that can be solved in practice. In this section we benchmark the tree and graph encodings.

Figure 6 shows the computational performance, measured in point evaluations per second over a range of complexities. The graph encoding remains roughly constant because it has a fixed encoding size. Variation still exists because it still executes operations in its list that do not affect the output.

The tree encoding is efficient on simple functions of less than five nodes. Performance drops significantly with complexity however as recursion deepens with complexity. The computational performance result indicates the tree encoding does not scale as well with complexity. At five nodes and higher, the graph encoding using an operator list more than triples the performance of the tree encoding.

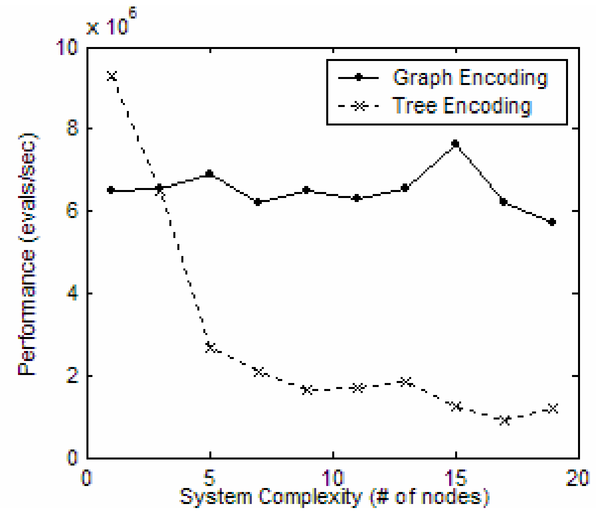

Figure 6. The point evaluations per second versus the function complexity.

11. CONCLUSIONS

We have compared two encoding schemes in increasingly complex problems using symbolic regression. While the tree and graph encodings are similar in application, they offer distinct advantages and disadvantages in genetic programming.

We have tested these two encodings on randomly generated nonlinear target functions, for both single variable and 8-variable input spaces.

Results show that the tree encoding solutions exhibit consistently higher bloat over all complexity targets. The tree encoding however offers slightly higher convergence rate (finding an exact fit) and time to converge, but there was no large trend difference over complexity. The tree encoding experiences more beneficial crossovers (offspring more fit than most similar parent) on single variable targets. Beneficial crossovers decrease with complexity. On 8-varible targets both encodings experienced similar trends in beneficial crossover trends. Finally, the computational comparison shows that the graph encoding to be significantly more efficient than the graph for high complexities.

From these results we conclude the graph encoding to be a attractive alternative to traditional tree based problems (eg. symbolic regression). Graph encodings provide similar performance in convergence over a range of complex target functions and different input sizes, and do so with less bloat. The graph encoding experiences fewer beneficial crossovers and converges slightly slower, however the computational performance outweighs this drawback.

12. ACKNOWLEDGEMENTS

This research was supported in part by the U.S. National Science Foundation grant number CMMI-0547376.

13. REFERENCES

[1] Koza, J.R. Genetic Programming: On the Programming of Computers by Means of Natural Selection. Cambridge, MA: The MIT Press, 1992.

[2] Augusto D. A. and Barbosa H. J. C. "Symbolic Regression via Genetic Programming," VI Brazilian Symposium on Neural Networks (SBRN'00), 01: 22-01, 2000.

[3] Schmidt, M., and Lipson, H. "Coevolution of Fitness Maximizers and Fitness Predictors", GECCO Late Breaking Paper, 2005.

[4] Rafal Kicinger, Tomasz Arciszewski and Kenneth De Jong, “Evolutionary computation and structural design: A survey of the state-of-the-art,” Computers & Structures, Volume 83, Issues 23-24, Pages 1943-1978, 2005.

[5] Duffy, J., and Engle-Warnick J., "Using Symbolic Regression to Infer Strategies from Experimental Data," S-H. Chen eds., Evolutionary Computation in Economics and Finance, Physica-Verlag. New York, 2002.

[6] Hoai, N. McKay R., Essam D., and Chau R., "Solving the symbolic regression problem with tree-adjunct grammar guided genetic programming: comparative results," Evolutionary Computation, Vol 2. pp. 1326-1331, 2002.

[7] Rothlauf, F. "Representations for Genetic and Evolutionary Algorithms." Physica, Heidelberg 2002.

[8] Morales, C.O. "Symbolic Regression Problems by Genetic Programming with Multi-branches," MICAI 2004: Advances in Artificial Intelligence, pp.717-726, 2004.

[9] Caruana, Rich, Schaffer, J. David, "Representation and Hidden Bias: Gray vs. Binary Coding for Genetic Algorithms." Fifth International Conference on Machine Learning, 1988.

[10] Breiman, L. “Bias, variance, and arcing classifiers,” Technical Report 460, Statistics Department, University of California Berkeley, 2996.

[11] Domingos, P. “A unified bias-variance decomposition and its applications,” In Proceedings of the 17th International Conference on Machine Learning, 2002.

[12] Wolpert, D. “On bias plus variance,” Neural Computation, 9, pp. 1211–1243, 1997.

[13] McKay, B., Willis, M., Barton, G. "Using a tree structured genetic algorithm to perform symbolic regression," First International Conference on Genetic Algorithms in Engineering Systems, vol. 414, pp. 487–492, 1995.

[14] Lipson H. "Principles of Modularity, Regularity, and Hierarchy for Scalable Systems," Genetic and Evolutionary Computation Conference (GECCO'04), 2004.