

# Construction of Examination Timetables Based on Ordering Heuristics

Syariza A. Rahman, Andrzej Bargiela, Edmund K. Burke, \*Barry McCollum and Ender Özcan



School of Computer Science, The University of Nottingham, Nottingham, UK

\*University of Queensland, School of Electronics, Electrical and Computer Engineering, St. Leonards, Queensland, Australia  
 School of Electronic and Electrical Engineering, Northern Ireland, UK

Examination timetabling is an NP hard real world problem. The complexity of the problem arises due to several reasons e.g. the introduction of flexible course structures, increasing student enrolments etc... Further research is required to enhance the quality of the timetable in such a manner so as to satisfy both the institutional and personal preferences.

A number of research papers formulate examination timetabling as a graph coloring problem [2] where the vertices represent the examination, the edges represent the conflict (students taking both corresponding examinations at one time) occurring between two examinations and the colors for the vertices represent the time slots for the examination (Figure 1).

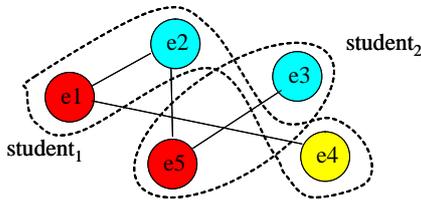


Figure 1: A graph model for a simple examination timetabling problem

## Adaptive Heuristics Ordering Based On Priorities

Based on the concept of squeaky wheel optimization [3]. It is an iterative greedy approach that cycles around three successive processes:

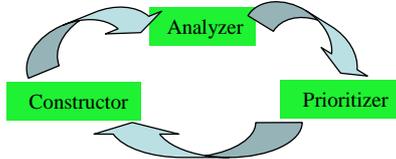


Figure 2: The concept of squeaky wheel optimisation

This study extends the previous work provided in [1]. We used the idea of difficulty and heuristic modifier within the Analyzer.

$$difficulty_i(t) = heuristic_i(t) + heurmod_i(t)$$

Different heuristic modifiers are used in order to stress the priority to the difficult examinations:

- Custom (C)

$$heurmod_i(t) = heurmod_i(t-1), heurmod_i(0) = 0$$

- Additive (AD)

$$heurmod_i(t) = heurmod_i(t-1) + 1, heurmod_i(0) = 0$$

- Multiplicative (MP)

$$heurmod_i(t) = heurmod_i(t-1) + c, heurmod_i(0) = 0, c = 2$$

- Expo

$$heurmod_i(t) = c \cdot heurmod_i(t-1), heurmod_i(0) = 1, c = 2$$

## Graph Coloring Heuristics

**Largest Degree (LD)** - ordering is based on the largest number of conflicting exams.  $heuristic_i(t)$  holds the number of conflicting exams for exam  $i$ .  $difficulty_i(t)$  will be increased at each of iteration  $t$  if the exams are unscheduled.  $heuristic_i(t)$  remains unchanged and the  $heurmod_i(t)$  will be increased at each the iteration.

**Saturation Degree (SD)** - Ordering is based on the number of time-slots in conflict where the exams with the fewest conflicts will be scheduled first. Initialised the  $difficulty_i(t)$  with 1. This value will keep increasing if exam cannot be scheduled during the iteration.

## Shuffling the Ordering of Exams

**Block size** - Partitioned all ordered exams into fixed size of block and shuffle all exams within each block randomly. Implemented for LD heuristic.

**Top window size** - Choose an exam randomly from a fixed number of top exams. After the selection, the difficulties are updated and no rearrangement takes place. Implemented for SD heuristic.

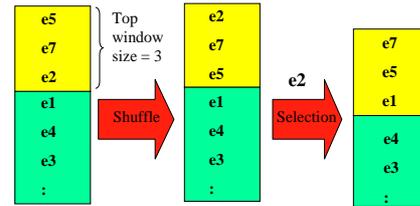


Figure 3: Shuffling the ordering of exams

## Time-Slots Choice

Assignment decision is based on the least penalty obtained considering all time-slots. In the case of equality, a random time-slot is chosen.

## Experimental Results

Different combinations of the algorithmic choices that aim to construct an examination timetable with no conflict and a good spread are investigated. The generic cost function is described in [2]. Toronto benchmark data set is used during the experiments. Each run is repeated 50 times. Block/top window size = {none, 2, 3, 4, 5, 6, 7, 8, 9, 10}.

Problem	Combination of Algorithmic Choices {number of iterations, heuristic type, modifier type, Block/Top Window size}	
	<i>bestLD</i>	<i>bestSD</i>
	car92 I	4.56 {4000, LD, EX, 3}
car91 I	5.36 {4000, LD, EX, 9}	<b>5.08</b> {4000, SD, EX, 5}
ears83 I	40.00 {4000, LD, MP, 3}	<b>38.44</b> {4000, SD, MP, 2}
hec92 I	11.84 {2000, LD, MP, 6}	<b>11.61</b> {2000, SD, C, 5}
kfu93	15.54 {4000, LD, EX, none}	<b>14.67</b> {4000, SD, EX, 2}
lse91	11.78 {4000, LD, EX, 3}	<b>11.69</b> {2000, SD, MP, 6}
rye92	9.69 {4000, LD, EX, 4}	<b>9.49</b> {4000, SD, AD, 5}
sta83 I	157.85 {4000, LD, EX, 9}	<b>157.72</b> {4000, SD, C, none}
tre92	8.88 {4000, LD, EX, 2}	<b>8.78</b> {4000, SD, C, 9}
uta92 I	3.66 {4000, LD, EX, 2}	<b>3.55</b> {4000, SD, EX, 3}
ute92	26.82 {4000, LD, EX, 7}	<b>26.63</b> {2000, SD, EX, 7}
yor83 I	41.59 {4000, LD, EX, 6}	<b>40.45</b> {4000, SD, C, 5}

Table 1: The best combination of algorithmic choices for each problem

## Conclusion

Increasing the difficulty in certain ways can obtain good approximate solutions. As a dynamic graph coloring heuristic, saturation degree has produced most of the best results as compared to the largest degree heuristic. In considering the appropriate heuristic modifier, exponential approach is the best for largest degree while, for saturation degree it has variation in the type of heuristic modifier. The block size and top window approach in this study has varied in certain ways since the incorporation of stochastic element in our approach. Our approach is simple yet effective.

## Reference

- [1] E. K. Burke and J. P. Newall, "Solving examination timetabling problems through adaptation of heuristics orderings," Ann. Oper. Res., vol. 129, pp. 107-134, 2004.
- [2] [17] M. W. Carter, G. Laporte and J. W. Chinneck, "A general examination scheduling system," Interfaces, vol. 24, pp. 109-120, 1994.
- [3] D. E. Joslin and D. P. Clements, "'Squeaky wheel' optimization," J. Artificial Intelligence Res., vol. 10, pp. 353-373, 1999.