A Constraint Logic Programming Based Approach to the International Timetabling Competition

Patrick Pleass[1], Mark Wallace[2], Mauro Bampo[3]

[1,2] Faculty of Information Technology
Monash University, Melbourne, Australia
{patrick.pleass, mark.wallace}@infotech.monash.edu.au

[3] Faculty of Engineering
University of Bologna, Bologna, Italy
mbampo@gmail.com

This paper outlines the modeling, implementation and refinement of a solution to the International Timetabling Competition using Constraint Logic Programming methods. This is primarily carried out within the ECLiPSe constraint programming framework using lib(ic), the hybrid integer/real interval arithmetic constraint solver library.

The International Timetabling Competition, organized by the Metaheuristic Network and sponsored by PATAT (Practice and Theory of Automated Timetabling) was held in 2003. The competition presented a reduced university course timetabling problem and associated problem datasets designed by Ben Paechter. The aim of the competition problem was to deliver feasible timetables, in a set execution time that meets all hard constraints and minimized occurrences of soft constraints.

Although this competition has already been held and winners announced [1,2,3,4], the outcome has provided researchers with a number of independently verified solutions and performance measures using a variety of different approaches. Further to this the Center for Emergent Computing at Napier University have posted new "Harder" Instances for the University Course Timetabling Problem [5] that can further challenge heuristic development in this field.

The aim of this research is to provide a solution to the timetabling problem using as much as possible the ECLiPSe framework and minimal use of external custom-built metaheuristics and solvers. The performance of this approach is then compared to the competition results and differences analyzed and discussed. This approach introduces a number of design challenges in providing acceptable performance within ECLiPSe as opposed to a custom built heuristic. These challenges are outlined and discussed.

The approach followed consists of the following stages:

Modeling the problem data

Data from the input file format specified by the competition is loaded into the constraint engine as a set of atomic facts such as:

```
timeslot(timeslot_id)
student(student_id,[classes]),
```

E. K. Burke, H. Rudová (Eds.): PATAT 2006, pp. 464–466. ISBN 80-210-3726-1.

```
room(room_id,capacity,[available_features])
class(class_id,[required_features],TIME,ROOM)
```

The objective is to find TIME and ROOM for each defined class that meets all hard constraints and as many soft constraints as possible in the given time.

Modeling the hard constraints

The hard constraints to be implemented include:

- No student attends more than one event at the same time
- The room allocated to an event is big enough to house all students and meets all feature requirements
- Only one event is in each room for any timeslot

There are multiple ways to model these constraints within ECLiPSe. We present a high performance solution that minimizes the search domain to reduce the total search space for the next stage of the solution. Central to this step is the application of the `alldifferent(1)` predicate to an array of compound variables (ROOM and TIME) unified with the `element(3)` predicate.

Modeling the soft constraints

The soft constraints to be implemented are as follows:

- Minimize students with a class in last slot of each day
- Minimize students with two consecutive classes
- Minimize students with a single class on a single day

There are many strategies that may be employed within ECLiPSe to minimize these values. We employ in the first instance using selective constraint propagation techniques and then extend as performance dictates to other strategies based on constraint based local search.

Non-Exhaustive Search Strategies

As more variables are added to the problem, the search space grows exponentially, and left unchecked a CLP based system will search all possibilities. We devise heuristics that perform effective search strategies that focuses on promising parts of the search tree in order to avoid an exhaustive search. This step will also utilize local and hybrid search and repair, chain swap and large neighborhood search.

Our approach is to find a complete assignment of variables that meet all hard constraints and as many soft constraints as possible in reasonable time. The total time the competition allows for finding a solution is 564 seconds on the hardware we employed. Reasonable time in our case is less than half of this time. The remaining time is used to perform constraint based local search to improve the solution.

Backtrack-Free Constructive Algorithms

To be able to produce solutions in reasonable time we have employed a backtrack-free, forward-checking constructive method as described by Schaerf [6]. This method will move from variable to variable and select the best value that meets the hard constraints and meets as many soft constraints as possible. The domains for each of the unassigned variables are then pruned to ensure that a total assignment of values exists before continuing.

The task is to find values for the time T and room R variables for each class. During the constructive search we focus only on the Time variable as most of the soft constraints rely on this. However with the assignment of every time variable, we also combine a channeling constraint that ensures that for any Time selection for a class there is at least one suitable room available that meets all hard constraints. The final allocation of rooms happens after the time allocation is complete.

References

1. Kostuch, P: The University Course Timetabling Problem with a 3-phase approach. PATAT 2004.
2. Jaumard, B., Cordeau, J., Morales, R.: Efficient Timetabling Solution with Tabu Search. International Timetable Competition, 2003.
3. Bykov, Y.: The Description of the Algorithm for International Timetabling Competition. International Timetable Competition, 2003.
4. Di Gaspero, L., Schaerf, A.: Timetabling Competition TTComp 2002: Solver Description. International Timetable Competition, 2003.
5. Lewis, R. and Paechter, B: New "Harder" Instances for the University Course Timetabling Problem http://www.emergentcomputing.org/timetabling/harderinstances.htm
6. Scharef, Andrea: Combining Local Search and Look-Ahead for Scheduling Constraint Satisfaction Problems. Proc. of the 15th International Joint Conference on Artificial Intelligence (IJCAI-97), 1997.
7. Liatsos, V.: An Environment for a Resource Allocation Problem in CLP, MSc Thesis, University of London, 1995