Adaptive Decomposition and Construction for Examination Timetabling Problems

Rong Qu, Edmund K Burke
Automated Scheduling, Optimisation and Planning (ASAP) Group, School of CSiT
University of Nottingham, Nottingham, NG8 1BB, UK, rxq,ekb@cs.nott.ac.uk

Decomposition techniques have not been widely investigated in timetabling research mainly due to the complexity of the problems. In this paper, we develop a new general adaptive decomposition technique where problems are iteratively partitioned into two sub-sets, each containing a set of events with different levels of difficulty. The events in these two sets, namely the *difficult set* and the *easy set*, are ordered in turn and used to construct the solutions. Potentially difficult events are adaptively included in the *difficult set*, whose size is also adaptively adjusted according to the solution quality obtained in previous iterations. It is observed that in most cases, the *difficult set*, although of small size, contributes to a much larger portion of the total cost of the solutions constructed. This simple yet effective adaptive technique obtained competitive solutions compared with state-of-the-art approaches in the literature for benchmark exam timetabling problems.

Keywords: [Adaptive, Decomposition, Timetabling].

1 Introduction

As one of the most important administrative activities in all universities, the exam timetabling process has attracted significant research attention in the last four decades [13, 15]. A general exam timetabling problem consists of scheduling a list of exams into a limited set of timeslots so as to satisfy some constraints absolutely (the *hard* constraints) and some others as much as is possible (the *soft* constraints). The most common hard constraint is that of avoiding assigning exams with common students to the same timeslot. The most common soft constraint in the literature is to spread students' exams as far as possible to allow enough revision time. These constraints are considered in the benchmark exam timetabling problems tested in this paper. They were introduced by Carter, Laporte and Lee [9] in 1996 and have since been widely used in the literature. The problem consists of assigning exams (81-682 across different instances) to a limited number of timeslots, while satisfying the above hard and soft constraints. The quality of the solutions is evaluated by the average cost of how exams are spread for each students. During the years, there has been an issue with different instances circulating under the same name. This situation is clarified in [15]. More details can be found at http://www.cs.nott.ac.uk/∼rxq/data.htm.

The last ten years have seen a considerable increase in the number of research publications where meta-heuristics were developed for educational timetabling (e.g. [7, 15, 16]). These include Tabu Search (e.g. [14, 18]), Simulated Annealing (e.g. [12, 17]) and Evolutionary Algorithms (e.g. [6]), etc. Usually different mechanisms are defined to underpin efficient exploration of the search space. Very often, the methodologies in the literature represent tailor-made algorithms that work very well on the specific problem for which they developed but not on others.

Recent research in timetabling has seen some development on improving the flexibility and generality of search algorithms (see [15]). For example, variable neighbourhood search [10] and large scale neighbourhood search [1] employ different neighborhood structures to enable wider exploration in the search space and have been developed with some success for exam timetabling.

Another example is provided by hyper-heuristic research which search upon a space of heuristics rather than upon the actual solutions (e.g. [3]) in attempt to be more generally applicable across a wider range of problems. Examples of recent research papers on hyper-heuristic approaches to exam timetabling include [2, 5].

This paper explores adaptive techniques, which are relatively new in timetabling. In [4], a methodology was developed to adaptively order the exams by how difficult they were in the previous solution's construction. At each iteration, those exams which contributed to costs greater than a certain threshold were assigned increased difficulty values and were ordered and scheduled earlier to construct solutions. This simple adaptive approach was very efficient on the benchmark problems tested. It was based upon the "squeaky wheel" optimisation technique [11] which was originally applied to both scheduling and graph coloring problems.

The basic idea of decomposition is to "divide and conquer", as (near) optimal solutions may be obtained more easily for smaller sub-problems using relatively simple approaches [8]. However, the task of decomposing the problem is challenging and problem specific. Determining how the sub-solutions obtained can be combined for the original problem also represents a key issue. There are very few papers in the timetabling literature which have investigated decomposition techniques. In [6], a look ahead mechanism in a multi-stage approach considered two sub-problems at a time. The methodology of [6] significantly reduced the computational time and improved the solution quality on the addressed exam timetabling benchmark. In [9], a clique of the graph that modelled the timetabling problems is first obtained and used to generate solutions. This can be seen as decomposing the problems into two parts, which are used one after another to construct solutions.

2 The Adaptive Decomposition and Construction Approach

In our adaptive decomposition approach, an initial ordering of the exams is first obtained by the Saturation Degree graph heuristic (which orders the exams by the number of remaining feasible timeslots during the solution construction). In some cases, the ordering may need to be adjusted by randomly swapping two exams in the ordering until a feasible solution can be obtained. Based on this initial ordering, this list of exams is decomposed adaptively into two subsets (called the *difficult* and the *easy set*) by a two-stage approach, which is described next.

In a constructive approach, if the assignment method that schedules exams into the timeslots is fixed, then the problem can be seen as being transferred into an ordering (permutation) problem. This problem has a search space of size $e!$, (where e is the number of exams) and thus is much larger than that of the original problems (of size t^e, where t is the number of timeslots). The adaptive decomposition approach decomposes the original ordering problem into two smaller ordering sub-problems and thus the size of the search space is significantly reduced.

2.1 Adaptive Detection on the *Difficult Set*

In the first stage, both the exams and their ordering in the *difficult set* are adaptively adjusted by using information derived from the solution construction that was observed in previous iterations. It is an iterative process where, at each iteration, the exams in both the *difficult set* and the *easy set* are used to construct a complete solution, whose quality is used to identify problematic exams. The aim is to collect these troublesome exams into the *difficult set*, which can be dealt with with higher priority in future iterations of solution construction. Thus better solutions can be generated.

The pseudocode presented in *Algorithm 1* outlines the process. In each iteration, ordering of the exams in the *difficult set* is adjusted to find the best order that produces an improved or feasible solution. If no feasible solution can be generated using the current ordering, then the exam which cannot be scheduled is moved forward in the *difficult set* ordering, and the *difficult set* is reduced to include only the exams before this exam. Initial tests showed that moving the exam 2 to 6 positions forward obtained good results thus 5 is selected in the approach. If a feasible or improved solution is obtained, then this order of exams is kept and the *difficult set* is increased to include the first exam in the *easy set* to detect more potentially difficult exams iteratively. The optional steps in *Algorithm 1* are only taken in the experiment presented in Section 3.2 for comparison purpose.

Algorithm 1: DIFFICULT_SET_DETECTION()

```
build the initial ordering of exams by Saturation Degree
initial size of difficult set Sd = number of exams / 2
MaxNoIterations = 10,000; iteration = 0
while iteration<MaxNoIterations
do {
    easy set = {e_{Sd+1}, e_{Sd+2}, ... e_e}
    reorder the exams in the difficult set {e_1, e_2, ... e_{Sd}}
    construct a solution using ordered exams in both difficult and easy sets
    //optional step: calculate costs of exams in the constructed solution
    //optional step: move forward the exam incurring the highest cost
    if a feasible solution or an improved solution is obtained
        Sd = Sd + 1 // include more potential exams in the difficult set
      else
        move forward the difficult exam causing the infeasibility
        re-set the size of the difficult set to where the exam was
    store the difficult set and its size S_best if the best solution is obtained
    iteration = iteration + 1
```

2.2 Ordering in the *Easy Set*

In the second stage, after the adaptive decomposition, the potentially difficult exams and their ordering in the *difficult set* are fixed. The exams in the *easy set* are then reordered to further improve the quality of solutions built by scheduling the ordered exams one by one in both the *difficult set* and the *easy set*.

In most decomposition approaches in the literature [6, 8], the sub-solutions obtained need to be carefully combined together. This particularly concerns global information pertaining to the original problem. In our adaptive decomposition approach, a solution is constructed by ordering the exams in one set while keeping the other set fixed. It thus deals with the global problem information as both sets are concerned while solving the sub-problems, thus no adaptation needs to be made. Also the *difficulty* of exams is adaptively obtained (rather than by using a fixed measure) based on online information from the previous solution quality value for the problem in-hand.

3 Experimental Results on the Benchmark Data Set

We carried out a set of experiments to analyse the adaptive decomposition and construction approach. We were particularly concerned with the effects of the decomposition, the ways the *difficult set* is adjusted, the relationship between the two sets, and the effects that the *difficult set* has on the overall solution quality. They are presented in the next sub-sections, respectively. For each experiment, five runs were carried out, from which both the average and best results are reported to give a better evaluation of the approaches. All the approaches in different experiments were run for the same number of iterations for fair comparisons. The results obtained by our approach, when comparing with the state-of-the-art approaches, indicate the efficiency, simplity and generality of the adaptive decomposition approach (see section 3.5). The coding is in C++ and experiments were carried out on a PentiumIV 3GHZ machine with 1G memory. Computational time is reported here only for comparisons between variants of the adaptive decomposition approach. Most of the approaches in the literature did not report the computational time as it is impossible to compare it across different platforms. Also, time is not a crucial issue in real world circumstances as usually the timetables are built weeks or months before they are utilised. The version of the exam timetabling problem that we tackle is that introduced in [9] and the naming conventions for the datasets are those introduced in [15].

3.1 Analysis on the Adaptive Decomposition

In the first set of experiments, the effect of decomposition in the adaptive approach is evaluated. We developed an adaptive approach without decomposition. That is, a single set of ordered exams is adjusted adaptively to construct solutions. At each iteration, the exam causing infeasibility is moved forward. Note that this approach is different from the adaptive ordering approach developed in [4], where at each iteration a subset of exams with costs higher than a certain threshold are given a higher priority in the next iteration. Different thresholds were thus tested in [4]. In this study, we try to keep the approach simple by introducing the least number of parameters.

This approach is compared with the adaptive decomposition approach presented in Section 2 and results are given in Table 1. It can be seen that the adaptive decomposition approach performs significantly better than its variant without decomposition. On all of the 11 problems, the improvement on solution quality ranges from 0.69% - 11.02%. This indicates that the adaptive decomposition can effectively decompose the problems in hand and produce better results.

Table 1: Average and best results from the adaptive approach with (upper I) and without (lower II) decomposition on benchmarks. (the best and average results are highlighted in bold; improve % = (without-with)/with

	car91 I	car92 I	ear83 I	hec92 I	kfu93 I	lse91 I	sta83 I	tre92 I	ute92 I	uta93 I	yor83 I
I avg	5.47	4.7	39.14	12.21	15.43	11.66	162.44	9.18	28.03	3.6	45.1
best	**5.38**	**4.53**	**36.76**	**11.45**	**14.79**	**11.2**	**157.4**	**8.83**	**26.87**	**3.53**	**42.04**
time (s)	3104	2140	126	32	185	148	47	192	39	2736	126
II avg	5.55	4.66	41.7	12.53	16.86	12.26	158.6	9.17	28.43	3.71	43.84
best	5.53	4.59	41.7	12.26	16.2	12.05	157.77	8.89	27.81	3.68	42.12
time (s)	2993	2020	128	32	191	180	32	183	30	2616	110
improve %	1.46	1.57	11.02	2.95	11.01	7.92	0.69	0.99	2.52	2.9	0.93

3.2 Analysis on the Construction of the *Difficult Set*

In the second set of experiments, a variant of the adaptive decomposition approach was developed. Not only is the exam (from the *difficult set*) that incurs an infeasibility, but also the exam that incurs the highest cost in the timetable constructed moved forward. Here, we implement the optional steps in *Algorithm 1* to see what benefit this could have in the adaptive decomposition approach. The above two ways of adjusting the *difficult set* are compared and presented in Table 2. The results obtained in both the 1^{st} stage (by detecting the *difficult set*) and the 2^{nd} stage (by ordering the *easy set*) are also presented. The computational time for the adaptive decomposition is the same as that in Table 1 so is not presented again in Table 2.

Table 2: Average and best results from the adaptive approach forwarding only exams causing infeasibility (upper "I") and also exams incurring the highest cost (lower "III"). "1^{st}" and "2^{nd}" present the results obtained after stage 1 and stage 2, respectively. "improve %" = $(1^{st}$-$2^{nd})/1^{st}$

	car91 I	car92 I	ear83 I	hec92 I	kfu93 I	lse91 I	sta83 I	tre92 I	ute92 I	uta93 I	yor83 I
I 1^{st} avg	5.59	4.7	39.54	12.21	15.43	11.66	162.44	9.18	28.03	3.67	45.1
2^{nd} avg	5.47	4.68	38.43	12.18	15.19	11.36	157.51	9.08	27.73	3.6	43.1
improve %	2.2	2.4	2.8	2	1.6	2.6	3	1.1	1	1.9	3.7
best	5.38	**4.53**	**36.76**	**11.45**	14.79	11.2	**157.4**	8.83	**26.87**	3.53	**42.04**
III 1^{st} avg	5.5	4.69	41.2	12.21	15.5	11.5	158.85	9.08	27.95	3.68	45.25
2^{nd} avg	5.38	4.59	40.66	12.2	15.3	11.31	157.67	9.01	27.72	3.62	43.72
improve %	2.2	2.3	1.3	0.1	1.3	1.7	0.7	0.8	0.8	1.58	3.4
best	**5.32**	**4.53**	38.59	11.75	**14.63**	**10.96**	157.51	**8.81**	26.88	**3.51**	42.16
time (s)	3203	2198	138	36	185	148	42	203	36	2980	154

We can see that these two approaches perform quite similarly on the benchmark problems (i.e. they obtained better results on 5 problems, respectively, and the same results on 1 problem). This indicates that considering exams of the highest cost does not improve the approach. Also, more computational time is required in the variant as, at each iteration, the costs of exams need to be re-calculated. It can also be observed that in both of the variants, reordering the *easy* exams also contributes to better solutions based on the *difficult set* obtained from the 1^{st} stage.

3.3 Analysis on the *Difficult Set* vs. the *Easy Set*

We further investigate the relationship between the *difficult set* and the *easy set* in the adaptive decomposition approach. Instead of using a distinct boundary between the two sets, in the 2^{nd} stage, the *easy set* will also include the 2^{nd} half of the *difficult set*. That is, the *difficult set* and the *easy set* have some overlapping exams (i.e. the exams in $\{e_{S_{best}/2}, e_{S_{best}/2+1}, \dots e_{S_{best}}\}$ are also considered in the 2^{nd} stage, S_{best} is obtained from *Algorithm 1*).

Table 3: Average and best results from the adaptive decomposition approach with (lower *IV*) and without (upper *I*) and overlapping between the *difficult set* and the *easy set*.

	car91 I	car92 I	ear83 I	hec92 I	kfu93 I	lse91 I	sta83 I	tre92 I	ute92 I	uta93 I	yor83 I
I avg	5.47	4.7	39.14	12.21	15.43	11.66	162.44	9.18	28.03	3.6	45.1
best	**5.38**	4.53	36.76	11.45	14.79	11.2	157.4	8.83	26.87	**3.53**	**42.04**
IV avg	5.5	4.58	37.85	12.09	15.16	11.31	157.55	8.98	27.6	3.61	43.27
best	5.45	**4.5**	**36.15**	**11.38**	**14.74**	**10.85**	**157.21**	**8.79**	**26.68**	3.55	42.2

It can be clearly seen from Table 3 that considering overlapping exams in the *easy set* contributes to a better performance for the approach. It obtained better solutions on 8 out of 11 problems. This indicates that some exams could be considered as both *easy* and *difficult* in the problem, and there is no distinct boundary between the two subsets. Computational time is almost the same for all the problems (thus it is not presented in Table 3).

3.4 Contributions of the *Difficult Set* to the Overall Solution Quality

Based on the above experiments, we finally analyse the exams in the *difficult set* detected by the above different adaptive approaches and evaluate their contribution to the overall cost of the solutions generated. Table 4 presents these evaluations.

Table 4: Average size ("size %") of the *difficult set* in different adaptive approaches. (I: forward exams causing infeasibility; III: I + forward exams of the highest cost; IV: I + overlapping *difficult set* and *easy set*. "match %" presents the number of exams in the *difficult set* that matches the first 50% of exams with the highest cost. "cost %" presents the summed cost occurring from the exams in the *difficult set*.

		car91 I	car92 I	ear83 I	hec92 I	kfu93 I	lse91 I	sta83 I	tre92 I	ute92 I	uta93 I	yor83 I
	size %	32	23	38	61	23	14	15	46	37	22	44
I	match %	89	88	83	73	99	98	32	86	66	98	68
	cost %	66	62	60	71	83	58	12	79	57	66	47
	size %	33	25	44	70	27	18	43	47	24	51	43
III	match %	88	91	75	65	99	94	40	84	97	64	65
	cost %	66	64	64	72	87	65	30	80	67	67	43
	size %	32	23	38	61	23	14	15	46	37	22	44
IV	match %	87	87	79	71	91	95	49	83	63	98	63
	cost %	65	62	59	70	78	58	23	79	54	66	46

First, we obtain the size of the *difficult set* with respect to the overall size of the original problem. It can be seen that, in all cases except "hec92 I", less than half of the exams are detected as *difficult* in the problem. The problems with the largest *difficult set* are "hec92 I", "tre92 I" and "yor83 I", which are actually the most difficult problems, in the sense that a feasible solution cannot be obtained using a pure Saturation Degree. It is also observed that the sizes of the detected *difficult set* by these different adaptive approaches are consistent, indicating that these approaches are adapting appropriately to very different problems.

To evaluate how important the exams are in the *difficult set*, we first order (in a descending manner) all the exams $e_1 > e_2 > ... > e_e$ by their costs in the best solution generated. Then the first 50% of the exams with the highest costs are checked to see if they are also detected as difficult exams in the *difficult set* (i.e. we check the membership of $\{e_1, e_2 ... e_{e/2}\}$). It can be seen that in all the problems, up to 98% of the difficult exams are contributing to the highest costs in the solutions. The exception is sta83 I, where only 12% - 30% of the difficult exams are incurring the highest costs.

Whilst the above evaluation quantitively indicates the difficulty of the exams in the *difficult set*, we further evaluate qualitively the difficulty of these exams. We calculated the summed cost incurred from the exams in the *difficult set* to the total cost of the solutions. In most cases, the cost from the *difficult set* contributes to 54% - 87% of the total cost except in the case of "sta83 I" and "yor83 I". Note that this should be read in conjunction with the size of the *difficult set*, which, in most cases, is much smaller, indicating that in most problems, ***a small set of difficult exams contributes to a large portion of the cost of the solutions***.

It is interesting to see that for problems "hec92 I", "sta83 I" and "yor83 I", roughly the same proportion of difficult exams contributes to the same proportion of cost and the difficult set (i.e. percentages across "size %", "match %" and "cost %" are close to each other for each problem). This indicates that almost all the exams in these problems are equally important. Furthermore, it also explains why forwarding the highest costs in approach III does not contribute to better performance of the adaptive approach. Further study on these problems in comparison with the others may reveal other interesting observations.

3.5 Comparisons with State-of-the-art Approaches

During the years, a number of approaches have been developed to solve these benchmark exam timetabling problems. We present the best results (taken from [15]) in Table 5 and evaluate our adaptive decomposition method (ADA) against these best results and against the adaptive method of [4].

Table 5: Best results from the adaptive decomposition approach with overlapping exams (ADA) and the best reported from the state-of-the-art approaches on benchmark exam timetabling problems.

	car91 I	car92 I	ear83 I	hec92 I	kfu93 I	lse91 I	sta83 I	tre92 I	ute92 I	uta93 I	yor83 I
ADA	5.45	4.5	36.15	11.38	14.74	10.85	**157.21**	8.79	26.68	3.55	42.2
[4]	4.97-5.55	4.32-4.68	36.16-38.55	11.61-12.82	15.02-16.5	10.96-12.53	161.91-170.53	8.38-8.96	27.41-29.67	3.36-3.6	40.77-42.97
best	4.2	4.0	29.3	9.2	13.46	9.6	157.3	8.13	24.21	3.2	36.11

By comparing our adaptive decomposition approach with that of the adaptive ordering approach developed in [4], we can see that in the 11 problems, our approach obtained better results on 5 problems, and similar results (i.e. within the range of results from [4]) for the other problems. Note that in [4] a number of variants were tested by adjusting the difficulty values added to the measure of exams in different ways. Also a threshold needs to be chosen to decide which exams need to be moved forward. Furthermore, in each iteration, the difficulty of all of the exams needs to be re-calculated, based on which an ordering can be set. Our approach does not involve these calculations on each exam in each iteration and thus is much simpler.

For the benchmark problems, different approaches in the literature worked particularly well on specific instances (but not on all of them). These best results are summarised in [15]. Compared with all the other approaches, our adaptive decomposition approach obtained competitive results which are in the range of the best reported. In particular, for problem "sta83 I", our approach obtained the best result (157.21) reported in the literature. Note that our approach is a simple pure constructive approach and does not reply on initial solutions and fine tuning of a number of parameters, which are crucial to the success of most approaches.

4 Conclusions and Future Work

This paper develops an adaptive decomposition and ordering approach which constructs solutions for exam timetabling problems. The original problems are decomposed adaptively into two subsets, the *difficult set* and the *easy set*. They are used to construct solutions by adjusting the orderings of the exams in one set while fixing the other. The approach integrates both the adaptive and the decomposition techniques and has links to a number of approaches developed in the field. Our first aim is to decompose the complex timetabling problems into small problems which are easier to handle. Another aim is to automatically integrate adaptive mechanisms into the simple constructive technique when dealing with different problems. Current non-adaptive mechanisms are usually specially designed and hard coded in most of the meta-heuristic approaches in the literature [15].

It is observed by experimentation that the potentially difficult exams detected by our approach usually represent a small proportion of the original problems but contribute to a large proportion of the costs of the solutions. The difficulty of the exams is adaptively adjusted during the problem solving rather than using static or pre-defined methods. It is observed that ordering the exams in the *easy set* also contributed to the generation of better solutions.

The comparisons of this adaptive decomposition approach with the state-of-the-art approaches indicate that it is a simple yet effective technique. It is also a general technique which can be adapted to quickly construct good quality solutions to any problems which can be solved using heuristic ordering strategies.

There are also exceptions detected by the adaptive decomposition approach i.e. that, in some cases, almost all exams are equally difficult when ordered and used to construct solutions. This requires further study. Also, the *difficult set* and the *easy set* do not have a distinct boundary. Future work will study more learning and classification techniques for detecting the *difficult set* more accurately. It will also be interesting to study more elaborate ordering methods on both the *difficult set* and the *easy set*.

References

[1] Ahuja R.K., Orlin J.B. and Sharma D. (2000). Very Large Scale Neighbourhood Search. *International Transactions in Operational Research* **7**: 301-317.

[2] B. Bilgin, E. Ozcan and E.E. Korkmaz. (2006) An experimental study on hyper-heuristics and exam timetabling. Proceedings of the 6th International Conference on Practice and Theory of Automated Timetabling. 123-140.

[3] Burke E., Hart E., Kendall G., Newall J., Ross P. and Schulenburg S. (2003). Hyper-heuristics: an Emerging Direction in Modern Search Technology. In: Glover F. and Kochenberger G. (eds). *Handbook of Meta-Heuristics*, Kluwer 457-474.

[4] Burke, E.K. and Newall, J. (2004). Solving Examination Timetabling Problems through Adaptation of Heuristic Orderings. *Annals of Operational Research*, **129**:107-134.

[5] Burke E.K., McCollum B, Meisels A., Petrovic S. and Qu, R. (2007). A Graph-Based Hyper Heuristic for Timetabling Problems. *European Journal of Operational Research*, **176**: 177-192, 2007.

[6] Burke E.K. and Newall J. (1999). A Multi-Stage Evolutionary Algorithm for the Timetabling Problem. *The IEEE Transactions on Evolutionary Computation.* **3**(1): 63-74.

[7] Burke E.K. and Petrovic S. (2002). Recent research directions in automated timetabling. *European Journal of Operational Research*, **140**(2): 266-280.

[8] M.W. Carter. (1983). A decomposition algorithm for practical timetabling problems. Technical Paper 83-06, Department of Industrial Engineering, University of Toronto.

[9] Carter M., Laporte G. and Lee S. (1996). Examination Timetabling: Algorithmic Strategies and Applications. *Journal of Operational Research Society*, **47**: 373-383.

[10] Hansen P. and Mladenovic N. (2001). Variable Neighbourhood Search: Principles and Applications. *European Journal of Operational Research*, **130**, 449-467.

[11] Joslin D.E. and Clements D.P. (1999). "Squeaky Wheel" Optimization. *Journal of AI Research*, **10**: 353-373.

[12] Merlot L., Boland N. Hughes B. and Stuckey P. (2002). A Hybrid Algorithm for the Examination Timetabling Problem. In: Selected Papers from PATAT02, Lecture Notes in Computer Science 2740, 207-231.

[13] McCollum B.G.C. (2006). Bridging the Gap between Research and Practice. Proceedings of the 6th International Conference on Practice and Theory of Automated Timetabling, (Keynote) 15-35. August 2006, Czech Republic.

[14] Nonobe K., Ibaraki T. (1998). A Tabu Search Approach to the Constraint Satisfaction Problem as a General Problem Solver. *European Journal of Operational Research*, **106**: 599-623.

[15] Qu R., Burke E.K., McCollum B., Merlot L.T.G. and Lee S.Y. (2006). A Survey of Search Methodologies and Automated Approaches for Examination Timetabling. Technical Report NOTTCS-TR-2006-4, School of CSiT, University of Nottingham.

[16] Schaerf A. (1999). A Survey of Automated Timetabling. *Artificial Intelligence Review.* **13**(2): 87-127.

[17] Thompson J. and Dowsland K. (1998). A Robust Simulated Annealing Based Examination Timetabling System. *Computers & Operations Research*, **25**: 637-648, 1998.

[18] White G.M., Xie B.S. and Zonjic S. (2004). Using Tabu Search with Longer Term Memory and Relaxation to Create Examination Timetables. *European Journal of Operational Research*, **153**(16): 80-91.