp-2007-2-14	0 (222)	0 (558)	0 (660)	0 (786)	27 (685)
comp-2007-2-15	0 (266)	0 (301)	0 (344)	0 (366)	0 (455)
comp-2007-2-16	0 (99)	0 (165)	0 (194)	0 (215)	0 (265)

Table 4.0: Results from 51 Runs on Each Instance. In each Case, the Distance to Feasibility is Displayed, Together with the Associated Soft Cost (in Brackets).

9 Curriculum based Course Timetabling

The second course timetabling track is concerned with Curriculum based Course Timetabling (Curriculum CTT). This problem consists of the weekly scheduling of lectures for several university courses within a given number of rooms and time periods, where con-

licts between courses are set according to the curricula published by the University and not on the basis of enrolment data.

This formulation applies to the University of Udine (Italy) and to many Italian and indeed International Universities, although it is slightly simplified with respect to the real problem to maintain a certain level of generality.

9.1 The Problem Model

The problem consists of the following entities:

Days, Timeslots, and Periods. We are given a number of teaching days in the week (typically 5 or 6). Each day is split in a fixed number of timeslots, which is equal for all days. A period is a pair composed of a day and a timeslot. The total number of scheduling periods is the product of the days times the day timeslots.

Courses and Teachers. Each course consists of a fixed number of lectures to be scheduled in distinct periods, it is attended by a given number of students, and is taught by a teacher. For each course there is a minimum number of days that the lectures of the course should be spread in, moreover there are some periods in which the course cannot be scheduled.

Rooms. Each room has a capacity, expressed in terms of number of available seats. All rooms are equally suitable for all courses (if large enough).

Curricula. A curriculum is a group of courses such that any pair of courses in the group have students in common. Based on curricula, we have the conflicts between courses and other soft constraints.

The solution of the problem is an assignment of a period (day and timeslot) and a room to all lectures of each course.

The set of hard constraints is the following:

Lectures: All lectures of a course must be scheduled, and they must be assigned to distinct periods. A violation occurs if a lecture is not scheduled or two lectures are scheduled in the same period.

RoomOccupancy: Two lectures cannot take place in the same room in the same period. Two lectures in the same room at the same period represent one violation. Any extra lecture in the same period and room counts as one more violation.

Conflicts: Lectures of courses in the same curriculum or taught by the same teacher must be all scheduled in different periods. Two conflicting lectures in the same period represent one violation. Three conflicting lectures count as 3 violations: one for each pair.

Availabilities: If the teacher of the course is not available to teach that course at a given period, then no lecture of the course can be scheduled at that period. Each lecture in a period unavailable for that course is one violation.

The soft constraints are the following:

RoomCapacity: For each lecture, the number of students that attend the course must be less or equal than the number of seats of all the rooms that host its lectures. Each student above the capacity counts as 1 point of penalty.

MinimumWorkingDays: The lectures of each course must be spread into the given minimum number of days. Each day below the minimum counts as 5 points of penalty.

CurriculumCompactness: Lectures belonging to a curriculum should be adjacent to each other (i.e., in consecutive periods). For a given curriculum we account for a violation every time there is one lecture not adjacent to any other lecture within the same day. Each isolated lecture in a curriculum counts as 2 points of penalty.

RoomStability: All lectures of a course should be given in the same room. Each distinct room used for the lectures of a course, but the first, counts as 1 point of penalty.

9.2 Description of Algorithm

The algorithm employed in the solution of this track is a dynamic tabu search. In details, it is a short-term tabu search with variable-size tabu length that also includes the dynamic modification of the weights of the cost components (including hard ones) based on the number of violations.

The neighborhood relation used is simple one that moves one lecture to a different period and/or a different room. The main parameters, namely the tabu length range and the rate of modification of the weights have been selected based on statistical tests.

9.3 Results

Table 5 shows the main features of all instances together with some statistical quantities. In details, it shows: courses, total lectures, rooms, periods per day, days, average number

of conflicts, average teacher availability, and average room occupation. With conflicts in this context we mean the pairs of lectures that cannot be scheduled at the same time (same course, same teacher, or same curriculum) divided by the total number of distinct pairs of lectures. Conflicts and availabilities are computed for single lecture rather than at course level, so as to take into account the fact that courses have different number of lectures.

Instance	Courses	Lectures	Rooms	Periods PerDay	Days	Curricula	Conflicts	Teacher Availability	Occupation
comp01	30	160	6	6	5	14	13,2	93,1	88,9
comp02	82	283	16	5	5	70	7,97	76,9	70,8
comp03	72	251	16	5	5	68	8,17	78,4	62,8
comp04	79	286	18	5	5	57	5,42	81,9	63,6
comp05	54	152	9	6	6	139	21,7	59,6	46,9
comp06	108	361	18	5	5	70	5,24	78,3	80,2
comp07	131	434	20	5	5	77	4,48	80,8	86,8
comp08	86	324	18	5	5	61	4,52	81,7	72
comp09	76	279	18	5	5	75	6,64	81	62
comp10	115	370	18	5	5	67	5,3	77,4	82,2
comp11	30	162	5	9	5	13	13,8	94,2	72
comp12	88	218	11	6	6	150	13,9	57	55,1
comp13	82	308	19	5	5	66	5,16	79,6	64,8
comp14	85	275	17	5	5	60	6,87	75	64,7

Table 5

Table 6 shows our results in the same way as the other tracks,

Instance	Best	Q₁	Median	Q₃	Worst
comp01	5	6	7	7	9
comp02	85	124	167	285	677
comp03	87	115	129	145	408
comp04	43	52	56	62	74
comp05	319	436	562	1024	1383
comp06	78	96	103	133	360
comp07	64	81	88	99	162
comp08	47	57	61	67	76
comp09	117	128	135	140	171
comp10	41	55	65	71	255
comp11	0	0	0	0	1
comp12	358	390	439	487	1094

comp13	83	91	95	99	110
comp14	63	72	78	81	91
comp15	---	---	---	---	---
comp16	---	---	---	---	---
comp17	---	---	---	---	---
comp18	---	---	---	---	---
comp19	---	---	---	---	---
comp20	---	---	---	---	---
comp21	---	---	---	---	---

Table 6

9.4 Discussion of results

Feasibility is reached for all instances quite easily. The most difficult instance turned out to be comp05, for which it took about 13 seconds on the average, followed by comp02 and comp07 that took about 3 seconds, all the others are solved to feasibility in about 1 second or less.

9.5 Limitation of Formulation

The actual formulation used at the University of Udine, with respect to the one issued for ITC2007, has the following extra features:

1. A cost component dealing with the lunch break for students: at least one free slot among those around the lunch time.
2. The curriculum compactness feature is more complex, and specific patterns are more penalized than others.
3. There is a maximum daily student load for each curriculum.
4. Some specific lectures must be (must not be) in consecutive periods.
5. Rooms might not be available in certain periods, and they must be not suitable for specific lectures.
6. If a room is too big for a class, this is also penalized (this is not only for the unpleasant feeling that an empty room provokes, but also to save big rooms for unforeseen activities).
7. Weight assigned to soft violations are more complex, and they depend also on the number of students in the curriculum.

8. Teacher preferences on periods and rooms are only included as soft constraints.

The only reason for which we have decided to remove all the above features is to maintain a certain degree of generality, so as to do not inflict to the participant the burden to understand all the details of the formulation. The selection of the features to include in the formulation has been based on the aim to balance different types of constraint. Needless to say, if in the future this formulation will prove to be inappropriate (e.g., too simple), some features could be reintroduced for future research.

We are currently working on new formulations (with different cost components) with the aim of getting a larger community to get an interest in this setting.

10 Conclusions

This paper has presented detailed information relating to the 2nd International Timetabling Competition. The competition rules have been outlined and discussed along with the major differences from the 1st Competition. In addition, limitations of each of the new formulations presented have been discussed as a means of illustrating the need for further work in addressing the identified gap which currently exists between research and practice in relation to this research area. Results have been presented for each of the competition tracks along with descriptions of the associated algorithms.

It should be noted that although the competition has been highly successful in bringing the community together introducing new ideas and creating general interest in the field. The results should be treated with care due to programming issues and the possibility of competitors tuning of algorithms to datasets. It is important to emphasise that the competition serves more as an encouragement to researchers as opposed an identifier of best techniques. Just because an algorithm beats another on some instances, it doesn't mean it is a superior algorithm in general.

References

Abdullah et al.(2007), S.Abdullah, E.K.Burke and B.McCollum, Using a Randomised Iterative Improvement Algorithm with Composite Neighbourhood Structures for the University Course Timetabling Problem, accepted for publication in *Metaheuristics - Progress in Complex Systems Optimization* (eds. K.F.Doerner, M.Gendreau,

- P.Greistorfer, W.J.Gutjahr, R.F.Hartl and M.Reimann), to appear in the Springer Operations Research / Computer Science Interfaces Book series, 2007.
- Burke, E. K., Bykov, Y., Newall, J. P., Petrovic, S., (2004) "A Time-Predefined Local Search Approach to Exam Timetabling Problems", IIE Transactions, 36(6), 509-528.
- Carter et al. (1996), M.W. Carter, G. Laporte and S.Y. Lee. (1996). Examination timetabling: Algorithmic strategies and applications. *Journal of Operational Research Society*, 47(3): 373-383.
- Chiarandini et al.(2006), M. Chiarandini, M. Birattari, K. Socha, and O. Rossi-Doria, An effective hybrid approach for the university course timetabling Problem, *Journal of Scheduling*, 9 (5): 403--432, 2006.
- Di Gaspero et al. (2007), L. Di Gaspero, B. McCollum, and A. Schaerf. The second International timetabling competition (ITC2007): Curriculum-based course timetabling (track 3). Technical Report QUB/IEEE/Tech/ITC2007/CurriculumCTT/v1.0/1, School of Electronics, Electrical Engineering and Computer Science, Queen's University, Belfast (UK), August 2007.
- Di Gaspero and Schaerf (2006), L. Di Gaspero and A. Schaerf, Neighborhood portfolio approach for local search applied to timetabling problems. *Journal of Mathematical Modeling and Algorithms*, 5 (1) 65--89, 2006, DOI: 10.1007/s10852-005-9032-z.
- Dueck, G., (1990), "Threshold Accepting: A General Purpose Optimization Algorithm Appearing Superior to Simulated Annealing", *J. Computational Physics*, Vol. 90, 161-175.
- Kostuch (2005), Ph. Kostuch, The university course timetabling problem with a three-phase approach. In Edmund Burke and Michael Trick, editors, *Proc.of the 5th Int. Conf.on the Practice and Theory of Automated Timetabling (PATAT-2004)*, selected papers}, volume 3616 of *Lecture Notes in Computer Science*}, pages 109--125, Berlin-Heidelberg, 2005. Springer-Verlag.
- Lewis et al (2007a) Lewis, R., Paechter, R., McCollum, B., Post Enrolment based Course Timetabling: A Description of the Problem Model used for Track Two of the Second International Timetabling Competition, *Cardiff Working Papers in Accounting and Finance A2007-3*. Prifysgol Caerdydd/ Cardiff University, Wales. ISSN: 1750-6658, v 1.0.
- Lewis et al. (2007b), R. Lewis, B. Paechter, and O. Rossi-Doria. Metaheuristics for University Course Timetabling. In *Evolutionary Scheduling (Studies in Computational In-*

- telligence, vol. 49), K. Dahal, Kay Chen Tan, P. Cowling (Eds.) Berlin: Springer-Verlag, pp 237-272
- McCollum(2007a), McCollum, B., A Perspective on Bridging the Gap between Theory and Practice in University Timetabling, Practice and Theory of Automated Timetabling VI, Springer LNCS Vol 3867, 2007, pp 3-23.
- McCollum (2007b), McCollum, B., McMullan, P., Burke, E.K., Parkes, A.J., Qu, R., The second International timetabling competition (ITC2007): Examination Timetabling Track. Technical Report QUB/IEEE/Tech/ITC2007/Exam/v4.0/17, School of Electronics, Electrical Engineering and Computer Science, Queen's University, Belfast (UK), August 2007.
- McMullan P. (2007) An Extended Implementation of the Great Deluge Algorithm for Course Timetabling, Computational Science – ICCS 2007, Springer LNCS Vol 4487, July 2007, pp 538-545.
- Lewis et al. (2007), R. Lewis, B. Paechter, and O. Rossi-Doria. Metaheuristics for University Course Timetabling. In Evolutionary Scheduling (Studies in Computational Intelligence, vol. 49), K. Dahal, Kay Chen Tan, P. Cowling (Eds.) Berlin: Springer-Verlag, pp 237-272
- Qu et al. (2007), 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.
- Schaerf (1999), A. Schaerf., A survey of automated timetabling, Artificial Intelligence Review, 13 (2): 87--127, 1999.
- Schaerf and Di Gaspero (2007), A. Schaerf and L. Di Gaspero, Measurability and Reproducibility in University Timetabling Research: Discussion and Proposals. In Edmund Burke and Hana Rudova, editors, Proc.of the 6th Int. Conf. on the Practice and Theory of Automated Timetabling (PATAT-2006), selected papers, volume 3867 of Lecture Notes in Computer Science, pages 40--49, Berlin-Heidelberg, 2007. Springer-Verlag.