The Phase-Transition Niche for Evolutionary Algorithms in Timetabling

Peter Ross[1], David Corne[2] and Hugo Terashima-Marín[3]

[1] Department of Artificial Intelligence, University of Edinburgh,
80 South Bridge, Edinburgh EH1 1HN, UK, Email: peter@aisb.ed.ac.uk
[2] Parallel Emergent & Distributed Architectures Laboratory,
Department of Computer Science, University of Reading, Reading RG6 6AY, UK
Email: D.W.Corne@reading.ac.uk
[3] Centre for Artificial Intelligence, ITESM, Monterrey, Mexico
Email: terashim@cia.mty.itesm.mx

Abstract. Constraint satisfaction problems tend to display phase transitions with respect to the effort required by specific problem solving strategies. So far, little is known concerning the causes of phase transitions, or the relative differences between performance of different algorithms around them, especially with respect to stochastic iterative methods such as evolutionary search. Also, work so far on phase transitions concentrates on homogeneous random problems, rather than problems displaying elements of structure typical of more realistic problems. We investigate some of these issues, and uncover some new phase transition regions on timetabling style problems, occurring in the context of varying degrees of problem homogenity as well as (the more standard) graph connectivity. Further, we find that a simple evolutionary algorithm outperforms a simple Stochastic Hillclimber in regions strongly associated with certain phase transitions, and not others. Finally, we discuss various clues to the underlying causes of these phase transitions.

1 Introduction

Constraint satisfaction problems tend to display phase transitions with respect to the effort required by specific problem solving strategies. For example, Prosser [8] and Smith [10] describe phase transition behaviour in binary constraint satisfaction problems (CSPs). Both Prosser and Smith tested the performance of a range of complete heuristic search algorithms on binary CSPs of varying tightness and density. Both found relatively small regions in the tightness/density parameter space where it grew sharply difficult for the algorithm used either to find a solution where one existed or to prove that no solution existed. Smith [10] coined the term 'mushy region' to describe the region of problem space where a given algorithm's performance deteriorates in this manner.

This kind of behaviour can also be shown to happen in the case of stochastic search algorithms applied to timetabling problems. For these kinds of technique, such as stochastic hillclimbing (SH) or evolutionary algorithms (EAs), the algorithm itself has no way of proving reliably that no solution exists; performance can only be discussed in terms of the empirically determined likelihood of finding a solution and this sort of measure is only useful if it is certain that at least one solution exists. Therefore in what follows only solvable problems are considered.

For example the performance of SH on solvable timetabling-style problems can be characterised in terms of the likelihood of SH finding a solution as the *allowed constraint density* of the problem is varied from 0% to 100%; allowed constraint density refers to the proportion of constraints used from a given maximally-constrained problem, so that 100 % density refers to the point beyond which the addition of more constraints would render the problem unsolvable. Experiments reported below demonstrate the general existence of a phase-transition region in which it becomes sharply more difficult for SH to find optima at certain intermediate levels of density, returning quickly to easier performance as constraint density increases beyond this point.

It might be argued that randomly-created problems are unrealistic in some way. An aspect of this concerns the typical homogeneity of a randomly generated problem. In contrast, real timetabling problems tend to be 'clumped' in the sense shown in figure 1. Following a standard convention in which vertices in a graph represent events, and edges joining two vertices indicate that those two events must not overlap in time, then the simple graph on the left of figure 1 represents a fairly homogeneous problem. There is no sense in which the events fall into distinct groups. The graph on the right, however, has the same number of edges, but falls into two distinct subgraphs, or clumps. There is no edge joining any two vertices in different clumps. Between these two extremes can be imagined varying degrees of homogenity, in which there are distinct clumps, but a relatively small number of edges exist between them. Real timetabling problems are typically rather more clumped than homogeneous. For example, exams within an arts faculty may typically form a distinct clump, largely separate from those within a science faculty. Later on, we begin to examine the effects of such varying homogeneity in relation to problem difficulty.

On many real and realistic timetabling problems, it is clear that SH generally performs better than previously reported evolutionary algorithm (EA) approaches. This is the case with regard to those addressed in papers by Abramson & Abela [1] and by Corne et al [5]. Other studies have also shown that simple hillclimbing strategies can outperform EAs on various instances of other sorts of problem [7, 6]. However, further study suggests that there are areas in timetabling problem space where this situation is reversed. By looking at a space of timetabling problems which vary in terms of constrainedness and homogeneity, our aim in this paper is to begin to discover where these regions are.

2 Generating solvable problems

In simple timetabling problems a number of events each have to be assigned to one of a given number of timeslots and there are various constraints each stipulating that two specific events may not occupy the same timeslot. In practice problems are much more complicated than this, involving further issues such as