A Four Stages Heuristic for Curriculum-based University Course Timetabling Problem

Mauritsius Tuga[1] and Regina Berretta[1]

School of Electrical Engineering and Computer Science
The University of Newcastle, NSW 2308, Australia
morits173@yahoo.com,Regina.Berretta@newcastle.edu.au

Abstract.
Problem Identification
By considering the lectures of a course as event and the cartesian product between the set of periods of a week and the set of rooms as the set of resources, the curriculum based course timetabling problem can be viewed as the assignment of events to resources, taking into consideration all constraints posed to them. Some related data structures such as matrices were utilised to cope with the hard constraints.
The timetabling process is carried out in four stages. The first two stages deal with the feasibility poblems, i.e. the problems of finding a timetable with no hard constraints violations, and the last two stages deal with the optimisation problems, i.e; the problem of minimising the soft constraints violations.
Stage 1. Constructive Heuristic
A Constructive Heuristic (CH) is used to try to find a feasible solution. There are cases that our chosen Constructive Heuristic failed to find a feasible solution for some instances. We therefore utilise the CH in the following way. For each instance, the CH tries to generate an initial solution with an increasing number of periods starting with the given periods + 1, until a "feasible" solution is found. A "feasible" solution in this case is a solution with no hard constraint violations, possibly using some extra periods. This solution is then passed to Stage 2.
Stage 2. Simulated Annealing (SA) for Feasible Solutions
We define the number of unscheduled events + the number of unscheduled students as the objective function in our SA schema. The process is terminated if a global optimum solution is found or if the time limit allocated to this stage is reached. The best solution found (complete or uncomplete) in this process is then passed to Stage 3.
Stage 3. Simulated Annealing for Improving the Solutions
This stage is aimed to minimise the soft constraints violations. The SA in this stage only deals with those events that have been put in the timetable. The process is terminated if a global optimum solution is found or if the time limit for Stage 3 is reached.
Stage 4. Kempe Chain-based Hill Climbing
In this stage, an effort to improve the best solution found in Stage 3 is carried out. A Kempe-chain neighbourhood structure is used, and the heuristic only accept an improving solution. The process is stopped if the total time limit granted to the whole stages is reached.

Computational Experiments and Results

We tested our approach using a PC pentium 4 - 3.0 GHz and 256 MB of RAM running under ms-windows xp. We tested 20 to 60 runs for each instance. Each run takes at most 429000 seconds. This time limit is obtained after benchmarking the machine using the program provided by the organiser of the competition. The solutions can be found in the .sln files and the quality of the solutions can be found in the file CurrResultMT.txt.