A Study on the University Courses Timetabling Problem

K. S. Sung* · Enzhe Yu**

* Dept. of Industrial Engineering
** Kangnung National University, Korea

Abstract

Timetabling is the allocation, subject to constraints, of given resources to objects being placed in space-time, in such a way as to satisfy as nearly as possible a set of desirable objectives. Timetabling is a purely constraint satisfaction problem(CSP), and a NP-hard problem as well. The elements of a typical timetable are: events, time slots, places, lecturers and classes. These elements are arranged in space-time subject to constraints. There are two types of constraints in a timetabling problem. They are hard constraints and soft constraints. Hard constraints are those that can never be violated and must be satisfied in a feasible timetable, while soft constraints do not deal with item conflicts and have a cost associated with them.

There are mainly four approaches toward timetabling problem(TTP) : Cluster Methods, Sequential Methods, Generalized Search Strategies and Constraint Based Approaches. Among them, Genetic Algorithm(GA) which is a Generalized Search Strategy, can solve problems effectively and now causes more and more concerns.

Genetic Algorithms are search algorithms based on the mechanics of natural selection and natural genetics, and they are problem specific. Different from conventional search techniques, Genetic Algorithms starts with an initial set of random solutions called *population*. Each individual in the population is called a *chromosome*, representing a solution to the problems at hand. The chromosomes *evolve* through successive iterations called *generations*. During each generation, the chromosomes are evaluated using some measures of *fitness*. To create the best generation, new chromosomes, called offspring, are formed by either (a) merging two chromosomes from current

generation using a crossover operator, or, (b) modifying a chromosome using a mutation operator. A new generation is formed by selection. After several generations, the algorithms converge to the best chromosome, which hopefully represents the optimum or suboptimal solution to the problem.

In our Genetic Algorithm approach, we designed special chromosomes and GA operators for this special case, and developed timetables within the feasible space. *Chromosomes* are designed to be in two dimensioned matrix form, column and row, represent time slots and places, respectively. The places are firstly sorted by types, i. e., lecture room, labs, etc., and then, places within each types are sorted according to their sizes, thus, the sectors by types and sizes are made. Genes(events) within a chromosome, like the natural ones, contains as much information as possible : subject name, lecturer name, class name, host department, class size, and subject type. The sorted places, which are done according to types and sizes, and the complete information the events contain makes it easy for the later developing job.

During the initialization procedure, we randomly assign the events to the proper area of their kinds and sizes, from common to special, within the chromosome. To avoid hard constraint violations, we also applied hard-constraint-check algorithm in this process. When there occurs a hard constraint violation during the checking, rather than repairing the chromosome, we generate random numbers again for locating the events, for we found that generating random numbers can save much time than repairing, which results in the same effect. These measures effectively help make feasible chromosomes, and reduced unnecessary time on repairing illegal ones to be feasible.

The Evaluation procedure deals with soft constraints and gives penalty values over the unsatisfied ones. The goal is to minimize the cost associated with each kind of soft constraints and drive the evaluation values toward 1 as close as possible.

The operators -- Sector Based PMX(Partially Mapped Crossover) and Sector Based Mutation are both problem specific. In Chromosome representation, the dimension regarding the places are sorted by both types and sizes within each types. The PMX guarantees the crossovers take place among same types and within proper sizes. So is the same with the sector based mutations. Rather than exchanges among randomly selected genes within the whole scale which obvious raises the possibility of making infeasible chromosomes, we design them to mutate within same type and proper sizes. This avoids the confusion of the crossover and mutation among different types of classes and reduces the possibility of making illegal chromosomes. However, the Crossover and Mutation may easily make infeasible chromosomes. Therefore, the repair algorithms aimed at satisfying hard constraints are also developed.

Among the Hard constrains considered in our paper, most of them can be satisfied due to the specific chromosome design. While those two regarding (1)"Neither a class nor a teacher, nor a room is assigned to more than one lesson in the same period" and (2)"All allocated rooms are large enough" are constantly violated after the operations.

As for the first one, the hard-constraint-check procedure is to check vertically within each time slot whether there are same lecturers or classes. If the violations are checked, the conflicted events should be moved to other slots except that at most one is left in the original time slot. The movement of events should also guarantee the feasibility of the chromosome, that is, moving by checking. This job is done within each time slot toward all the events.

As for the second hard constraint mentioned, the checking procedure is simply the comparison between the sizes of events and those of places. The sizes of places should always be no smaller than the events' sizes. If size violations are checked, the repairing job is to move conflicted events in the horizontal direction, also, by checking and avoiding hard constraint violations. The difficulties in the whole checking procedure is the coordination among all the hard constraint violations.

The real university courses timetabling data from Kangnung National University is sued for the test. Most of the calculation time is spent on the search, check and repair of the hard constraints. So the future work aimed at the effectiveness of these problems seem necessary. Due to the results of our study, we draw the conclusion that it is possible for the algorithm to be applied to larger scale data with more complex cases such as more constraints and more kinds of places and types of subjects.