International Timetabling Competition 2007
Track 3 - Curriculum Based Course Timetabling
Solver description

Technological Educational Institute of Thessaloniki –
Department of Informatics.
Students: Karapetsas Eleftherios and Georgiadis Kostas

The algorithm used for the solution of the problem is a genetic algorithm. The JGAP Genetic Algorithm package for Java was used with some changes by us to suit the problem. The chromosome representation is simple. A chromosome's size is the number of all the lectures of the dataset solved, multiplied by 2. Each pair of 2 genes represent a lecture. The first gene's value being a room and the second gene's a timeslot. By timeslot meaning a variable which describes which period of which day the lecture is being programmed on. So this way we have an accurate representation of a possible solution.

The population's size is determined by the number of courses in the dataset. It is initialized in a way to contain all the legal values equal times. The reason for that is to have as varied genetic material at the start as possible. The fitness function is checking for the hard constraints violations first. The more hard constrains violations there are the bigger the fitness value returned is. Only when there are no hard constraints violations the algorithm starts checking the chromosome for the number of soft constraints it violates making of course sure that the hard constraints stay at zero. It then returns a number (always smaller than the smaller hard constraints violation) stating the score of the potential chromosome solution.

Each generation crossover and mutation is applied on the population of the chromosomes. They are added on the potential chromosomes for the next generation pool. With the passing of generations if a certain fitness value is being repeated a lot then the mutation rate is increased in an effort to get out of the local minimum the algorithm got stuck in. From there a percentage of the fittest chromosomes are selected via the use of a natural selector to go to the next generation. The natural selector used is a tournament selector. It picks some chromosomes from the population and enters them in a tournament. From there it selects the fittest one with a certain percentage. That is to allow varied genetic material to exist while the algorithm runs

The algorithm ends when the time given by the competition rules is elapsed and the fittest chromosome of the final generation is returned as the solution the algorithm has reached up to that point.