Further Investigations into the Evolution of Agents with Concurrent Genetic Programming

Adrian Trenaman
Department of Computer Science, National University of Ireland,
Maynooth, Co. Kildare, Ireland.
Phone: +353-1-7083354 Email: trenaman@cs.may.ie

1 Introduction

This paper follows from previous work by the author in [1, 2], where controllers were evolved for agents in virtual block-worlds. The agent's behaviour is determined by a set of genetically evolved programs executed in a concurrent fashion. The motivation behind this work is threefold: first, to investigate the use of concurrency in genetic programming, second, to examine the ability of this paradigm to effect emergent agent behaviour, and third, to assess the efficacy of implicit state mechanisms in building effective internal state. Teller's *Tartarus* world and Ashlock's *Dozer* world were chosen as benchmarks because they require intelligent use of state for good performance and require relatively little computational overhead.

An early implementation of concurrent genetic programming used the Java multi-threaded runtime environment to provide the underlying mechanism for concurrent evaluation [1], however results were hard to interpret due to the non-determinism of the runtime-scheduler. A deterministic runtime environment for concurrent genetic programming was introduced in [2], and used to evolve agents made up of 1, 2 and 4 programs evaluated in a fine-grained manner. This paper adds to these results by investigating more fully the evolution of agents with up to 5 programs using different levels of concurrency from fine- to coarse-grained.

2 Experiments and Results

Each experimental run was characterised by the number of programs in the agent (1, 2, 3, 4 or 5), the optional inclusion of indexed memory into the function set, and the *program energy* (described in [2]), controlling the granularity of concurrency. Program energy was set to 1, 2, 5, 10 or *infinite*-energy, and agents used a `Yield` function to evolve scheduling strategies around the deterministic round-robin scheduler. In this way, the full gamut of concurrency from highly parallel fine-grained to evolved multi-threaded programs could be investigated. 84 experiments were identified (42 for each world), and each was run 20 times with different random seeds.

Concurrent agents were evolved scoring significantly better than single-tree agents: two-tree coarse-grained agents scored 5.9 points out of 10 for Tartarus and 13.3 points out of 30 for Dozer. The standard deviation of best fitness for each experiment was generally large, making it hard to make strong statistical statements, however, some clear trends emerged. The concurrent approach produces individuals of good fitness, *regardless* of the level of concurrency. In general, the mean-best-fitness of coarse-grained agents is poorer than that of fine-grained, however, the higher standard deviation observed in these experiments reveals the presence of outliers of high fitness. The best results consistently come from two-tree individuals: there is no clear advantage in using greater numbers of trees. The use of indexed memory with the concurrent approach gives only a minor improvement in fitness and is not necessary for high performance: agents without indexed memory evolved to exploit the characteristics of the deterministic runtime scheduler to build an implicit form of internal state, effecting sequential behaviour strategies.

3 Conclusions

The concurrent approach affords an alternative to explicit state mechanisms for agents in Tartarus and Dozer. The effectiveness of this implicit state must be addressed in future work with more complex environments where sequential behaviour strategies are insufficient for good performance.

Acknowledgements

The author wishes to thank Prof. David Vernon of NUI Maynooth for his continuous guidance and assistance in this work.

References

[1] Adrian Trenaman. Concurrent genetic programming and the use of explicit state to solve problems in partially-known environments. In John R. Koza et al., editors, *Genetic Programming 1998: Proceedings of the Third Annual Conference*, Madison, Wisconsin, July 1998. Morgan Kaufmann.

[2] Adrian Trenaman. Concurrent genetic programming, Tartarus and dancing agents. In Peter Nordin, editor, *EuroGP '99, Proceedings of the* 2^{nd} *European Workshop on Genetic Programming*, Goteburg, Sweden, 1999. Springer Verlag.