Description of Algorithm

Abstract. The algorithm described below was developed in order to satisfy the author's curiosity how well a rather simple algorithm compares to more sophisticated methods that are based on longterm research in the field of timetabling. The author's goal was to keep the programming effort relatively low and to see whether the algorithm would return a respectable solution anyway.

The algorithm consists of two parts. It starts with a heuristic that finds a solution in a greedy fashion, followed by a simple local search procedure that uses the remaining time to improve upon the solution found so far. During the heuristic, one lecture after the other is assigned to a room-time slot, thereby giving priorities to lectures with lower schedulability, that is lectures, for which the number of appropriate room-time slots is lowest. A room-time slot is appropriate for a lecture (and vice-versa), if it is unoccupied, no lecture conflict is induced, no unavailability constraint is violated, and the room fits almost all of the students. Among all appropriate room-time slots for a lecture, we give preference to the ones with low schedulability and unwasted capacity. That is, we prefer a room-time slot for which the number of appropriate lectures is low, based on the time table so far, and which is at the same time not too big for the class size. Interestingly, the heuristic returns for each non-hidden dataset a time table that does not violate any of the hard constraints. All soft constraints, except for the room capacity constraint are disregarded entirely during the heuristic procedure. For the local search procedure, the neighborhood is defined by changing the room-time slot for one lecture to a different one. However, occupied or unavailable room-time slots are excluded. If the penalty costs for the new solution are lower, the new solution is accepted. If no improvement for a lecture can be found, we allow a deteriorating step with a certain probablity which will increase linearly with time, and which also depends on the degree of deterioration ("simulated annealing"). The start temperature of each instance is computed based on its input parameters (such as the number of curricula, lectures, rooms, times, unavailability constraints, lectures in conflict), and parameters calculated during the greedy phase, such as the sum of the schedulability of the lectures.