Very Large-Scale Neighborhood Search Techniques in Timetabling Problems

Carol Meyers[1] and James B. Orlin[2]

[1] Operations Research Center, Massachusetts Institute of Technology,
77 Massachusetts Avenue, E40-130, Cambridge, MA 02139, USA
[2] Sloan School of Management, Massachusetts Institute of Technology,
77 Massachusetts Avenue, E53-363, Cambridge, MA 02139, USA
{carol,jorlin}@mit.edu

Abstract. We describe the use of very large-scale neighborhood search (VLSN) techniques in examination timetabling problems. We detail three applications of VLSN algorithms that illustrate the versatility and potential of such algorithms in timetabling. The first of these uses *cyclic exchange neighborhoods*, in which an ordered subset of exams in disjoint time slots are swapped cyclically such that each exam moves to the time slot of the exam following it in the order. The neighborhood of all such cyclic exchanges may be searched effectively for an improving set of moves, making this technique computationally reasonable in practice. We next describe the idea of *optimized crossover* in genetic algorithms, where the parent solutions used in the genetic algorithm perform an optimization routine to produce the 'most fit' of their children under the crossover operation. This technique can be viewed as a form of multivariate large-scale neighborhood search, and it has been applied successfully in several areas outside timetabling. The final topic we discuss is *functional annealing*, which gives a method of incorporating neighborhood search techniques into simulated annealing algorithms. Under this technique, the objective function is perturbed slightly to avoid stopping at local optima, while neighborhood search techniques help provide an effective search of the feasible space.

1 Introduction

1.1 Timetabling Problems

The scheduling of classes and examinations is a key practical problem that is faced by nearly all schools and universities. Substantial effort has been devoted to developing effective timetabling procedures over the last thirty to forty years. The problems tackled by such procedures include *examination timetabling*, in which a set of exams is to be scheduled over a set of time periods, and *course timetabling*, where a set of courses must be scheduled over the length of an entire semester.

Timetabling problems are often complicated by numerous constraints; for instance, in the examination timetabling problem, students should not be scheduled to take two exams at the same time. These constraints are typically divided

E.K. Burke and H. Rudová (Eds.): PATAT 2006, LNCS 3867, pp. 24–39, 2007.

into *hard constraints*, which must not be violated (in the course timetabling problem, a hard constraint might be that no teacher is scheduled to teach two classes at once), and *soft constraints*, which possess a penalty for being violated (in the examination timetabling problem, a soft constraint might be to minimize the number of students who take two exams back-to-back). Because of the number and variety of constraints, such timetabling problems typically constitute NP-hard problems that are quite difficult to solve manually. This in turn has led to an increased emphasis on finding effective *automated* timetabling algorithms.

Recent surveys on automated timetabling (see [22,24,25,44]) illustrate the wide array of methods that have been applied to timetabling problems. Traditional techniques tested in timetabling include *direct heuristics* [35], which fill up the timetable one event at a time and resolve conflicts by swapping exams, and a reduction to the *graph coloring* problem [39], where events are associated with vertices of a graph and edges with potential conflicts. More modern heuristics include *memetic* [21] and *genetic algorithms* [20,28,31], which use techniques inspired by evolutionary biology; *simulated annealing* algorithms [19,48], where nonimproving solutions are permitted with progressively decreasing probability; *tabu search* heuristics [2,27,43], where a list of recently visited timetables are forbidden to be visited; and *constraint logic programming* approaches [26], which are based on applying declarative logic programming systems to constraint satisfaction problems.

In this paper, we address the application of very large-scale neighborhood search techniques (see Section 1.2) to timetable scheduling problems, including one approach based on genetic algorithms (Section 3) and one that resembles simulated annealing (Section 4). Neighborhood search has long been used in timetable scheduling, from the swap (2-opt) techniques used in the direct approaches to the variety of forms of neighborhood search used in genetic algorithms. However, the area of *very large-scale neighborhood search* has only recently been investigated with respect to timetable scheduling [1,14,34] (see Section 2). We believe there are many untapped possibilities for useful algorithms in this context.

1.2 Very Large-Scale Neighborhood Search

Neighborhood search algorithms (also known as *local search* algorithms) are a class of algorithms that start with a feasible solution and attempt to find an improving solution in the *neighborhood* of the current solution. The neighborhood structure may be defined in a variety of ways, typically so that all solutions in the neighborhood of the current solution satisfy a set of prescribed criteria. In *very large* neighborhoods, the size of the neighborhood under consideration is extremely large (typically, exponential) in the size of the problem data, making it impractical to search such neighborhoods explicitly.

A *very large-scale neighborhood search* (VLSN) algorithm is one that searches over a very large neighborhood, giving an improving solution in a *relatively*