Solving the University Timetabling Problem with Optimized Enrollment of Students by a Self-adaptive Genetic Algorithm

Radomír Perzina

Department of Mathematical Methods in Economics,
School of Business Administration, Silesian University,
University Sq. 1934/3, 733 40 Karviná, Czech Republic
perzina@opf.slu.cz

Abstract. The timetabling problem is well known to be an NP-complete combinatorial problem. The problem becomes even more complex when addressed to individual timetables of students. The core of dealing with the problem in this application is a timetable builder based on mixed direct–indirect encoding evolved by a genetic algorithm with a self-adaptation paradigm, where the parameters of the genetic algorithm are optimized during the same evolution cycle as the problem itself. The aim of this paper is to present an encoding for self-adaptation of genetic algorithms that is suitable for timetabling problems. Compared to previous approaches we designed the encoding for self-adaptation of not only one parameter or several ones but for all possible parameters of genetic algorithms at the same time. The proposed self-adaptive genetic algorithm is then applied for solving the real university timetabling problem and compared with a standard genetic algorithm. The main advantage of this approach is that it makes it possible to solve a wide range of timetabling and scheduling problems without setting parameters for each kind of problem in advance. Unlike common timetabling problems, the algorithm was applied to the problem in which each student has an individual timetable, so we also present and discuss the algorithm for optimized enrollment of students that minimizes the number of clashing constraints for students.

1 Introduction

Genetic algorithms are search algorithms based on the idea of natural selection and natural genetics. It is well known that the efficiency of genetic algorithms strongly depends on their parameters. These parameters are usually set up according to vaguely formulated recommendations of experts or by the so-called two-level genetic algorithm, where at the first level the genetic algorithm optimizes parameters of the second level. Self-adaptation seems to be a promising feature of genetic algorithms, whereby the parameters of the genetic algorithm are optimized during the same evolution cycle as the problem itself. The aim of this paper is to present an encoding and genetic operators for self-adaptation of

E.K. Burke and H. Rudová (Eds.): PATAT 2006, LNCS 3867, pp. 248–263, 2007.

genetic algorithms that is suitable for solving the university timetabling problem. Compared to previous approaches (e.g. [5, 17, 25]) we designed the encoding for self-adaptation for not only one parameter but for all or nearly all possible parameters of genetic algorithms at the same time. Moreover, the parameters are encoded separately for each element of a chromosome.

The proposed self-adaptive genetic algorithm is then applied for solving the real university timetabling problem at Silesian University. The problem is known to be NP-complete and hence there is no known algorithm for solving it in polynomial time [12]. The requirements for timetabling differs from university to university, but in general the timetabling problem consists of assigning each lecture from a set of lectures to a suitable room and a time slot subject to a number of hard and soft constraints, such as no teacher can teach more lectures at the same time, at no room can be taught more than one lecture at the same time, teachers time and room preferences, etc.

At some universities, including Silesian University, each student has an individual timetable, i.e. there are no groups of students which have the same timetable, and it is even difficult to find two students with the same timetable; thus solving the problem becomes very complex. In order to be able to deal with individual timetables of students we designed an algorithm for optimization of enrollment of students that effectively decreases the number of constraints for student clashes.

A large number of diverse methods have been already proposed in the literature for solving timetabling problems. These methods come from a number of scientific disciplines like Operations Research, Artificial Intelligence, and Computational Intelligence and can be divided into four categories:

1. Sequential methods that treat timetabling problems as graph problems. Generally, they order the events using domain-specific heuristics and then assign the events sequentially into valid time–room slots [24].
2. Cluster methods, in which the problem is divided into a number of event sets. Each set is defined so that it satisfies all hard constraints. Then, the sets are assigned to real time–room slots to satisfy the soft constraints too [28].
3. Constraint-based methods, according to which a timetabling problem is modeled as a set of variables (events) to which values (resources such as teachers and rooms) have to be assigned in order to satisfy a number of constraints [9, 16].
4. Meta-heuristic methods, such as genetic algorithms, simulated annealing, tabu search, and other heuristic approaches, that are mostly inspired from nature, and apply nature-like processes to solutions or populations of solutions, in order to evolve them towards optimality [2, 11, 19, 23, 26, 27].

When applying genetic algorithms to some optimization or scheduling problem, the crucial element is encoding. For a timetabling problem there are two main types of encoding: direct [23] and indirect [19]. The advantage of direct encoding is that the whole search space can be encoded, but it usually leads to violation of many hard constraints. Indirect encoding is based on some rules or instructions for building the resulting timetable and so there is less probability