Strengths and Weaknesses of FSA representation

Pavel Petrovic
Department of Computer and Information Science
NTNU, 7491 Trondheim, Norway, ppetrovic@acm.org

Categories and Subject Descriptors

I.2 ARTIFICIAL INTELLIGENCE
I.2.4 Knowledge Representation Formalisms and Methods [**Representations (procedural and rule-based)**]:

General Terms

Performance

Keywords

Evolutionary Programming, Finite State Automata, Incremental Evolution

1. FSA REPRESENTATION

We used FSA [2] as a representation of behavior-arbitration mechanism for behavior-based mobile robot controllers designed by the means of EC. State-based representations share structural similarity with robotic tasks: robots always stay in states while reactively responding with immediate actions and proceeding to other states as a response to environmental percepts. The activity of a robotic agent can be modelled by a state diagram accurately thus FSA are a suitable platform for robot controllers representation. They are easier to understand, analyze, and verify than neural networks. Evolutionary roboticists often conclude that simple evolutionary run is not sufficient for evolving complex behaviors. Incremental Evolution provides a possible scenario for improving the evolvability. Our hypothesis is that the FSA representation is suitable for incremental construction of the controller. In this work, we analyzed several example problems in order to study the performance of the FSA representation. We compare the performance of the FSA representation to common GP platform on symbolic sequence-processing problems, and simple robotics tasks. The properties of the symbolic problems are studied from the perspective of robotics tasks. See [3] for treatment of related work, formal specifications and details of the experiments. Advanced operators of homologic crossover, and brooding crossover[1] were used to increase crossover success rate.

GP-tree programs tend to have a linear path of execution, the FSA are powerful in representing repeated and possibly irregular patterns and behaviors that react to percepts and possibly launch different mode (or state) of operation. Making the EA incremental can both help and hinder the success rate of the EA. Incremental evolution introduces *incremental bias*, requiring the evolution to pass through stages that could possibly be avoided in a single run, but it can be compensated by benefits in performance.

GECCO'07, July 7–11, 2007, London, England, United Kingdom.
ACM 978-1-59593-697-4/07/0007.

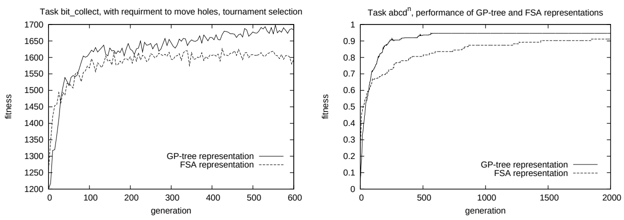

Figure 1: Performance of the FSA and GP-tree representations on symbolic tasks without sufficient repetitive interactions. The GP representation outperforms the FSA representation. Average of best indiv. in each generation from 10+/23+ runs.

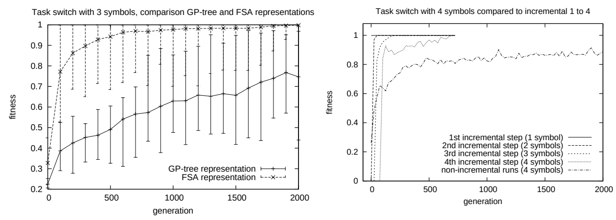

Figure 2: Average of the best fitness from 10+ runs on a symbolic task with repetitive FSA representation. FSA representation outperforms the GP-tree representation (left). Incremental evolution speeds up evolution and increases evolvability (right).

2. REFERENCES

[1] W. Banzhaf, P. Nordin, R. E. Keller, and F. D. Francone. *Genetic Programming, An Introduction.* Morgan Kaufmann Publishers, Inc., 1998.

[2] P. Petrovic. Evolving automatons for distributed behavior arbitration. Technical Report IDI 05/05, Norwegian University of Science and Technology, 2005.

[3] P. Petrovic. Comparing finite-state automata representation with gp-trees. Technical Report IDI 05/06, 2006.