Matching Hyper-heuristics and Genetic Programming

Students: Sean N. Harris & Travis J. Bueter
Natural Computation Laboratory
Department of Computer Science

Mentor: Matthew A. Martin
Natural Computation Laboratory
Department of Computer Science

Faculty Advisor: Dr. Daniel R. Tauritz
Natural Computation Laboratory
Department of Computer Science

SAND2015-2325C

Motivation

Modern society is faced with ever more complex problems, many of which can be formulated as generate-and-test optimization problems. General-purpose optimization algorithms are not well suited for real-world scenarios where many instances of the same problem class need to be repeatedly and efficiently solved, such as routing vehicles over highways with constantly changing traffic flows, because they are not targeted to a particular scenario. Hyper-heuristics automate the design of algorithms to create a custom algorithm for a particular scenario.

Project Objective

Hyper-heuristics typically employ Genetic Programming (GP), an evolutionary algorithm-based methodology inspired by biological evolution, and this project aimed to investigate the relationship between the choice of GP and performance in Hyper-heuristics. This poster presents results demonstrating the existence of problems for which there was a statistically significant performance differential between the use of different types of GP. Also presented are some preliminary findings on how to match the type of GP employed in a hyper-heuristic with the problem being addressed.

Approach

- 5 GP techniques were chosen to be evaluated:
  - Tree-based
  - Linear
  - Cartesian
  - Grammatical
  - Stack-based
- Each GP used the same pool of primitives to evolve solutions.
- Each GP was employed to evolve iterative search algorithms for solving a Boolean satisfiability problem (SAT) with 2000 clauses and 500 variables.
- Each GP ran 30 times, with each run allowed 40 generations, and each algorithm allowed 100 generations.
- The most effective algorithms generated were then used to quantify the effectiveness of each GP technique.

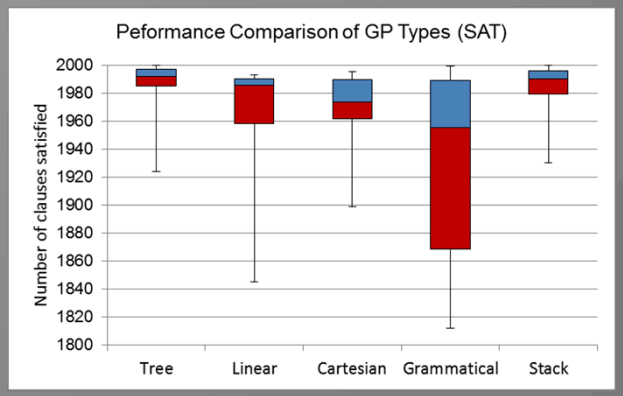

Figure 1: Performance comparisons are of the best individuals generated over 30 runs per GP.

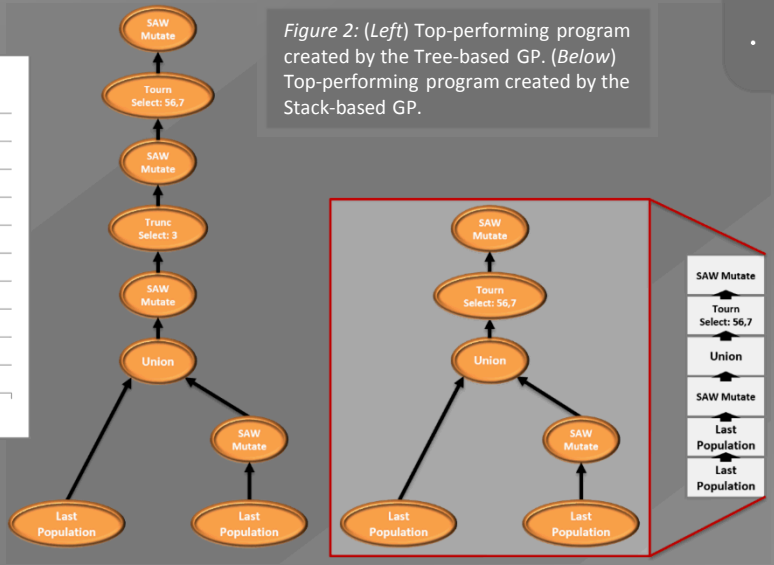

Figure 2: (*Left*) Top-performing program created by the Tree-based GP. (*Below*) Top-performing program created by the Stack-based GP.

Tree vs. Stack

- Tree- and stack-based GPs not only equivalently outperformed the other GPs, but also evolved similar solutions.
- Tree-based GP had a greater density of good solutions than stack-based GP.
- Tree-based GP's genetic operators respect locality during evolution whereas stack-based GP's do not.
- Stack-based GP's structure allows for introns whereas tree-based GP's does not.
- Stack-based GP's allowance of introns could be enhancing its performance while its low-locality could be hindering it.

Conclusion

- Results demonstrate a statistically significant benefit of using tree- or stack-based GP.
- Stack-based GP shows potential of being a viable alternate to tree-based GP where it has been traditionally used.
- Results show the choice of GP does have a strong impact on the quality of algorithms produced.
- Evidence of the importance for any work with hyper-heuristics to carefully consider which type of genetic programming is used.

Future Work

- Greater focus on testing tree- and stack-based GP, and comparing their performance.
- Expand testing to problem configurations ideal for an individual GP and compare how the others GPs perform against it.
- Implement GP-specific primitives that take advantage of unique traits to improve its performance.
- Develop system for matching the optimal GP for a given problem configuration.

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.