10th International Conference of the Practice and Theory of Automated Timetabling
PATAT 2014, 26-29 August 2014, York, United Kingdom

Decomposition and Recomposition Strategies to Solve Timetabling Problems

Dulce J. Magaña-Lozano · Ender Özcan · Santiago E. Conant-Pablos

1 Introduction

Timetabling is a crucial and often extremely time consuming task in many educational institutions. This task is generally performed periodically (each year, semester, quarter) for fulfilling the requirements imposed by the institution and people involved, such as students and teachers/lecturers making efficient and effective use of the available resources. The educational timetabling problem has been widely studied and different classifications have been proposed (see [1], [5]). The course timetabling problem (CTTP) is a combinatorial optimisation problem which involves assignment of a given set of meetings along with available resources to appropriate time slots subject to a set of constraints. In general, two types of constraints can be identified: *hard* and *soft*. The hard constraints are those that *must be* satisfied under any circumstances. Timetables that do not violate hard constraints are called *feasible*. On the other hand, soft constraints are those that *need to be* respected as many as possible, but can still be violated if necessary, i.e. they are desirable but not essential. These constraints are frequently used to evaluate how good the solutions (timetables) are.

Two classes of well known solution methods in timetabling are construction and decomposition methods (e.g. [2], [3], [6]). In this study, we investigate an approach which decomposes a given problem into smaller subsets and then sequentially constructs a partial solution using the subsets recomposing them into a complete solution. The proposed approach is tested on the Post Enrolment based Course Timetabling problem instances of the Track 2 from the second International Timetabling Competition (ITC2007) for solving the hard constraints. The

Dulce J. Magaña-Lozano · Ender Özcan
ASAP Research Group, University of Nottingham, School of Computer Science
Jubilee Campus, Wollaton Road, Nottingham, NG8 1BB, UK
{ psxdjm, ender.ozcan }@nottingham.ac.uk

Santiago E. Conant-Pablos
Tecnológico de Monterrey
Av. E. Garza Sada 2501, Monterrey, NL, 64849, Mexico
sconant@itesm.mx

main characteristic of this track is that the timetable is produced after student enrolment on courses has taken place.

2 Experiments and Results

In this study, we formulate the post enrollment based course timetabling problem as a Constraint Satisfaction Problem (CSP). Firstly, all meetings are ordered using a heuristic. In our experiments, we used *Largest Degree* heuristic, which sorts the meetings in decreasing order by the number of conflicts with other meetings. After decomposing a given set of meetings into fixed size subsets, we construct a partial and independent solution using each subset. At this stage, each subset is also in order. The meetings in the first subset will be the events with the largest degree, and in the last subset, the meetings with the smallest degree. Then the subsets are recomposed sequentially towards a complete solution with a certain strategy embedding a conflict resolution heuristic, since merging partial solutions into a larger partial solution could cause hard constraint violations.

Each subset represents a subproblem which are solved by a backtracking algorithm with forward checking. This algorithm dynamically selects the next variable to assign using the *minimum remaining values* heuristic which uses the variable with the smallest domain. To break ties, the *saturation degree* heuristic is used, which chooses the variable with the maximum number of constraints over the unassigned variables. Moreover, the *least constraining value* heuristic is employed to assign a value, which reduces the size of the domain of unassigned variables at the least, to the chosen variable. Ties are broken randomly.

An *incremental* recomposition strategy, similar to the one proposed in [3], is utilised. Without changing the initial ordering of subsets and using the first subset as the initial partial solution, the next subset is incrementally added into the partial solution in hand until all subsets are covered. Each time a subset is included in the growing partial solution, conflict resolution algorithm is invoked. An important feature of the recomposition strategy is that it should be able to reduce the conflicts between the variables that belong to different subsets. Hence, it embeds the *Min-Conflict* local search algorithm as a conflict resolution method while integrating the subsets. This algorithm produces a list of variables in conflict and randomly chooses one of them to assign a different value which generates the minimum number of conflicts with the other variables, namely, the one that minimizes the number of unsatisfied hard constraints. These steps for conflict resolution are repeated for a fixed number of times (attempts) in order to gradually reduce the number of conflicts between the variables, and eventually find a feasible solution.

It is possible that all the events can not be scheduled in the given time without breaking some hard constraints, thus some events in the timetable will not be placed in order to ensure that no hard constraints are being violated. If there are unplaced events, the *Distance to Feasibility* (DtF) measure is calculated [4] as proposed in ITC2007. This measure represents the total number of students that attend to each of the unplaced events, 0 means that all events were scheduled without violations to the hard constraints. The DtF is used to measure the performance of the algorithms. The Equation 1 shows how to calculate the DtF measure where e_i represents the event i, s_{e_i} the number of students that attends to the event i and, n the number of events.

$$DtF = \sum_{i=0}^{n} s_{e_i}[e_i \text{ is unplaced}] \quad (1)$$

We have fixed the number of subsets for a given problem instance. An initial set of exhaustive parameter tuning experiments were conducted to determine the ideal number of subsets. The setting of $\sqrt{Number\_Events}$ for the number of subsets yielded the best performance over multiple runs across the instances with respect to DtF. We repeated each experiment fixing the number of subsets to this value thirty times for each instance. Table 1 shows the best and the average DtF over all runs, as well as, the associated standard deviation for each instance using the Largest Degree decomposition heuristic and the Incremental recomposition heuristic. In all problem instances, except five of them, namely; comp-2, comp-9, comp-10, comp-21 and comp-22, the proposed approach is capable of obtaining feasible solutions. The approach always achieves a feasible solution across all runs for the following five instances: comp-4, comp-8, comp-11, comp-15 and comp-17.

Table 1 Results of one of the proposed algorithm (Largest Degree heuristic with Incremental Recomposition), where *best*, *avr.* and *st.d.* denotes the best, average and standard deviation of the DtF. For each instance, the bold entry marks the best performing approach; the comparison criteria is the average number of DtF.

Instance	best	avr.	st.d.	Instance	best	avr.	st.d.
comp-1	0.0	701.6	447.1	comp-13	0.0	66.0	69.7
comp-2	467.0	1384.7	401.8	comp-14	0.0	114.4	68.3
comp-3	0.0	5.0	27.6	comp-15	0.0	0.0	0.0
comp-4	0.0	0.0	0.0	comp-16	0.0	2.3	12.6
comp-5	0.0	46.2	37.0	comp-17	0.0	0.0	0.0
comp-6	0.0	10.9	20.2	comp-18	0.0	3.5	19.0
comp-7	0.0	4.3	13.0	comp-19	0.0	799.9	479.8
comp-8	0.0	0.0	0.0	comp-20	0.0	4.9	18.5
comp-9	152.0	1152.3	482.7	comp-21	29.0	448.9	266.0
comp-10	816.0	1473.7	376.1	comp-22	3249.0	4759.6	491.6
comp-11	0.0	0.0	0.0	comp-23	0.0	1117.3	638.2
comp-12	0.0	14.6	44.5	comp-24	0.0	396.4	354.4

References

1. Victor A. Bardadym. Computer-aided school and university timetabling: The new wave. In E. Burke and P. Ross, editors, *Practice and Theory of Automated Timetabling*, volume 1153 of *Lecture Notes in Computer Science*, pages 22–45. Springer Berlin Heidelberg, 1996.
2. M. W. Carter. A decomposition algorithm for practical timetabling problems. Technical paper, Department of Industrial Engineering, University of Toronto, 1983.
3. Ender Özcan, Andrew J. Parkes, and A. Alkan. The interleaved constructive memetic algorithm and its application to timetabling. *Computers and Operations Research*, 39(10):2310–2322, 2012.
4. Ben Paechter. International timetabling competition. `http://www.cs.qub.ac.uk/eventmap/postenrolcourse/course_post_index_files/evaluation.htm`, 2007.
5. A. Schaerf. A survey of automated timetabling. *Artificial Intelligence Review*, 13(2):87–127, 1999.
6. Petr Šlechta. Decomposition and parallelization of multi-resource timetabling problems. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3616 LNCS:177 – 189, 2005.