Is Genetic Programming a Sensible Research Direction for Timetabling?

E. Burke, S. Gustafson, and G. Kendall

ASAP Research, School of Computer Science & IT
University of Nottingham
{ekb | smg | gxk}@cs.nott.ac.uk

1 Overview

This presentation is concerned with discussing the applicability of genetic programming to automatic timetabling. Given that evolutionary algorithms and genetic algorithms have been successfully investigated for solving timetabling problems for many years[1][2][3][4][5], we are interested as to whether genetic programming[6], an evolutionary algorithm similar in some respects to genetic algorithms, would be suitable and beneficial for timetabling. The authors do not know the answer to this question but intend to explore and discuss it in the presentation. The field of genetic programming mainly differs from genetic algorithms in the representation (programs vs. bit strings) and the size of individuals/genomes (variable vs. fixed). The direct representation of timetables is straightforward as a bit string and genetic algorithms are a well established timetabling approach. Two questions are then raised when considering genetic programming for timetabling: 1) Why would we want a program to represent a timetable instead of a bit string? 2) Why would it be beneficial to use a variable length representation instead of fixed length? We consider three possible approaches for exploring genetic programming methods for a timetabling problem. Experiment design is underway to determine the feasibility of the following proposals.

1.1 Proposal 1

Typical meta/hyper heuristics applied to timetabling involve finding an appropriate combination or low-level heuristics to solve the timetabling problem. The number of heuristics used can be variable and the portion of the timetable that these heuristics are applied to can be variable. Thus, a genetic program which represents a hyper-heuristic (i.e. a 'heuristic to choose heuristics'), where program inputs represent portions of the timetable and program functions represent low-level heuristics, may be advantageous. A population of these individuals would represent different variations of hyper-heuristics that could evolve to produce a variety of feasible (or near) solutions. The motivation for this would be to develop timetabling systems that are fundamentally more general than the ones currently available.

1.2 Proposal 2

We could represent the timetable as a computer program in a similar way to the construction method used by Paechter[7], where the genetic algorithm represents a list of instructions to build the timetable. In this case the genetic program is made up of similar instructions, possibly the ones that low-level heuristics use. Instead of applying a fixed low-level heuristic (e.g. one that orders events for tabling) we could evolve these low-level heuristics to be specialised for a particular data set. This research could possibly extend ideas proposed in [8] and [9]. Also, since genetic programs are of variable length, the program of instructions would vary in length, allowing for simpler problems a shorter program and more difficult problems a longer, more complex program for building the timetable.

1.3 Proposal 3

Westerberg and Levine [10] discuss the seeding of genetic populations for planning problems. Because it may be advantageous in real-world timetabling for an officer to have several feasible (or near feasible) solutions, given a set of instructions that produce one timetable, we could seed a population of variations of these "programs" and evolve them by employing a fitness function that encourages diverse and near feasible solutions. In this manner, the officer would be presented with a constantly evolving population of possible timetables that might satisfy unrealised soft constraints more adequately than the original timetable. Ross and Corne emphasise the benefit of the genetic algorithm to produce a variety of possible timetables in [11].

2 Conclusions

The literature on timetabling and evolutionary algorithms suggest that indirect representations and combinations of heuristics is a promising research direction. With that in mind, we propose that a combination of the above proposals 1 and 2 will be suitable. Evolving computer programs, with variable length and structure, that represent the selection of heuristics for different aspects of building a timetable given our specific problem is the tenative direction of our research. Considerations of the genetic programming algorithm details and analysis guidelines are future work prior to implementation. The authors look forward to discussing these very initial ideas at the conference.

References

1. M.W. Carter and G. Laporte. Recent developments in pratical examination timetabling. In *The Practice and Theory of Timetabling: Selected Papers from the 1st Int'l Conf. on the Practice and Theory of Automated Timetabling, 1995.* Springer-Verlag LNCS 1153, 1996.

2. V.A. Bardadym. Computer-aided school and university timetabling: The new wave. In *The Practice and Theory of Timetabling: Selected Papers from the 1st Int'l Conf. on the Practice and Theory of Automated Timetabling, 1995*. Springer-Verlag LNCS 1153, 1996.
3. E. Burke, J. Kingston, K. Jackson, and R. Weare. Automated university timetabling: The state of the art. *The Computer Journal*, 40(9):565–571, 1997.
4. M.W. Carter and G. Laporte. Recent developments in practical course timetabling. In *The Practice and Theory of Timetabling II: Selected Papers from the 2nd Int'l Conf. on the Practice and Theory of Automated Timetabling, 1997*. Springer-Verlag LNCS 1408, 1998.
5. A. Schaerf. A survey of automated timetabling. *Artificial Intelligence Review*, 13(2):87–127, 1999.
6. W. Banzhaf, P. Nordin, R. E. Keller, and F. D. Francone. *Genetic Programming: An Introduction*. Morgan Kaufmann, Inc., San Francisco, USA, 1998.
7. B. Paechter. Optimising a presentation timetable using evolutionary algorithms. In *Proceedings of the AISB Workshop on Evolutionary Computation*. Springer-Verlag, 1994.
8. P. Ross, E. Hart, and D. Corne. Some observations about ga-based exam timetabling. In *The Practice and Theory of Timetabling: Selected Papers from the 2nd Int'l Conf. on the Practice and Theory of Automated Timetabling, 1997*. Springer-Verlag LNCS 1408, 1998.
9. H. Terashima-Marin, P. Ross, and M. Valenzuela-Rendon. Evolution of constraint satisfaction strategies in examination timetabling. In W. Banzhaf et al., editors, *Proceedings of the Genetic and Evolutionary Computation Conference*, volume 1, pages 635–642, Orlando, Florida, USA, 13-17 July 1999. Morgan Kaufmann.
10. C.H. Westerberg and J. Levine. Investigatioons of different seeding strategies in a genetic planner. In E.J.W. Boers et al., editors, *Applications of Evolutionary Computing*, volume LNCS 2037, pages 505–514, Lake Como, Italy, April 2001. Springer-Verlag.
11. P. Ross and D. Corne. Comparing genetic algorithms, simulated annealing, and stochastic hillclimbing on timetabling problems. In T.Fogarty, editor, *Evolutionary Computing; AISB Workshop, Sheffield, Selected Papers*. Springer-Verlag LNCS 993, 1995.