A Swarm-based Crossover Operator for Genetic Programming

Tony White
School of Computer Science
Carleton University
Ottawa, Canada
arpwhite@scs.carleton.ca

Amirali Salehi-Abari
School of Computer Science
Carleton University
Ottawa, Canada
asabari@scs.carleton.ca

ABSTRACT
A swarm-based improvement to Genetic Programming (GP) is described and tested on the domain of symbolic regression in this paper. The motivating idea is to keep all of the benefits of genetic programming such as crossover and fitness proportional selection within a population of candidate solutions. The improvement comes in using swarm-based ideas similar to Ant Colony Optimization (ACO) to improve the operation of the crossover operator. Statistically significant results are reported in support of the hypothesis that ACO-inspired crossover can improve GP.

Categories and Subject Descriptors
I.2.2 [**Computing Methodologies**]: *Artificial Intelligence – program modification, program synthesis.*

General Terms
Algorithms, Measurement, Performance, Design, Experimentation, Languages, Theory.

Keywords
Automatic Programming, Genetic Programming, Swarm-Based Algorithms, Ant Colony Optimization, Evolutionary Computation.

1. INTRODUCTION
Automatic programming is an active research area that has been stimulated by the development of Genetic Programming (GP) and swarm-based algorithms such as the Ant Colony Optimization (ACO) metaheuristic.

The contribution of this paper is the creation of a swarm-based crossover operator useful for Genetic Programming. Experiments in the domain of symbolic regression are performed in order to demonstrate the utility of the proposed operator. Statistically significant results are provided for several test cases.

2. PROBLEM BACKGROUND
The specific problem that will be examined for the swarm-based improvement to genetic programming will be the Symbolic Regression problem. The task in symbolic regression is to find a function that best fits a number of sample points.

Given a set D with sample points { x(i), y(i) }, the fitness, f, of any program k may be given by:

$$f(k) = h(k) - \frac{1}{\max(h(k),1)} \sum_{i=1}^{size(D)} e(k,i)$$

$$e(k,i) = abs\big((v(k,x(i)) - y(i))\big)$$

$$v(k,x) = Value of\, k^{th}\ program\, for\, x$$

$$h(k) = \sum_{i=1}^{size(D)} hits(k,i)$$

$$hits(k,i) = \begin{matrix} 0 & if\, e(k,i) \leq \varepsilon \\ 1 & otherwise \end{matrix}$$

The maximum fitness for any program is the size of the set D, which occurs when the function passes through all of the sampled points.

3. APPROACH
3.1 Function Couplings
One possible way in which GP could be improved is during crossover; in fact, context-aware crossover has demonstrated such improvements [2] and a study of the search properties of several candidate crossover operators undertaken [3]. A survey of probabilistic model building using GP is presented in [1].

The hypothesis here is that by keeping important couplings during crossover and instead breaking less useful ones should lead to an improvement in the genetic programming algorithm. The principal task is how to determine the importance of different function and terminal couplings. To this end, a matrix, C, is kept that represents a coupling from every function to every other function or terminal.

3.2 Modified Crossover
Once the important couplings are determined, crossover can be modified to preserve important couplings. A random branch from the root to the leaf of the tree is chosen. The pheromone levels for node couplings for every edge of the branch are considered. The probability of choosing a node as root of subtree for crossover is proportional to the amount of pheromone for the coupling with its parent node. The optimization algorithm is summarized below:

In this algorithm, tree edges are labeled in a depth-first manner.

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.
ACM 978-1-60558-130-9/08/07.

Pheromone initialization:

For every function and terminal coupling (i, j) do

Initialize pheromone, $\tau_{i,j}$, to initial value, τ_0

For each generation:

Pheromone Update:

For the best k programs, P_n

For every edge i, j in program tree P_n

$C(V(n,i),V(n,j)) \leftarrow (1-\rho)\, C(V(n,i),V(n,j)) + h(n) / size(D)$

where ρ is the pheromone evaporation rate,

C(i,j) = the strength of the coupling between the i^{th} function and the j^{th} function or terminal

V(n,i) = the index value of node or terminal at the i^{th} node in the n^{th} program tree and

size(D) = the size of the set of sampled points which represents the maximum possible fitness of any program.

Crossover for programs P_n and P_m:

Choose a random branch, B, from root to a leaf in program tree $\mathbf{P_n}$

For every edge i, j in B

Probability of choosing node i as root of subtree S_n

where i is parent and j is a child node is given by:

$$p(i, n) = (\tau_{max}(n) - \tau_{min}(n) + \tau_{i,j}(n)) / T(n)$$

Choose random branch, B, from root to a leaf in program tree $\mathbf{P_m}$ use equation above to choose crossover node.

where:

$$T(k) = \Sigma_{i,j \in E(k)} (\tau_{max}(k) - \tau_{min}(k) + \tau_{i,j}(k)) \text{ and}$$

$$\tau_{i,j}(k) = C(V(k,i),V(k,j))$$

$$\tau_{max}(k) = \max_{i,j \in E(k)} (\tau_{i,j}(k))$$

$$\tau_{min}(k) = \min_{i,j \in E(k)} (\tau_{i,j}(k))$$

$$E(k) = \{ \text{edges in } k^{th} \text{ program subtree} \}$$

Swap S_n and S_m in programs $\mathbf{P_n}$ and $\mathbf{P_m}$ using the normal GP crossover mechanism.

4. RESULTS

Three test functions were used with varying population sizes. Results for the mean and standard deviation number of generations taken to find the function are shown in Table 1.

F1: $\cos(X^2)+\sin(X^2)+X^2$

F2: $\cos(X^2)+\sin(X^2)+X^2+\cos(X)+\sin(X)$

F3: $\sin(X)*X^4+\sin(X)*X^3+\sin(X)*X^2+\sin(X)*X$

While the results in Table 1 are statistically significant, it is important to understand whether the SB-GP algorithm is learning useful function couplings. From Figure 1 it can clearly be seen that the SB-GP algorithm has learned important function couplings.

Table 1. Testing results

Test	GP Mean	GP STD	SB-GP Mean	SB-GP STD	P Value	Population Size
F1	6.6	2.01	4.35	0.75	0.0001	1500
F1	24.73	18.9	6.18	2.89	0.0043	500
F1	37.7	19.8	24.4	22.1	0.1745	200
F2	27.8	19.6	11.6	4.69	0.0043	800
F3	43.1	14.5	28.2	16.8	0.0488	500

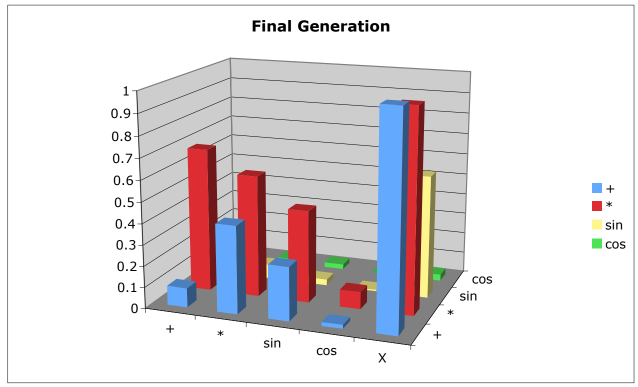

Figure 1: Single Experiment Example Couplings for F3

5. CONCLUSIONS

The swarm-based crossover operator was found to lead to a statistically significant reduction in the number of generations required to converge to an optimal solution in several cases of the symbolic regression problem.

6. REFERENCES

[1] Shan Y., McKay R., Essam D., Abbass H., A Survey of Probabilistic Model Building using Genetic Programming. Technical Report TR-ALAR-200510014, The Artificial Life and Adaptive Robotics Laboratory, School of Information Technology and Electrical Engineering, University of New South Wales, Australia, 2005.

[2] Hammad M., and Conor R., Using context-aware crossover to improve the performance of GP, Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation (GECCO-2006), pages 847-854, Seattle, Washington, July 2006. Morgan Kaufmann.

[3] Riccardo Poli and William B. Langdon. On the search properties of different crossover operators in genetic programming. In Genetic Programming 1998: Proceedings of the 3rd Annual Conference, pages 293-301, University of Wisconsin, Madison, Wisconsin, USA, 22-25 July 1998. Morgan Kaufmann.