Strengths and Weaknesses of FSA representation

NTNU
Norwegian University of Science and Technology

Pavel Petrovič, ppetrovic@acm.org
Department of Computer and Information Science

ABSTRACT

Genetic Programming and Evolutionary Programming are fields studying the application of artificial evolution on evolving directly executable programs, in form of trees similar to Lisp expressions (GP-trees), or Finite State Automata (FSA). In this exercise, we study the performance of these methods on several example problems, and draw conclusions on the suitability of the representations with respect to the task structure and properties. We investigate the role of incremental evolution and its bias in the context of FSA representation. The experiments are performed in simulation and/or confirmed on real robots.

Keywords: Evolutionary Programming, Finite State Automata, Incremental Evolution

REPRESENTATION

Figure 1. Illustration of GP and FSA representations.

GP mutation:
mut change, mut exchange, mut insert, mut remove, random node;

FSA mutation:
mut change, mut exchange, mut insert, mut remove, random fsa, mut insert

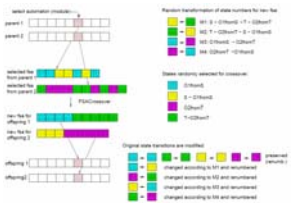


Figure 2. FSA Crossover.

TASKS

bit collect:
• fill all holes (easy) or pack ones (hard)
• read/write pointer on finite tape
• operations *left*, *right*, *write0*, *write1*, *done*
• example: 10111001010001 to 11111111111111 (easy) or to 11111110000000 (hard)

$(abcd)^{n>}$
• repeating a fixed pattern
• same settings
• example: 1111111111111111111 to abcdabcdabcdabcdabcd

switch:
• unpredictable interaction
• same settings
• example: 10004030000200013004000000003000020 to 11114433333222213334444444443333322

find_target:
• navigating in 2D area with obstacles
• operations: *fd, bk, fdlong, bklong, lt, rt, done*
• example: figure 3

dock:
• robot docking into prescribed rectangle
• sensor sensitivity: *stopOn, stopOff*
• example: figure 4

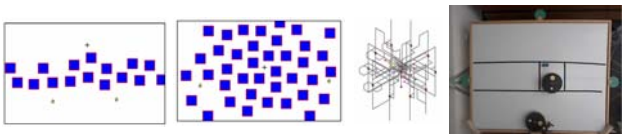
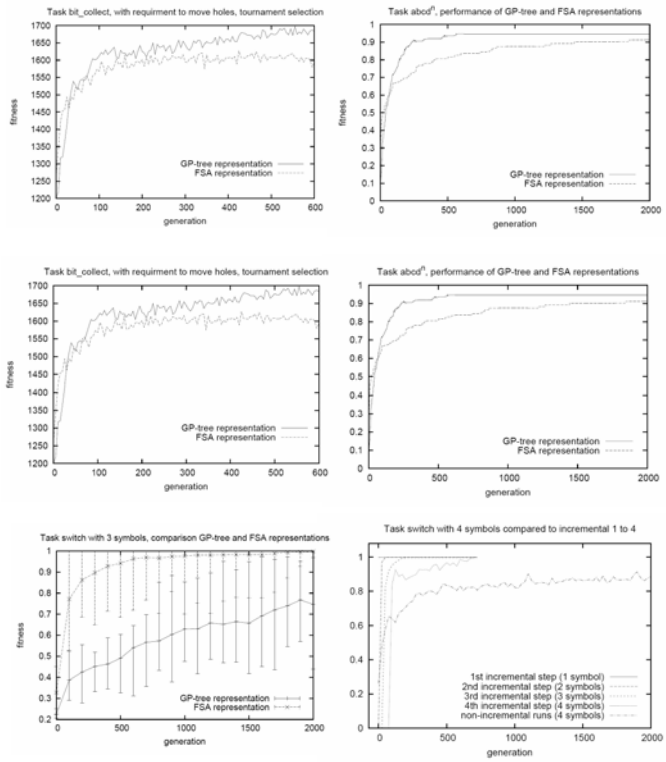


Figure 3. 2D environments for find_target and dock task.

RESULTS



CONCLUSIONS

- Both GP and FSA representations outperform each other on different tasks
- FSA is suitable when the task requires switching between different tasks/contexts
- Incremental Evolution successfully improves evolvability with FSA representation, but care must be taken for the cost of the incremental bias.

REFERENCES

[1] P. Petrovic. Evolving automatons for distributed behavior arbitration. Technical Report IDI 05/05, Norwegian University of Science and Technology, 2005.
[2] P. Petrovic. Comparing finite-state automata representation with gp-trees. Technical Report IDI 05/06, 2006.