Course Timetabling Using Graph Coloring and A.I. Techniques

Jordan Rickman and Jay Yellen

Department of Mathematics and Computer Science

Rollins College

1000 Holt Ave.

Winter Park, FL, 32789 USA

jrickman@rollins.edu jyellen@rollins.edu

We present a dual-objective course-timetabling system designed to construct course schedules for the Science Division at Rollins College. Our work builds on the system described in [Wehrer and Yellen (2013)], which models the timetabling problem as a vertex-coloring problem in a weighted graph. The weighted graph model allows the system to incorporate both hard and soft constraints by assigning a 2-component weight to each edge that reflects the undesirability of assigning various pairs of timeslots to its endpoints. An earlier version, using single-component penalty weights, was initially developed in [Kiaer and Yellen (1992)]. The two objectives of our system are: (1) minimize the number and severity of conflicts resulting from the schedule; and (2) create compact schedules for students and faculty. Accordingly, the two edge-weight components are: the conflict penalty, incurred when the endpoints are assigned overlapping timeslots (colors); and the proximity penalty, incurred when the endpoints are assigned timeslots with a large gap between them on the same day. The overall objective is to minimize the total penalty of the completed graph coloring.

Wehrer and Yellen used a one-pass algorithm to color the graph. Heuristics adapted from [Carrington, Pham, et al (2007)] were used to select the most “troublesome” uncolored vertex (a troublesome vertex is one that is likely to create problems if its coloring is deferred), and then to select the best color for that vertex. The process repeats until all vertices are colored. Vertex-selection and color-selection both used linear combinations of a few “primitive” heuristics, as introduced in [Burke, Pham, et al (2008)].

Our system uses a new algorithm that incorporates artificial intelligence techniques via a search-tree representation. The root node is the original uncolored weighted graph, and the nodes at level *k* correspond to the (partial) colorings for which *k* of the vertices have been colored. Each child of a node is the result of assigning a color to an uncolored vertex (see Figure 1). The leaf nodes of the search tree represent complete colorings, and our objective is to find a leaf node of minimum total penalty. Our system performs a best-first search by maintaining a priority queue to determine which nodes to expand. Each node generated is placed in the priority queue according to how “promising” it is.

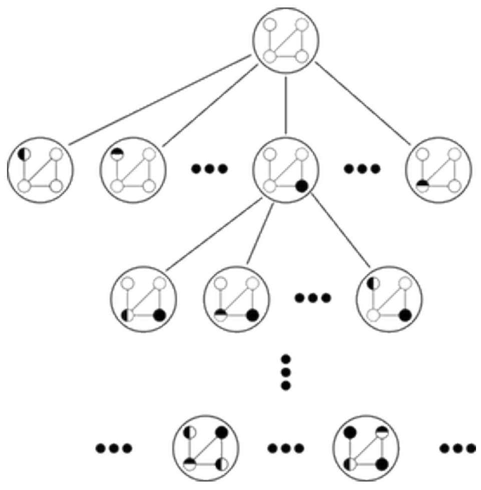
Figure 1: The search tree of 3-colorings of a 4-vertex graph.

Promising nodes are those that appear likely to be the ancestors of low-cost complete colorings. A heuristic evaluation of how promising a node is uses various properties of the partial coloring, including its total penalty and its relation to the rest of the (uncolored) graph. The most promising node is removed from the queue and expanded (i.e., its children are generated) in an iterative process that ends with a complete coloring. The priority queue enables backtracking: if the children of a node have high penalties, they will move towards the back of the queue, and the algorithm will attempt a different branch of the tree. As the number of vertices in the graph increases, the size of the search tree grows exponentially, making a search of the entire solution space intractable. Therefore, we restrict the number of children generated when expanding a node (i.e., the branching factor). We do so by adapting the vertex- and color-selection heuristics to generate a small subset of promising children.

For the purposes of testing, we have developed software that randomly generates course-timetabling problems using a seed problem. The procedure makes random changes to the seed problem, via genetic techniques of mutation and recombination, resulting in a set of problems with similar characteristics. For our seed problem, we use the set of Rollins Science Division courses offered in Fall 2011. We evaluate the effectiveness of various branching and priority-queue strategies by testing them on these randomly generated problems. Currently, we manually adjust various parameters for the branching strategy and the priority queue's heuristic evaluation function, based on their performance on these problems. Our framework will eventually enable us to apply machine learning to adjust these parameters.

Preliminary results on the Fall 2011 problem and 10 randomly generated problems show that even with some simple priority functions and a small branching factor, our search-tree approach produces timetables slightly better to those produced by the old one-pass algorithm. Table 1 compares the performance of the old one-pass algorithm to four different versions of our new algorithm (labeled PQ1, …, PQ4). For the Fall 2011 problem, the table displays the total penalty and its two primary components, conflict penalty and proximity penalty. For the 10 random problems, the values are averaged over the 10 resulting timetables. As we refine our priority function, we expect that our algorithm's performance will continue to improve.

Table 1: Comparison of One-Pass to Priority-Queue Algorithm

Algorithm	Fall 2011			Averages on 10 Random Problems		
	Total Penalty	Conflict Penalty	Proximity Penalty	Total Penalty	Conflict Penalty	Proximity Penalty
One-Pass	7366	83	5291	16364	456	4754
PQ1	7224	100	4724	11601	278	4646
PQ2	7338	85	5213	12594	310	4654
PQ3	7322	99	4847	13974	364	4784
PQ4	7345	86	5195	12149	297	4714

Keywords *Graph Coloring, Heuristics, Timetabling, Weighted Graph, Artificial Intelligence, Genetic Algorithms, Search Algorithms*

References

[Burke, Pham, et al (2008)] Burke, E.K., Pham, N., Qu, R., and Yellen, J., Linear Combinations of Heuristics for Examination Timetabling, Annals of Operations Research, Vol. 194, No. 1 (2012) 89-109.
[Carrington, Pham, et al (2007)] 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 (2007) 9-15.
[Kiaer and Yellen (1992)] Kiaer, L., and Yellen, J., Weighted Graphs and University Timetabling, Computers and Operations Research Vol. 19, No. 1 (1992), 59-67.
[Wehrer and Yellen (2013)] Wehrer, A., and Yellen, J., The Design and Implementation of an Interactive Course-Timetabling System, Annals of Operations Research, May 2013.