Toward Building a Course-Timetabling Decision-Support System

Anthony Wehrer and Jay Yellen

Department of Mathematics and Computer Science, Rollins College
1000 Holt Ave., Winter Park, FL 32789
awehrer@rollins.edu and jyellen@rollins.edu

Abstract

We report on our progress on building a decision support system for course timetabling. Our goal is to construct course timetables that minimize some measure of total con flict and tend toward producing compact schedules for pro fessors and students. Our initial application is for the Sci ence Division courses at Rollins College, but the results should ultimately lead to a more general timetabling system that is adaptable to other institutions. The core of the system is an automated timetable construction, whose strategies and heuristics are based on a weighted graph model that the sec ond author has been developing over the past several years. The other key component is a graphical user interface (GUI) that will allow the user to input problem specific conditions and impose changes, improvements, and/or repairs at vari ous stages of the construction.

Introduction

At many institutions, including Rollins College, course schedules for each term are created manually. This paper describes our progress on the design and implementation of a course-timetabling system for the Science Division at Rollins College, as a first step toward our longer-range goal of a robust decision-support system for university timetabling.

In an actual course-timetabling problem, the undesirability of assigning various pairs of courses to overlapping timeslots varies greatly. If all undesirable conflicts are treated equally, as they are in the traditional vertex-coloring approach to timetabling, then a conflict-free timetable is likely to be impossible. A more realistic objective is to minimize total conflict severity, where conflicts are assigned different levels of severity. A secondary objective and further complication is the desire to create timetables that result in relatively compact schedules for the professors and students.

Copyright © 2010, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Our system is based on a weighted-graph model that the second author has been developing over the past several years. We used the exam-timetabling system previously developed in (Carrington et al. 2007) and (Burke, Pham, et al. 2008) as a starting point.

Description of the System

Weighted Graph Model

Each class to be scheduled is represented by a vertex, and two vertices are joined by an edge if the corresponding classes are to be taught by the same professor, require the same resource, or are expected to be wanted by one or more students. Each edge is assigned a two-component weight. The first component, *conflict severity* (CS), represents the undesirability of assigning the courses to the same or overlapping timeslots, and the second component, *proximity impact* (PI), represents the degree to which assigning the two classes to timeslots far apart will contribute to non-compact schedules. Each timeslot corresponds to a different color, and a timetable corresponds to a vertex-coloring of the weighted graph. Associated with each vertex is a *penalties list* of penalty pairs, one pair for each color. The two components of the penalty pair for a given color correspond, respectively, to the conflict penalty (CP) and proximity penalty (PP) contributions if that color is assigned to the vertex. The penalties list is initialized with either all zero penalty values or can be used to discourage or prohibit certain color assignments before the coloring begins. As the coloring proceeds, the penalties lists of *neighbors* (adjacent vertices) are updated appropriately.

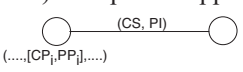

Figure 1. Penalties list and edge weight

Selection Heuristics

Vertex and color-selection strategies are based largely on *heuristic evaluation functions* that are linear combinations

of various *primitive* vertex- and color-selection heuristics, some adapted from the second author's previous work with (Carrington et al. 2007) and (Burke, Pham, et al. 2008), and some new. Such a function, say *f*, is used to identify the most troublesome vertices. The vertex having the largest evaluation is chosen to be the next vertex to be colored. More specifically, for each uncolored vertex *v*,

$$f(v) = a_1 x_1(v) + a_2 x_2(v) + \cdots + a_t x_t(v)$$

where the x_i's are primitive heuristic functions, and the a_i's are nonnegative weights that account for their relative importance and any scaling differences among them. Some of the heuristics that we have been testing in our current implementation are:

- $x_1(v)$ – The number of colors that have no suitable rooms available for *v* or whose corresponding penalty pair in the penalties list of *v* has a component exceeding its respective threshold.
- $x_2(v)$ – The number of edges incident on *v* whose other endpoint is uncolored and who has a weight component greater than its respective threshold.
- $x_4(v)$ and $x_5(v)$ are continuous analogs of $x_1(v)$ and $x_2(v)$, respectively.

As with vertex selection, colors are selected based on a linear combination of primitive heuristics, $y_1 \cdots y_s$. For each color *c*, the y_i's used in this work are as follows:

- $y_1(c, v)$ – The conflict penalty of assigning *c to v*.
- $y_2(c, v)$ – The proximity penalty of assigning *c* to *v*.
- $y_3(c, v)$ – The degree to which the rooms available to *c* are unsuitable for *v*.

GUI

Currently, our GUI is used only for displaying a timetable, but eventually, it will enable the user to influence several key aspects of the construction as it proceeds.

Results Section

The table shown compares our best solution obtained so far with the actual Fall 2009 schedule used at Rollins.

	Our best	Actual
Total Conflict Penalty	118	2066
Avg. Proximity Penalty	14.28	57.91
# of Heavy Conflicts	0	5
# of Medium Conflicts	5	2
# of Light Conflicts	18	26
# of Rooms Needed	0	0

Some Further Research Directions

- Create and evaluate new heuristics as well as those in the current set, especially the continuous analogues.
- Further develop the GUI to include an input component as well as a component for the *repair* and *improvement* of a timetable that can handle number of "what if" scenarios such as when a currently assigned timeslot is no longer suitable for a certain event.
- Add a *backtracking* component to the one-pass construction algorithm. In particular, if during the initial construction, a vertex is selected for which there is no satisfactory color assignment (according to some pre-defined threshold), then one or more vertices are selected for uncoloring to free up a satisfactory color for the given vertex.
- Identify problem characteristics that help determine the most effective weights and thresholds for heuristics. This could lead naturally to a hyper-heuristics approach (see, e.g., Qu and Burke 2009).
- Introduce a learning mechanism and feedback loop that uses characteristics of the solution generated to adjust the threshold values, linear-combination weights, and various other aspects of the underlying selection heuristics.

References

Burke, E.K., McCollum, B., McMullan, P., and Yellen, J., Heuristic Strategies to Modify Existing Timetables, presented at PATAT08, Montreal, Canada, 2008.

Burke, E.K., Pham, N., Qu, R., and Yellen, J., Linear Combinations of Heuristics for Examination Timetabling, submitted to Annals of Operations Research, November 2008.

Carrington, J.R., Pham, N., Qu, R., and Yellen, J., An Enhanced Weighted Graph Model for Examination/Course Timetabling, Proceedings of 26th Workshop of the UK Planning and Scheduling Special Interest Group, Prague, Czech Republic, (2007), 9-15.

Carter, M. W., Laporte, G., and Lee, S., Examination Timetabling: Algorithmic Strategies and Applications, Journal of the Operations Research Society 47 (1996), 373-383.

Qu, R. and E. K. Burke, Hybridisations within a Graph Based Hyper-heuristic Framework for University Timetabling Problems, Journal of Operational Research Society 60, (2009), 1273-1285.

Acknowledgements

The second author thanks Rollins College for its support of this research through a McKean Award.