Applying Ahuja-Orlin's Large Neighbourhood for Constructing Examination Timetabling Solution

Extended Abstract

S. Abdullah[1], S. Ahmadi[2], E.K.Burke[1], M. Dror[3]

[1] Automated Scheduling, Optimisation and Planning Research Group, School of Computer Science & Information Technology, University of Nottingham, Jubilee Campus, Wollaton Road, Nottingham NG8 1BB, United Kingdom
[2] School of Computing, De Montfort University, The Gateway, Leicester LE1 9BH, United Kingdom
[3] MIS Department, College of Business and Public Administration, University of Arizona, Tucson, Arizona 85721, USA

1. Introduction

Examination timetabling has been defined as the problem of assigning a set of exams into a limited number of time slots for a given set of resources [1]. A great variety of solution approaches to examination timetabling problems have been described and discussed in the literature and tested on real and artificial data. Carter [2] classified four major solution approaches in examination timetabling arena: sequential methods [3], cluster methods [4], constraint-based methods [5], and meta-heuristics [6,7,8]. This was extended to multicriteria approaches, case-based reasoning approaches, and hyper-heuristics and self adaptive approaches [9]. The interested reader can find more details in surveys by Schaerf [10], Carter [2], Burke et al. [11], Carter and Laporte[12] and Bardadym[13]. Some recent timetabling research directions are discussed in Burke and Petrovic [14].

In this paper, we focus on the adaptation of a search technique which can search efficiently over a very large set of 'adjacent' (neighbourhood) solutions. This search methodology, originally described by Ahuja and Orlin [18], has been applied successfully to a number of difficult combinatorial optimization problems. It is based on constructing an appropriate improvement graph and identifying improvement moves by solving negative cost subset-disjoint graph cycles or path problems using a label-correcting algorithm. This paper describes the first attempt to adapting Ahuja-Orlin's search ideas to the examination timetabling problem.

2. Problem Description

This problem description is adapted from [16]. The input for the examination timetabling problem can be given as follows:

- N is a number of exams.
- P is a given number of avaliable timeslots.
- M is a number of students
- $C = (c_{ij})_{NxN}$ is a conflict matrix where each element denoted by $\boldsymbol{c_{ij}}$, $i,j \in \{1,\ldots,N\}$ is the number of students taking exams i and j.

The objective function is to minimize the total penalty, F_c formulated aas follows:

$$F_c = \frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} c_{ij} \cdot proximity(t_i, t_j)}{M} \quad \text{(Eq.1)}$$

where $proximity(t_i, t_j) = \begin{cases} 2^5 / 2^{|t_i - t_j|} & if \quad 1 \leq |t_i - t_j| \leq 5 \\ 0 & otherwise \end{cases}$ (Eq. 2)

subject to:

$$\sum_{i-1}^{N-1} \sum_{j=i+1}^{N} c_{ij} \cdot x(t_i, t_j) = 0 \quad \text{(Eq. 3)}$$

where $x(t_i, t_j) = \begin{cases} 1 & if \quad t_i = t_j \\ 0 & otherwise \end{cases}$

where t_k ($1 \leq t_k < P$) which specifies the assigned timeslot for exam k ($k \in \{1,\ldots,N\}$).

Eq. 2 represents a proximity value between two exams that was suggested by Carter *et al.* [17]. If a student has two consecutive exams then a proximity value of 16 is assigned. If a student has two exams with a free timeslot in between then a value of 8 is assigned and so on. These values are summed up and divided by the number of students, M, to give an average penalty per student. Eq. 3 represents a clash-free requirement such that no student is asked to sit in two exams at the same time. The clash-free requirement is considered as a hard constraint.

3. Solution Approach

To implement Ahuja-Orlin's search procedure for large neighbourhood [18], we addressed the examination timetabling problem as a variant of a partitioning problem. Given the input for the initial feasible assignment of exams schedule as follows:

- D is the number of days
- T is the maximum number of timeslots used in one day

We construct a tree with D chains in addition to a root node (dummy node). We represent the subset of exams to a timeslot x ($x \in \{1,\ldots,T\}$) in day y $\{y \in \{1,\ldots,D\}$) as a cliques (subsets). S_r ($r \in \{1,\ldots,W\}$ where W is a maximum number of subsets) represents a subset of exams that are scheduled in day y timeslot x. Figure 1 shows the tree construction for examination timetabling with just one single large room.

For any hard combinatorial optimisation problem, the notion of a neighbourhood structure is of crucial importance when conducting a heuristic search [18]. We propose a neighbourhood structure and methods to search over this structure by drawing upon the ideas from [18]. The neighbourhood structure is created through the cyclic exchange operation of exams in the corresponding graph. We present a cyclic exchange as $i_1 \rightarrow i_2 \rightarrow \ldots \rightarrow i_k \rightarrow i_1$, where i_1, i_2 ,…, i_k are exams which belong to different subsets. This means that exam i_1 moves from subset S_1 to the subset S_2, exam i_2 moves from subset S_2 to subset S_3 and so on, and finally exam i_k moves from subset S_k to subset S_1, thus completing a cycle of changes. A cyclic exchange is feasible if each of the subsets satisfies problem-specific feasibility (there is no student taking more than one exam in the same subset). Path exchanges are defined similarly to the cyclic exchange but without exchanging back to the original subset. The neighbourhood generated by this move is extremely large.

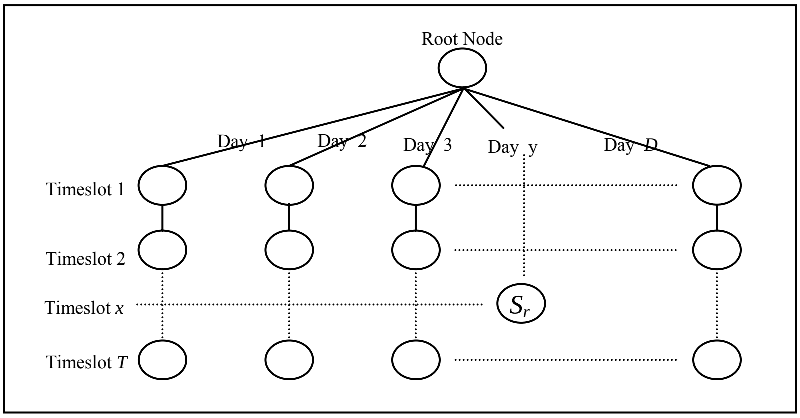

Figure 1: Tree Representation of Exam Timetabling Problem

Once the neighbourhood structure is created, we proceed by defining an improvement graph [15,18]. We then resort to a network flow optimization technique, called the shortest path label-correcting algorithm [19] to find improving moves by finding a negative cost subset-disjoint cycle or path (referred to as a valid cycle or path) for the improvement graph. The pseudo-code for our algorithm for the examination timetabling problem is shown in Figure 2.

```
Obtain a feasible initial solution, InitSol;
Create subsets from InitSol;
Construct the improvement graph, G;
do while (not termination-criteria)
      Find a negative cost subset-disjoint cycles or
       path for  G;
      Update the solution, SolUpdate;
      Create subsets from SolUpdate;
      Updating the G;
end do
```

Figure 2: Pseudo-code for the examination timetabling problem

The algorithm starts with a feasible initial solution of the examination timetabling problem. Then the subsets are created based on days and timeslots. We construct the improvement graph once. In each iteration in the *do-while* loop, we implement the modified shortest-path label-correcting algorithm to find the negative cost subset-disjoint cycles or path for the improvement graph. The modified shortest-path label-correcting algorithm is run several times with a different source exam (origin node), since the success in finding the valid cycles or path is related to the exam from which the search is initiated.

In order to benchmark the performance of this neighbourhood structure, we have conducted computational experiments using instances from Carter's collection [15]. The experiments indicate that our approach produces better timetabling solutions to the current best published results on some of these problems. Figure 3 shows the behavior of the algorithm when applied to one of the instances, *ute-s-92*. The previous best published result for *ute-s-92* was 24.4 by Caramia et al. [20]. Our best result for *ute-s-92* is 24.21. Note that our solution reduced the cost function by 33.91% (with respect to the initial solution), and 0.78% (with respect to the result presented in [20]).

The search algorithm applied to this graph structure of timetabling solutions is able to find a valid improvement cycle or a path, and may consist of more than one move in a single iteration. Thus, this solution scheme helps in reducing the penalty cost when compared to a single move, which is normally used in most of the approaches applied in a timetabling setting. This demonstrates the superiority of this algorithm. The result also indicates that the algorithm can escape from local optima by accepting local worse moves.

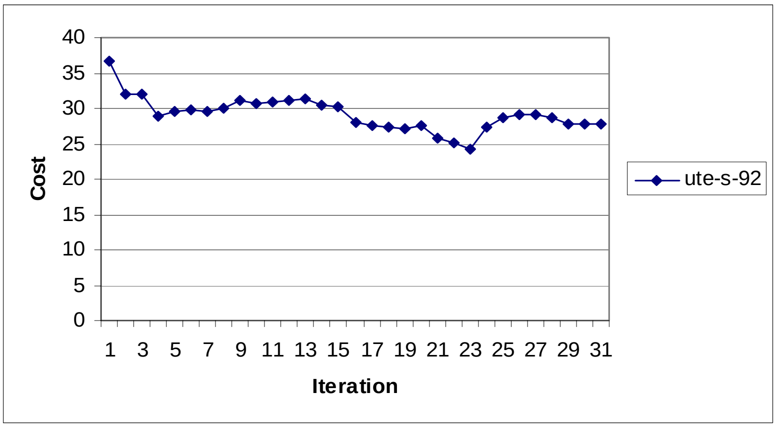

Figure 3 : A behavior of Ahuja-Orlin's algorithm on *ute-s-92*

4. Conclusion and Future Work

Preliminary results indicate that the algorithm produces better solutions on some benchmark problems when compared to other approaches from the literature. The merit of our approach is the combination of a very large neighbourhood structure and the technique of identifying improvement moves in the improvement graph. However, the limitation of our system is the expensive computational time required in the graph updating and negative cycle identification. Our future research aims to shorten the time taken in constructing the improvement graph and the search strategy.

Acknowledgements

This work has been supported by the Public Services Department of Malaysia (JPA) and the University Kebangsaan Malaysia (UKM). Prof. Dror's contribution was funded by an EPSRC Visiting Fellowship (GR/S071241/01).

Reference

[1] Wren, A., *Scheduling, Timetabling and Rostering – A Special Relationship? (1995).*In Edmund Burke and Peter Ross, editors, The Practice and Theory of Automated Timetabling. Lecture Notes in Computer Science 1153, 46-75. Springer-Verlag, Berlin.

[2] Carter, M.W., (1986). *A Survey of Practical Applications of Examination Timetabling*. Operations Research 34, 193-202.

[3] D. de Werra, (1985). *An Introduction to Timetabling*. European Journal of Operations Research, 19, 151-162.

[4] Balakrishnan, N., Lucena, A., Wong, R.T, (1992). *Scheduling Examinations to Reduce Second Order Conflicts*. Computers and Operation Research 19, 353-361.

[5] David, P., (1997). *A Constraint-Based Approach for Examination Timetabling Using Local Repair Techniques*. In Edmund Burke and Micheal Carter, editors, The Practice and Theory of Automated Timetabling. Lecture Notes in Computer Science , Vol. 1408. Berlin, Germany, Springer-Verlag.

[6] J.M. Thompson and K.A. Downsland, (1998). *A Robust Simulated Annealing Based Examination Timetabling System*. Operation Research, 25, 637-648.

[7] Luca Di Gaspero, and Andrea Schaerf, (2001). *Tabu Search Techniques for Examination Timetabling*. In Edmund Burke and Wilhelm Erbens, editors, The Practice and Theory of Automated Timetabling. Lecture Notes in Computer Science 2079, 104-117. Springer-Verlag, Berlin.

[8] D. Corne, H.L. Fang, and C. Mellish, (1993). *Solving the Modular Exam Scheduling Problem with Genetic Algorithms*. In Proceedings of the Sixth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, 370-373.

[9] S. Petrovic and E. Burke., (2004). *University Timetabling*, Ch. 45 in the Handbook of Scheduling: Algorithms, Models, and Performance Analysis (eds. J. Leung), Chapman-Hall/CRC Press.

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

[11] E.K. Burke, K.S. Jackson, J.H. Kingston and R.F. Weare, (1997). *Automated Timetabling: The State of the Art*, The Computer Journal, Vol. 40, No. 9, pp 565-571.

[12] Carter, M.W., and G. Laporte, (1996). *Recent Developments in Practical Examination Timetabling*. In Edmund Burke and Peter Ross, editors, The Practice and Theory of Automated Timetabling. Lecture Notes in Computer Science 1153, 3-21. Springer-Verlag, Berlin.

[13] Bardadym, V.A., (1996). Computer-Aided School and University Timetabling: The New Wave. In Edmund Burke and Peter Ross, editors, The Practice and Theory of Automated Timetabling. Lecture Notes in Computer Science 1153, 22-45. Springer-Verlag, Berlin.

[14] Burke, E.K., Petrovic,S., (2002). *Recent Research Directions in Automated Timetabling*. European Journal of Operational Research (EJOR), Vol. 140, No. 2, pp. 266-280.

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

[16] Burke, E.K., Bykov,Y., Newall, J., Petrovic, S., (2004). *A Time-Predefined Local Search Approach to Exam Timetabling Problem*. Accepted for Publication in IIE Transaction on Operations Engineering.

[17] Carter, M.W., G. Laporte, and Lee, S.Y., (1996). *Examination Timetabling: Algorithmic Strategies and Applications*. Journal of the Operation Research Society 74, 373-383.

[18] Ahuja RK, Orlin JB and Sharma D (2001) Multiexchange Neighbourhood Search Algorithm for Capacitated Minimum Spanning Tree Problem. Mathematical Programming 91, 71-97.

[19] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin, (1993). *Network Flows: Theory, Algorithms, and Applications*. ISBN 1000499012, Prentice Hall, New Jersey.
[20] Caramia, M., Dell'Olmo, P., Italiano, G.F., (2001). *New Algorithms for Examinations Timetabling*. In Naher, S., Wagner, D., editors, Algorithm Engineering 4th Int. Workshop, Proc. WAE 2000 (Saarbrucken, Germany, September. Lecture Notes in Computer Science, Vol. 1982, 230-241. Springer-Verlag, Berlin Heidelberg New York.