MISTA 2009

An Extended Great Deluge Approach to the Examination Timetabling Problem

B. McCollum, P.J. McMullan, A. J. Parkes, E.K. Burke, S. Abdullah

Abstract A two phased approach incorporating the extended great deluge technique is detailed in relation to the Examination Timetabling Problem as described in the 2nd International Timetabling Competition (ITC2007). The approach proves to be both robust and general. Robust in the sense that it is capable of producing six of the best results published in literature so far on the benchmark datasets and general as the technique has produced in the recent past some of the best results on existing course timetabling benchmark datasets. The datasets used as part of this research, introduced during ITC2007, are described and discussed in detail. We present the results of our technique in relation to the competition results and provide a comparison between the outlined method and those of the competition entrants from the international arena, in order to highlight both characteristics of the technique on the datasets used.

B G C McCollum
School of Electronics, Electrical Engineering and Computer Science, Queen's University, Belfast
E-mail: b.mccollum@qub.ac.uk

P J McMullan
School of Electronics, Electrical Engineering and Computer Science, Queen's University, Belfast
E-mail: p.p.mcmullan@qub.ac.uk

A J. Parkes
School of Computer Science, University of Nottingham
E-mail: ajp @ cs.nott.ac.uk

E K Burke
School of Computer Science, University of Nottingham
E-mail: ekb@cs.nott.ac.uk

Salwani Abdullah
Center for Artificial Intelligence Technology, Universiti Kebangsaan Malaysia, 43600 UKM, Bangi, Selangor, Malaysia
E-mail: salwani@ftsm.ukm.my

1 Introduction

The automated timetabling of examinations associated with courses within universities is often a complex and time consuming task. From a practical perspective, the quality of the final solution is usually measured firstly in terms of how workable the solution is in terms of resource assignment e.g. availability of students, accessibility of rooms, ordering of examinations, etc. and secondly, on how well a number of institutional defined quality measures are satisfied e.g. how well student exams are spread out throughout the designated examination period [1]. From a research perspective, this measure of quality translates into the satisfaction of what are termed hard and soft constraints and is often achieved through an automatic scheduling process comprising of a two-stage methodology involving the construction and subsequent improvement of a timetabled solution. Construction involves achieving a feasible solution whereby no hard constraints are broken e.g. no students should be expected to take more than one examination at one time. Improvement minimises the violation of desirable constraints termed soft constraints e.g. a student having to sit two examinations in a row. The quality of the final solution is assessed by measuring the outcome after these phases i.e. the degree of satisfaction of hard and soft constraints. Although the problem is similar across universities, various formulations of the problem have appeared within the academic literature over the past fifty years or so. Many search based techniques have been applied to various formulations of the problem and the associated datasets. For a detailed overview of these the reader is referred to [2].

Recently, the 2nd International Timetabling Competition [3] had examination timetabling of one of its three tracks. This track introduced a formulation of the problem which incorporated and brought together a number of real world constraints [4]. This problem formulation adds significantly to previously used models within the research field as many more real world constraints were described and whose cost to overall solution quality were incorporated within the evaluation process. In addition, during the competition, twelve associated data sets were released to the research community. The problem studied here relates to that introduced formulation. Detailed information relating to this exam timetabling formulation, details of the other two tracks, and also overall competition rules can be found at the competition website (www.cs.qub.ac.uk/itc2007). Five finalists from the original submissions were chosen by the competition organisers based on reported results. Importantly, a limit was placed on the computation time allowed for competitor's technique to produce a solution. Competition organisers ran the finalists submitted techniques on a number of previously non-released datasets ('Hidden'), as well as those datasets released as part of the competition ('Early and Late'). Subsequently an ordering was calculated to decide the overall winner based on performance on all the datasets. This can be viewed at http://www.cs.qub.ac.uk/itc2007/winner/bestexamtrack.htm.

Here, the organisers of the examination track report the results achieved through the use of an two phase approach incorporating an adaptive construction phase followed by the employment of an extended great deluge technique. The purpose of running the organisers' solvers on the instances and releasing these results is twofold: On 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 what can be considered a baseline (upper-bound) that can be exploited in future comparisons.

The remainder of this paper is as follows; Section 2 provides the necessary information on the formulation of the examination timetabling problem as introduced as part of ITC2007; Section 3 provides a background to the technique reported as part of this work; Section 4 describes the approaches taken by the five finalists who took part in the competition. It should be noted that all background material described is related to these finalists. This will be updated as further researchers trial their techniques on the formulation and datasets; Section 5 describes the detailed implementation of the extended great deluge along with the results obtained. Finally the paper is concluded making comment on the effectiveness of the technique studied and potential future research areas.

2 The Examination Timetabling Model

The formulation of the Examination Timetabling Problem introduced as part of ITC2007 is described as 'post-enrolment', as the individual examinations each student is taking are known before the timetabling process. This is an important distinction within the area of University Timetabling as the closely related problem, that of course timetabling, can be considered as either post-enrolment or curriculum based. Conflicts between course events can be determined either by common students in the post-enrolment model, or by a hierarchical modular structure within the curriculum based model. Both of these formulations are described as part of ITC2007 [3]. The timetabling process associated with examination timetabling involves placing exams into a number of pre-defined periods within a defined examination session while satisfying a number of hard and soft constraints. A feasible solution is one in which all hard constraints are satisfied. As described in the introduction, the quality of the solution is measured in terms of soft constraints satisfaction. The characteristics which are used to define an individual problem instance include the examinations, students and enrolments, resources such as periods and rooms, hard constraints applied to resources, and general institutional soft constraints. Table 1 lists the main characteristics for each 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	7.84	169	655	25	3	10	0
Exam_10	4.97	214	1577	32	48	58	0
Exam_11	2.62	934	16439	26	40	170	15
Exam_12	18.45	78	1653	12	50	9	7

Table 1 – Problem Benchmark Characteristics

The conflict density is a measurement of the number of conflicts examinations i.e. how tightly the problem is constrained in terms of student enrolments. The initial observation is that the conflict density for most of the data sets is quite low (for the most part around 5% or 6%). This is reflective of the amount of choice allowed to students within the modern curriculum, with a large variation in course or subject choices between each student. 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*/*exam_11* or *exam_7* and the smallest as *exam_9* or *exam_12*. The periods and rooms available will also have a measurable effect on the difficulty of achieving feasibility and/or a quality solution. The instances *exam_3* and *exam_11* are practically the same data sets, although *exam_11* has a much smaller set of period resources available with which to timetable. The instances used reflect the 'real-world' nature of the data sets which are encountered in actual institutions. In the case of *exam_3* and *exam_11* a common situation arises where the examination session must be shortened to minimize space and staff costs, although all existing constraints must still be adhered to as much as possible.

Information on the structure, length and number of individual periods is also made available. An examination session is made of a number of periods over a specified length of time. This can range from one to two weeks in relation to the data provided. Period lengths, within which a set of examinations or varying duration must be placed, range from one to three hours. A set of rooms and associated capacities are provided.

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 duration restrictions are not violated;
- Period related hard constraints e.g. Exam_A must be placed 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:

- Student has to sit two exams in a row (adjacent on same day);
- Student has to sit two exams in a day;
- Student does not have a specified spread (in terms of periods) of examinations;
- Mixed durations of examinations occur within individual periods;
- Examinations of large class sizes appear later in the examination session;
- Period related soft constraints;
- Room related soft constraints;

As can be seen, these constraints can effectively be split into two groups; those which are resource specific and those which can have a global setting. Resource specific constraints can be set for each period and each room and allows ‘control’ of how resources can be used in constructing a solution. Global Setting constraints can be set relative to each other. Within the described model, institutions 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 defined 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. 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 are reasonably similar across the data sets. However, given the size and amount of exams and enrolments in some of the problem instances compared to resources available, some could be considered more difficult in this regard and possible more difficult to schedule, both in terms of achieving feasibility and in obtaining a competitive evaluation score.

3 Competition Entrants and Placings

A summary of the competition results is presented in Table 3a in Section 5. However the overall placings are as follows.

Tomáš Müller from Purdue University was the competition winner, having achieved ten out of twelve of the best scores. The algorithm described used a three phased incremental approach [5]. During construction an Iterative Forward Search algorithm was employed in finding an initial feasible solution. During each iteration of the algorithm an examination is chosen and assigned to a room and time. If the assignment causes a hard constraint to be broken then the existing assignment which is causing the problem is unassigned. The process ends when all examinations are assigned a room and time. The algorithm employs both ordering and assignment heuristics in order to speed up the process. In addition Conflict-based Statistics are used during the iterations in an attempt to avoid repetitive assignments which have previously proven to be detrimental to the developing solution. During the second phase of the algorithm, hill climbing is used to find the local optimal. A neighbourhood is chosen with equal probability from a determined list relating to swapping/changing periods and rooms for randomly chosen examinations. This phase is terminated after a specified number of iterations during which no improvement is experienced. The Great Deluge Algorithm is then engaged in an attempt to improve the solution through widening the search.

Christos Cogos from University of Patras, Greece, was placed second in the competition. His method utilised a GRASP (Greedy Randomized Adaptive Search Procedure) based process combined with other meta-heuristic techniques [6]. The construction phase begins by building five lists of examinations based on various criteria. A tournament based algorithm is used in selecting which exam should be placed in the timetable until all lists are empty. This is carried out iteratively using different starting points in relation to initial time periods. A backtracking strategy, employing a tabu list is employed as required. A Simulated annealing procedure is used in the second phase. In a third phase, integer programming using branch and bound is used to scrutinize and analyse individual periods with the purpose of room changes. Periods are chosen based on an ordering based on overall satisfaction (CSP) of particular soft constraints.

Atusta et al. from Japan used a constraint satisfaction problem solver incorporating tabu search and iterated local search [7,17]. By specifying initial weights, the solver distinguishes between soft and hard constraints and their weights are dynamically controlled during computation to improve performance. The instances were formulated using linear 0-1 inequalities, quadratic 0-1 inequalities, and all-different constraints. The technique proved to be very effective across all three tracks of the competition.

Geoffrey De Smet from Belgium incorporated local search techniques within the Drools solver [8]. Drools is an Open-Source Business Rule Management System (BRMS) (http://www.jboss.org/drools/). The developers have provided an integrated environment which allows problem specific semantics to be linked with Domain Specific Languages, graphical editing tools, web based tools and developer productivity tools. The approach described develops the Drools-solver incorporating the Drools rule engine and a local search mechanism. Initially each constraint is written in the Drools Rule Language. Examinations are subsequently ordered on the basis of size and duration. Examinations are assigned to the 'best' position within the timetable chosen by a placing heuristic. This is followed by a local search mechanism which uses three neighbourhoods related to moving and swapping time periods and rooms. The searching continues based on a heuristic which incorporates a tabu based approach.

Nelishia Pillay from the University of KwaZulu-Natal, South Africa came fifth. Inspired by a biological approach, the algorithm mimics cell behaviour [9]. After examinations are ordered heuristically using saturation degree, the examinations are sequentially allocated to the available “cells” within the timetable structure i.e. available times. If more than one time allocation is possible the choice is based on minimum overall solution penalty. If there is a choice once again, allocation is made on a random basis. Rooms are chosen on a best fit heuristic with respect to rooms. Eventually, no cells remain where an examination on the list can be placed without breaking a hard constraint. When this is the case, the already placed examination is moved to a cell which allows minimisation of the overall soft constraint penalty. This is described as cell division. Once this is not possible without the subsequent breaking of hard constraints, a process called cell interaction takes place. This involves a swapping process with the purpose of removing the hard constraint. This process continues until a feasible solution has been reached. The authors liken this to the development of a fully functional organism. Once a feasible solution is reached, improvement is achieved through a process known as cell migration. This involves heuristically swapping the contents of cells that have equal durations.

4 The EGD Approach

The EGD approach has a construction phase followed by improvement. Construction is implemented using the existing adaptive ordering heuristic from [10]; it uses a weighted order list of the examinations to be scheduled based on individual soft penalties and 'difficulty to schedule' penalties. Weightings are increased for each examination based on localised penalties encountered as each are placed, with unscheduled examinations given a much larger increase, based on a formulation involving the maximum general penalty encountered. This latter is an extension to [10] and has been seen in experimentation to achieve improved construction solutions to the standard adaptive construction technique. Once feasibility is achieved, the heuristic continues with the aim of providing an improved solution. It has been found that it is preferable that the construction phase should continue until approximately 9% of the entire process has completed, at which point the improvement phase begins [11].

Figure 1: Extended Great Deluge Algorithm

```
Set the initial solution s using a construction heuristic;
   Calculate initial cost function f(s)
   Set Initial Boundary Level B0 = f(s)
   Set initial decay Rate •B based on Cooling Parameter
   While stopping criteria not met do
      Apply neighbourhood Heuristic S* on S
      Calculate f(s*)
      If f(s*) <= f(s) or (f(s*) <= B Then
          Accept s = s*
      Lower Boundary B = B – •B
      If no improvement in given time T Then
         Reset Boundary Level B0 = f(s)
         Set new decay rate •B based on Secondary
             Cooling Parameter
```

Improvement is carried out through implementation of an extended version of the Great Deluge Algorithm. Pseudo code for the algorithm for the Extended Great Deluge is presented in Figure 1. The Great Deluge (also known as Degraded Ceiling) was introduced by Dueck [12] as a faster alternative to Simulated Annealing. It uses a boundary condition rather than a probability measure with which to accept worse solutions i.e. if the penalty function is below a certain value a move in the selected neighbourhood will be automatically accepted, but if it is above it will be rejected The boundary is initially set slightly higher than the initial solution cost, and reduced gradually through the improvement process. This has already been applied successfully to construction and improvement techniques in timetabling [13,14]. In addition, [15] describes a modification of the Great Deluge algorithm for Multi-criteria decision making. 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 [15]. The aim of this approach is to both improve the speed at which an optimal solution can be found and at the same time utilise the benefits of this technique in avoiding the trap of local optima. Once again, in order to reduce the amount of time taken, relatively simple neighbourhood moves are employed. In further explanation of the difference, generally, the Great Deluge will terminate when a lack of improvement has been observed for a specified amount of time, as the best solution using a particular neighbourhood has been reached. Rather than terminating, the extended approach employs reheating in order to relax the boundary condition to allow worse moves to be applied to the current solution. Cooling continues and the boundary is reduced at a rate according to the remaining length of the run.

Two basic neighbourhoods are employed within local search, i.e. random moving and swapping individual examinations, 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 allows 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. A full description of the algorithm along with the results achieved in relation to a set of course timetabling problems is given in [16].

The parameters used within the Extended Great Deluge determine the behaviour of the method during improvement, and are listed as follows:

- Initial Boundary
- Decay Rate
- 'Wait for Non-Improvement' parameter
- Post-reheat boundary
- Post-reheat decay rate
- Probability of heuristics employed

The primary parameters used, initial boundary and initial decay rate, dictate how fast the boundary is reduced and ultimately the narrowing condition for accepting worse moves. The approach outlined in this paper uses a Decay Rate proportional to 50% of the entire run. Based on initial iteration timings and the remaining time to produce a final solution within the time limit, the total number of iterations allowed can be calculated in order to determine the appropriate proportions for decay rate. This faster decay will force the algorithm to attempt to reach the optimal solution by, at the very most, half-way through the process. Generally, a continuous lack of improvement will occur before this is reached, at which time the re-heat mechanism is activated.

The 'wait' parameter dictates when to activate the re-heat mechanism due to lack of improvement, specified in terms of percentage or number of total moves in the process. Through experimentation with a number of data set instances a general value for this parameter was established, roughly 5% of the complete run. After reheat the Boundary ceiling is once again set to be greater than the current best evaluation by a similar percentage to that applied in the initial boundary setting. The subsequent decay is set to a 'quicker' rate than with the initial decay, in order to increase the speed of the exploration of neighbouring solutions for improvement. The general setting chosen for the algorithm outlined is set to 25% of the remaining time, with the improvement wait time remaining unchanged. All parameters were established as the most effective as a result of experimentation. The two heuristics used are "Move" (random examination is moved to a random timeslot) and "Swap" (two random examinations swap timeslots), while ensuring that a feasible solution is maintained. Both heuristics may cause a new choice of room allocation if required. There are approximately two "Moves" made for each "Swap".

The choice of parameter values is historical from application to the original course timetabling problems. The ultimate values chosen through experimentation proved highly effective in obtaining best results for these problem data sets, but required further application to a wider range and type of data sets in order to ensure these were not tuned to the specific data sets used. As we shall see in the next section, the results obtained when applying the technique to the new competition data sets are also highly competitive to current results from the other entrants. It was important to use the same parameter values for these data sets rather than apply new values in order to obtain best results.

5 Experimental Results

As with the competition submissions, the algorithms described here are stochastic, meaning that different runs are generated with different random seeds. During construction of a feasible solution, random weightings may be given for equally 'difficult to schedule' examinations. Random choices of examinations, periods and to a certain extent room allocations allow a more diverse search of each current neighbourhood. 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. Firstly for EGD we performed 51 runs on each instance using a different seed in each case and, for each run, calculated the 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 obtain the median without the need for interpolation. By using 51 runs rather than the 10 or 11 of the competition we are also able to give reasonable results for the quartiles.

Instance	Best	Q1	Median	Q3	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	1071	1071	1071	1076	1079
Exam_10	14374	14464	14552	14629	14668
Exam_11	29180	29257	29358	29594	29699
Exam_12	5693	5693	5699	5711	5751

Table 2 - Results using the EGD method

Table 2 provides a summary of the results achieved by the EGD algorithm. The first purpose of these results is to give an indication of the variability between runs of the algorithm. The next 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.

The "ITC-2007" block of results in Table 3a provides the best values (highlighted in bold) obtained using the submitted competitor's techniques using 11 independent runs on each instances (that is, using a different random seed for those entries that used randomised methods). The results in this block were obtained by the organisers under the competition rules. The result format is a dash indicating that no feasible solution was achieved. Otherwise, the best overall soft constraint violation score is provided.

	(a) ITC-2007 (11 runs per instance)						**(b) Post ITC-2007 (51 runs per instance)**	
Solver **Instance**	**Muller**	**Cogos**	**Atsuta et al**	**De Smet**	**Pillay**		**Muller (ITC2007 code)**	**EGD**
Exam_1	**4370**	5905	8006	6670	12035		**4370**	4633
Exam_2	**400**	1008	3470	623	3074		**385**	405
Exam_3	**10049**	13862	18622	-	15917		9378	**9064**
Exam_4	**18141**	18674	22559	-	23582		**15368**	15663
Exam_5	**2988**	4139	4714	3847	6860		**2988**	3042
Exam_6	**26950**	27640	29155	27815	32250		26365	**25880**
Exam_7	**4213**	6683	10473	5420	17666		4138	**4037**
Exam_8	**7861**	10521	14317	-	16184		7516	**7461**
Exam_9	**1047**	1159	1737	1288	2055		**1014**	1071
Exam_10	16682	-	15085	**14778**	17724		14555	**14374**
Exam_11	**34129**	43888	-	-	40535		31425	**29180**
Exam_12	5535	-	**5264**	-	6310		5357	5693

Table 3 - Combined Results, (a) The entries in the first block are results for entrants as obtained in the competition itself, (b)The second block contains the more recent results.

The winner, Tomas Muller, produced 10 of the best results from the top five competitors. Cogos achieved 10 feasible solutions. Atsuta produced 11 feasible solutions and recorded the best result on *Exam_12*. De Smet achieved 7 feasible solutions and the best recorded score for *Exam_10*. Pillay achieved 12 feasible solutions. Note that although Cogos came second, he only achieved 50% feasibility with the hidden data sets *Exam_9* to *Exam_12*, and had the worst result for *Exam_11* compared to Pillay (fifth place) who achieved 100% feasibility and had a better result than Cogos on *Exam_11*. This could suggest that Pillays technique is more effective as a general solution, in terms of feasibility at least, than that employed by Cogos.

Given that Muller was the winner of the competition, Table 3b compares Mullers original submission and the EGD technique on a larger number of runs than the competition. Note that the 51 runs for EGD make it incomparable with the competition results as such. Also, we do emphasise that this should not be taken to reflect badly on the initial entrants. In particular the EGD is produced after the competition and so has had significantly more development time. Furthermore, the parameters within EGD were selected on the basis of access to all 12 instances; in contrast, the competition entrants did not have access to the hidden instances and so might well be argued to be less suitable on those.

Under this comparison Muller and EGD both get 6 best each, and so it seems that EGD is comparable to existing state of the art techniques, and from previous application to other data sets and a different problem domain (course timetabling), can be considered as a generalised technique to solving timetabling problems. It will be interesting to run all techniques at some future point with further hidden data sets, in order to provide a wider test for consistency in the approaches outlined above.

6 Conclusion and Future Work

This paper has described the application of a two-phased approach to solving the Examination Timetabling problem introduced as part of the 2^{nd} International Timetabling Competition. An adaptive heuristic is used to gain feasibility during an initial construction phase. Improvement is achieved through employment of a variant of the Great Deluge (Degraded Ceiling) algorithm. The approach attempts to exploit the inherent advantages with this Extended Great Deluge technique in escaping from local optima while also maintaining a relatively simple set of neighbourhood moves. This is particularly effective here as it reduces the time required for an iterative search process, allowing a balance to be achieved between diversification and intensification within the search strategy. As has been shown, the approach has been successful in achieving a general improvement in solution evaluation as compared to currently published results for the examination timetabling problem, given a common termination criterion.

It is stressed that the results presented here are obtained by examining a greater number of runs that those used during the verification stage of ITC2007. As detailed this was to allow more detailed analysis of the technique. Although, the rules of the competition did not permit the organisers to enter for sound academic reasons, it was felt here for the sake of fairness, the competition time limit should be adhered to in reporting results. In future comparisons, it is proposed various time limits will be used to allow analysis of the effectiveness of techniques when more realistic times are available for the provision of solutions.

In general, this research is part of a much wider analysis of the technique in order to determine whether further improvement can be achieved by modification of the associated parameters and variables used in the process, as outlined in Section 4, while retaining generality for all data sets. It is intended to provide some mechanism for self-adaptation of these parameters based on characteristics of the data sets under consideration as part of this goal. In addition the relationship between the construction and improvement will be investigated to establish rules between the trade-off between the two phases. This would further add to the capability of providing a general algorithm which can ultimately be used for both examination and course timetabling.

References

1. B.McCollum, A Perspective on Bridging the Gap between Research and Practice in University Timetabling (2007), Practice and Theory of Automated Timetabling VI (eds. E.K.Burke and H.Rudova), Lecture Notes in Computer Science Volume 3867, Springer 2007, 3-23.
2. R.Qu, E.K.Burke, B.McCollum, L.G.T.Merlot, S.Y.Lee, A Survey of Search Methodologies and Automated System Development for Examination Timetabling, Accepted for publication by Journal of Scheduling. online publication October 2008, doi: 10.1007/s10951-008-0077-5.
3. B. McCollum, A.Schaerf, B.Paechter, P. McMullan, R.Lewis, A. Parkes, L. Di Gaspero, R.Qu, E. Burke, Setting The Research Agenda in Automated Timetabling: The Second International Timetabling Competition, Accepted for publication to INFORMS Journal of Computing, To be published 2009.
4. B.McCollum, P.McMullan, E.K.Burke, A.J.Parkes, R.Qu, A New Model for Automated Examination Timetabling, Submitted Post PATAT08 special issue of Journal of Scheduling, Available as technical report QUB/IEEE/Tech/ITC2007/Exam/v4.0/17, From http://www.cs.qub.ac.uk/itc2007/examtrack/exam_track_index.htm

5. Muller, T., ITC2007 Solver Description: A Hybrid Approach, Practice and Theory of Automated Timetabling (PATAT 2008), Montreal, 19-22, August 2008.
6. Cogos, C., Alefragis, P., Housos, E., A Multi-Staged Algorithmic Process for the Solution of the Examination Timetabling Problem, Practice and Theory of Automated Timetabling (PATAT 2008), Montreal, 19-22, August 2008.
7. Mitsunori Atsuta, Koji Nonobe, and Toshihide Ibaraki (Japan), ITC2007 Track 1: An Approach using general CSP solver. www.cs.qub.ac.uk/itc2007
8. Geoffrey De Smet: ITC2007 - Examination Track, Practice and Theory of Automated Timetabling (PATAT 2008), Montreal, 19-22, August 2008.
9. Pillay, A Developmental Approach to the Examination timetabling Problem. www.cs.qub.ac.uk/itc2007
10. Burke, E. K., Newall, J. P., (2004), "Solving Examination Timetabling Problems through Adaptation of Heuristic Orderings", Annals of operations Research, 129: 107-134.
11. E. K. Burke, G. Kendal, B. McCollum, P.McMullan (2007) , Constructive versus Improvement Heuristics: An Investigation of Examination Timetabling, 3rd Multidisciplinary International Scheduling Conference: Theory and Applications, 28-31 August 2007, Paris.
12. Dueck, G., (1990), Threshold Accepting: A General Purpose Optimization Algorithm Appearing Superior to Simulated Annealing", J. Computational Physics, Vol. 90, 161-175.
13. Burke, E. K., Bykov, Y., Newall, J. P., Petrovic, S., (2003) "A Time-Predefined Approach to Course Timetabling", Yugoslav J. of Operational Research (YUJOR), 13(2): 139-151.
14. 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.
15. Petrovic, S., Burke, E. K., (2004), University Timetabling, Handbook of Scheduling: Algorithms, Models and Performance Analysis, CRC Press, Chapter 45.
16. McMullan, P (2007), An Extended Implementation of the Great Deluge Algorithm for Course Timetabling, Lecture Notes in Computer Science, Springer, Vol 4487, 538-545, 2007.
17. K. Nonobe, and T. Ibaraki (2001), An Improved tabu search method for the weighted constraint satisfaction problem, INFOR, Vol. 39, No.2, pp. 131-151, 2001.