In, L.Wang, K. Tan, C. Furuhashi., J-H Kim, and X. Yao (Eds.), Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution And Learning (SEAL2002), Singapore; (pp 662-666). (2002) (ISBN 981-04-7523-3)

COMBINING HUMAN AND MACHINE INTELLIGENCE TO PRODUCE EFFECTIVE EXAMINATION TIMETABLES*

Peter Cowling[1], Samad Ahmadi[2,1,3], Peter Cheng[3] and Rossano Barone[3,2,1]

[1] MOSAIC research group, Dept. of Computing, Univ. of Bradford, Bradford BD7 1DP, UK
[2] ASAP research group, Sch. of Comp. Sci. & IT, Univ. of Nottingham, Nottingham NG8 1BB,UK,
[3] CREDIT research group, Sch. of Psychology, University of Nottingham, Nottingham NG7 2RD, UK,
Peter.Cowling@scm.brad.ac.uk, S.Ahmadi@cs.nott.ac.uk, rb@psychology.nott.ac.uk, Peter.Cheng@nottingham.ac.uk

ABSTRACT

In this paper we present a system for producing good examination timetables, by integrating the search capabilities of computer heuristics and the cognitive capabilities of timetabling users. We allow users to view and easily edit a cognitively manageable representation of each timetable, using the STARK (Semantically Transparent Approach to Representing Knowledge) approach. Further, we allow users to directly control the heuristics which are used to automatically generate solutions, using the HuSSH (Human Selection of Scheduling Heuristics) approach. We present experiments which show that using these two approaches in combination can lead to a very effective system for examination timetabling, even in cases where model inaccuracies mean that simply optimizing the objective function does not provide adequate solutions. Our approach may be generalized across other domains where optimization decision support is used.

1. INTRODUCTION

Heuristics have proven to be effective at producing good solutions to a wide range of problems in planning, scheduling and optimization, especially where solutions must be generated quickly, or computational complexity results suggest that optimal solutions may be difficult or impossible to find in practice [5]. However, since it is difficult or impossible to capture all problem details, it is often the case that solutions may need considerable editing before they may be used in practice, and many decision support systems support this solution editing process [4]. This editing process is nearly always at the level of final solution, or initial model. It is rare that users are given the tools to interrupt the solution process and edit the heuristics which are then used to produce an improved solution. We believe, from past experience [7], that users, due to their extensive knowledge of the problem being solved, can quickly build a surprising level of competence in handling complex parameters of the solution process. Indeed, many legacy systems are used in exactly this way in practice, since the original model for which these systems was designed is no longer directly applicable. As the size and complexity of problems and systems increases, it becomes impractical for a user to manually "edit out" all of the problems at the level of solution, and it would be useful if the user could instead "steer" the heuristic process towards desirable outcomes.

In this paper we present a system which allows this type of user intervention. It provides a STARK interface which has been developed using principles of representational design from the cognitive science community [6] and supports solution-level editing using this interface. In addition to this, our system provides a HuSSH interface which allows users to directly interact with the heuristics used to produce solutions, by modifying parameters and choosing heuristics which are used to build timetables and dismantle areas of the solution where users wish to find improvements. We present experiments which show that our system is effective, both when the model is assumed to completely reflect the preferences of the user, and when the model misses important user preferences.

2. EXAMINATION TIMETABLING

The Examination timetabling problem is a difficult problem in practice, which must be solved at least once a year in every school and university. It is usually modeled as an NP-hard combinatorial optimization problem [5]. The problem demands that a given number of exams are scheduled in a limited number of periods and venues in such a way that no student will have more than one exam at a time and other constraints are satisfied. The objective is generally to allocate timeslots, rooms and other

* Research supported by ESRC/EPSRC grant L32853012 under the PACCIT programme, in collaboration with OPTIME.

In, L.Wang, K. Tan, C. Furuhashi., J-H Kim, and X. Yao (Eds.), Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution And Learning (SEAL2002), Singapore; (pp 662-666). (2002) (ISBN 981-04-7523-3)

resources to examinations so as to minimize some measure of constraint violation. Different institutions have a wide variety of different types of constraints and goals with different levels of importance.

Conflicting objectives and the changing set of constraints in different institutions makes the examination timetabling problem very challenging. A simplified model of this problem considered in the literature is the graph coloring problem [10]. Nevertheless, capturing *all* user preferences in an exact model for the problem is difficult (if not impossible). Even when a computerized scheduling system is available, it is normally necessary for the examinations officer to modify the automatically generated solution in order to produce and adequate solution. It is thus useful that the automated scheduler has some level of flexibility to handle new constraints and to incorporate user knowledge.

This problem has been tackled with different heuristic, optimisation and metaheuristic algorithms. Burke *et.al.* [3] used a multicriteria approach to solve the problem in several phases. Thompson and Dowsland [9] used simulated annealing with Kempe chain neighbourhoods to preserve the feasibility of the solutions in terms of first order clashes during local moves. Ross *et.al.* [8] analysed the behaviour of genetic algorithm on different instances of the problem. Arani and Lotfi [1] investigated a three phase process using the quadratic assignment problem, the set covering problem and the travelling salesman problem in different phases respectively. Further references and a general survey of the problem are in [5].

3. EXAMINATION TIMETABLING HEURISTICS

Sequential heuristics and clustering heuristics are the two major categories of constructive heuristics for examination timetabling. The order in which exams are scheduled, and the order in which periods and rooms are chosen for each exam, has a highly significant effect on the characteristics and quality of the resulting timetable. We propose the following heuristics for ordering exams, periods and rooms:

3.1. Exam selection heuristics

1. **Intersections:** The unscheduled exam with the highest number of students in common with other scheduled and unscheduled exams will be scheduled next. We use a modified version of the largest degree first heuristic for graph colouring.
2. **Restrictions:** Some of exams are restricted to be held at pre-specified periods or venues. Dealing with such exams in the later stages of the scheduling process may create problems due to usage of their corresponding rooms and periods for other exams. In this heuristic we prioritise restricted exams to be scheduled first.
3. **Available Periods:** This heuristic dynamically finds the number of available periods where an exam can be scheduled without penalty each time another examination is scheduled. Each iteration, the unscheduled exam with the smallest number of available periods is selected to be scheduled next. This is a generalisation of least saturation degree for the graph colouring problem [2] where we consider constraints other than clashes for checking the availability of each period.
4. **Available resources:** This heuristic is similar to 3. above, but considers the number of periods *and rooms* where an exam may be scheduled without penalty.
5. **Combination:** This heuristic combines the previous four heuristics in two ways. First, the four above heuristics are prioritised and where two exams are tied using one of the heuristics, this tie will be broken using the next lower priority heuristic. Second, to deal with complex coincidence and concurrency constraints where a group of exams needs to be scheduled at the same time and/or place, an extra priority value is added to the other related exams after each member of a coincident/concurrent group is scheduled. In this way we aim to schedule all members of a group together before other scheduled exams make this impossible.
6. **Random:** A random heuristic is placed in the pool of heuristics to examine the ability of the user to identify heuristics of low quality, and to provide a mechanism for diversification.

3.2. Period selection heuristics

After selection of an exam *e* for scheduling, we select a period which minimises the penalties associated with violations of constraints with *e*. Our approach uses a general penalty function which finds a period to minimize the sum of weighted combination of clashes, consecutive exams, order constraint violations, size violations, duration excesses, pre-specified rooms violations and period violations.

1. **Penalty-based:** In this heuristic for a given exam *e*, the potential penalty of assigning *e* to each period is calculated and the period with the minimum penalty is selected. Ties are broken arbitrarily, with the sorting algorithm in use tending to allocate exams to periods early and late in the schedule, which empirical evidence shows to be more effective than random allocation.
2. **Random selection of period.**

3.3. Room selection heuristics

In, L.Wang, K. Tan, C. Furuhashi., J-H Kim, and X. Yao (Eds.), Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution And Learning (SEAL2002), Singapore; (pp 662-666). (2002) (ISBN 981-04-7523-3)

After selection of an exam, different periods are examined for availability of rooms in the order induced by the period heuristic. For each period, an ordered list of permitted rooms for this exam is created, based on the following heuristics:

1. **Best fit:** this heuristic will find the room with the smallest amount of remaining capacity into which the exam will fit.
2. **Largest-first:** in this heuristic priority is given to fill the largest spaces available (sports halls, large lecture theatres). This policy is reasonable in the context of optimising usage of large spaces to minimise the number of venues and invigilators.
3. **Random selection of room.**

3.4. Unscheduling heuristics

In order to remove specific "problems" from a timetable, or partial timetable, the user can choose to automatically unscheduled exams using two heuristics, so that they can be rescheduled using manual approaches or using any of the heuristics above:

1. **Penalty values:** Exams in an order based on the weighted sum of their penalty value are unscheduled. The user specifies the percentage of exams to be unscheduled. By adjusting weights, the user has a useful tool to remove specific types of violations.
2. **Exams in a Period:** Here all the exams in a specific period will be unscheduled. This, in conjunction with a constructive scheduling heuristic is a useful tool for shuffling exams in a period to find a better fit and removing capacity violations.

3. EXPERIMENTAL FRAMEWORK

Two experiments were performed. In the first experiment, we made the assumption that the model was entirely correct, and so the users' goal was to use STARK and HuSSH to provide a solution which yielded the best possible objective function value. In the second of experiment, we tried to assess the ability of the user and the system in incorporating un-modeled constraints. The four subjects of the experiment were from optimisation and cognitive science backgrounds. Each experiment used a data set of the University of Nottingham with 800 exams, 33998 enrolments and 7896 students, available from ftp://ftp.cs.nott.ac.uk/ttp/Data/Nott94-1.

After an introduction to the system, each subject was asked to generate a solution using any combination of heuristics in HuSSH with user interventions at any point using the unscheduling facilities of HuSSH or manual modifications using STARK. A total of 90 minutes was allowed for this experiment. Then each subject took a break before continuing with the second experiment, where, six exams were to be scheduled in the early part of the week, without having any two of them scheduled in the same day. This corresponds to the situation, which is considered in some Universities, where the scheduler has knowledge about the difficulty of certain exams based on students' performance in previous years and plans to hold the exams after a weekend. A maximum of 30 minutes was allowed for each subject to perform the second experiment.

We report quantitative and qualitative results for each subject in the first experiment. We report qualitative results only for each subject in the second experiment, where we do not have a numerical performance measure.

4. COMPUTATIONAL RESULTS

Graphs 1a to 4b summarize the results of experiment one for each of the four subjects. To clarify later changes to the solution, which cannot be seen at the original scale, the second graph of each pair zooms in on the most "interesting" section of the experiment. The total number of timetabled exams and objective function value are plotted against time, expressed as a percentage of the total experiment time of around 90 minutes.

The graphs show that the four subjects used widely different strategies. Figs. 1a, and 2a show a strategy where the subjects used an automated approach to produce a first "draft" schedule of all exams. Figs. 3a and 4a show a combination of automatic and manual approaches to produce the first complete timetable. In terms of objective function and CPU time it is clear that the first strategy is more effective. However, analysis of comments made by the subjects illustrated in figs. 3a and 4a show a high level of "engagement" with the solution process and a modification of mental models to satisfy the experimental goal of objective minimization, to propose modifications to add to model realism, and in response to the perceived priority and difficulty of constraints. For example, manual fixing of consecutive exams was perceived as a very high priority by the subject illustrated in fig. 3a. This subject's comments show that upon further investigation and probing using STARK and HuSSH tools, it was accepted that a number of consecutivity constraints were impossible to remove for this problem instance, which greatly improved the subject's confidence in the solution finally produced. Feedback from users shows a steep learning curve of behaviour of the heuristics and the model as time proceeds.

Figs. 1b, 2b, 3b and 4b illustrate the methods employed by the subjects to improve on the "draft" timetable. Again we see a variety of strategies. User adjustment of penalty weights, especially when unscheduling to fix a particular set of constraint violations, was shown to be useful by all subjects. However, adjustment of penalty weights when rescheduling gave rise to a relatively poor solution in figure 3b. Fig. 2b shows a very sophisticated use of the

STARK and HuSSH approaches. Further investigation of the final timetable found in this case show that the solution produced is optimal, and the nonzero penalty is due to inconsistent constraints. In all cases an acceptable solution was produced. Generally the figures show that unscheduling-rescheduling was effective when applied to small sets of exams, with manual enhancement where necessary.

In the second experiment all the subjects succeeded in scheduling the six selected exams in the first periods of the week without having two of them at the same day. The subjects' comments reveal a high level of satisfaction with the solutions produced. Subjects were again able to critically evaluate the model as solutions were produced. For example, one subject created a solution with a preference for the second week over the first, to allow students an extra week's revision. All subjects expressed a high level of confidence in the quality of the solution created.

One of the additional benefits of giving expert users access to such a wide range of solution tools is that model improvements, and new heuristics may be identified. One subject observed that the relative importance of clashes and consecutive exams were incorrect in the model, and that the unscheduling approach ought to consider the ease of rescheduling of exams when deciding the order for unscheduling. Another observed the effectiveness of the heuristic which unscheduled whole periods and then rescheduled them using a combination of automatic and manual approaches.

5. CONCLUSIONS

In this paper we have presented a combination of a tool for effective representation and editing of examination timetables (STARK), and a tool to allow users to directly control the heuristic solution process (HuSSH). Experiments showed that this approach was very effective at producing timetables which satisfied the objective and subjective criteria of users, and provided the user with an effective means of dealing with inaccuracies and oversimplifications in the model used. In addition, our experiments show that the combination of these two approaches allow users to consider possible enhancements to the model, and to suggest possible new heuristic approaches which might be partially or completely automated. This last point is one which we are actively pursuing in this project.

6. REFERENCES

[1] T. Arani, V. Lotfi, "A Three Phased Approach to Final Exam Scheduling," *IIE Trans. 21* pp. 86–96, 1989.

[2] E.K. Burke, J.P. Newall, "A Multi-stage Evolutionary Algorithm for the Timetable Problem," *IEEE Trans. Evol. Comput. 3*, pp. 63–74 , 1999.

[3] E.K. Burke,Y. Bykov and S. Petrovic, "A Multi-Criteria Approach to Examination Timetabling", in selected papers from PATAT 2000, *Springer LNCS vol. 2079*, pp. 118-131, 2001.

[4] M.W. Carter, "A comprehensive course timetabling and student scheduling system at the University of Waterloo" in Selected papers from PATAT 2000, *Springer LNCS vol. 2079*, pp. 64-82, 2001.

[5] M.W. Carter, G. Laporte, "Recent Developments in Practical Examination Timetabling," in Selected papers from PATAT'95, *Springer LNCS vol. 1153*, pp. 3-21, 1996.

[6] P. C-H. Cheng, R. Barone, P.I. Cowling, S. Ahmadi, "Opening the information bottleneck in complex scheduling problems with a novel representation: STARK diagrams". To appear in proceedings of Diagrams 2002, *Springer LNAI*, 2002.

[7] P.I. Cowling "Design and Implementation of an Effective Decision Support System: A Case Study in Steel Hot Rolling Mill Scheduling", in: *Human Performance in Planning and Scheduling*, Taylor & Francis, pp. 217-230, 2000.

[8] P. Ross, E. Hart, D. Corne, "Some Observations about GA based Timetabling". in Selected papers from PATAT'97, *Springer LNCS vol. 1408*, pp. 115-129, 1998.

[9] J.M. Thomson, K.A. Dowsland, "Variants of simulated annealing for the examination timetabling problem", *Ann. Operational Research*, pp 105-128, 1996.

[10] D. J. A. Welsh and M. B. Powell. "An upper bound for the chromatic number of a graph and its application to timetable problems". *The Computer Journal vol. 10,* pp.85–86, 1967.

In, L.Wang, K. Tan, C. Furuhashi., J-H Kim, and X. Yao (Eds.), Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution And Learning (SEAL2002), Singapore; (pp 662-666). (2002) (ISBN 981-04-7523-3)