

Gary D. Boetticher  
 Department of Computer Science

Kim Kaminsky

## The Impact of Chromosome Lineage upon Genetic Program Modeling

**Abstract**—One of the challenges in data mining is the task of providing sufficient coverage of the search space in order to produce an acceptable model. Traditionally, Genetic Programs (GP) consider all chromosomes within a population for breeding purposes. Considering the enormity of the search space, it is imperative to focus the breeding efforts of Genetic Programs in order to attain a better solution in less time. This research examines the lineage of Genetic Programs in order to identify any breeding patterns. Five separate experiments had chromosomes grouped into five classes. Lineage patterns were assessed for the best, middle, and worst-class parental chromosomes. Based upon the results, researchers propose a new genetic programming process.

**T**RADITIONALLY, GENETIC PROGRAMS (GP) SOLVE PROBLEMS by generating a set of mathematical equations, or chromosomes, that represent a mapping between two sets of variables. Collectively, these chromosomes form a population. GPs repetitively breed new generations of chromosomes seeking to find an optimal, or at least satisfactory, solution. Normally, a GP experiment runs until a satisfactory solution is found; the GP runs for  $n$  generations, or the user terminates the learning.

GPs are frequently deployed on large, complex, noisy datasets where the search space is extremely large. Finding a solution within the search space is an extremely difficult challenge. For example, theoretically there may be billions of equation permutations. If a GP consists of 1000 chromosomes and runs for 1000 generations, then the GP generates at most one million possible equations (1000 chromosomes \* 1000 generations). Such an experiment would cover only 0.1 percent of the total search space.

This argument assumes that GPs are static in structure. It is well known that GPs rapidly increase in size as the population evolves,<sup>14</sup> thus increasing the expanse of the search space and reducing the probability that a solution will be found.

Solving large problems using GPs consumes excessive amounts of computer resources.<sup>4</sup> Though Genetic Programs may successfully evolve solutions to complex problems, their use may sometimes be cost-prohibitive. What is desired is a more efficient approach to exploring the search space. This may be accomplished qualitatively by focusing the search efforts or quantitatively by increasing the number of searches.

This research explores the qualitative approach by examining the breeding patterns of a GP. A key question is, Does chromosome lineage information provide any insight into the effectiveness of solving problems? If so, researchers ask how these insights could be used to make better breeding decisions?

Gaining a better understanding about a chromosome's lineage could be beneficial in several ways. Greater emphasis could be placed on those chromosomes that produce better offspring. Secondly, the utility of such a discovery could focus search efforts, thus reducing the training time, and requiring less computing resources. All these benefits are immensely important when applying GPs to large, complex, noisy problem spaces.

To explore the role chromosome lineage plays in the breeding process; five experiments are conducted using synthetic datasets. Chromosomes have been clustered into different classes (e.g. upper, middle, and lower classes). Each of these classes is tracked over a generation to determine whether certain classes are prone to producing good or poor solutions.

### Related Research

McPhee et al.<sup>5</sup> analyze node level genetic diversity in a GP population over its genetic history. They observe that there is a profound loss of diversity over the evolution process indicating that a standard GP does not perform opportunistic breeding.

Burke et al.<sup>6</sup> use lineage selection to increase diversity by reducing the selection pressure from "most fit" to "fit and diverse." They find that introducing diversity can avoid get-

ting trapped in local optima.

Both McPhee and Burke use lineage information as a method to promote diversity within the population. This research uses lineage information more as a mechanism to improve the selection process.

### How Genetic Programs Work

Genetic Programs solve problems by genetically breeding a population of individuals, or chromosomes, over a series of generations. Inspired by theories of evolution, Genetic Programs use the analogy of evolutionary operators on chromosomes to optimize a fitness function. A fitness function assesses the goodness of a chromosome (represented as an equation) in terms of how well (or poorly) that equation fits a given dataset. The goodness of a chromosome serves as the basis for propagation decisions.

An implementation of a Genetic Program starts with a population of individual equations, usually represented as tree structures. Each tree, or chromosome, can be viewed as a potential solution to the given problem (training data). Each node on the tree represents a gene, or some trait within a problem. Programmatically, each gene corresponds to either an operator or an operand. Collectively, the set of the genes would make up a mathematical expression.

Collectively the set of chromosomes, which represent potential solutions, are known as a population. This population 'reproduces' to create a future generation. Each iteration of a Genetic Program produces a new generation of individuals.

After the population has been initialized and the fitness of each individual has been evaluated, the selection of parent chromosomes occurs. During selection, the fittest individuals are selected to engage in reproduction. A fitness function evaluates the individuals and ranks them in terms of performance. The fitness function is defined as follows:

$$Fitness = 1 + e^{(7 \cdot (1 - n - k)/(n - 1) \cdot Se^2/Sy^2)}, \quad (1)$$

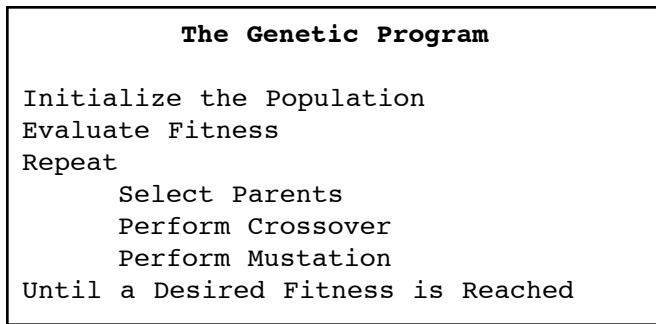
where

$k$  corresponds to the number of inputs,  
 $n$  represents the number of valid results,  
 $Se$  is the standard error, and  
 $Sy$  equals the standard deviation.

This allows a range of values from 1 through 1067. These values are scaled to span 1 through 1000.

Chromosomes are paired together by their fitness for breeding purposes. Once two individuals within a population have been chosen, several reproductive, or recombination, operations may occur. One example is crossover. The Crossover process takes a subtree from each chromosome parent, chooses a random branch, and then crosses over the genetic material. Crossover occurs at one point, or at several points, within each chromosome.

A second step in the breeding process is mutation. Mutation also alters the genetic material of a chromosome. It prevents a solution from falling into local minima or maxima, which is



**Figure 1. GP Algorithm**<sup>7,9</sup>

a problem experienced by most optimization algorithms.<sup>7,8</sup> While crossover exchanges genetic material between two chromosomes, mutation randomly selects and changes some genes within one chromosome and passes this change onto the offspring. Mutation randomly selects a node in the tree and changes the genetic material. Mutation promotes diversity within a population by randomly adding in gene variations.

This process of creating a population of individuals, ranking the individuals by fitness and recombining these individuals to produce a better set of solutions, is called a generation. The modeling process spans multiple generations until an acceptable solution is found, the experiment runs for a specified number of generations, or the user terminates the run. Figure 1 shows the general Genetic Programming algorithm.

**Proposed Research**

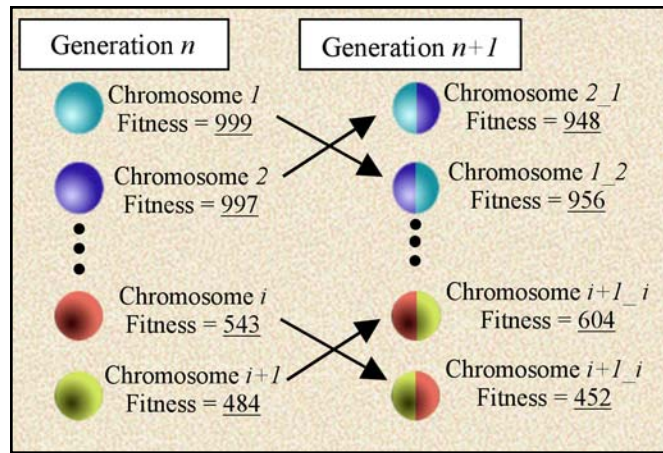
In the GP modeling process, once a new generation is created, all legacy information about the previous generation is discarded. Perhaps this discarded ancestral fitness information could offer valuable clues on how to make better propagation decisions. This research statistically analyzes a chromosome’s lineage, in terms of the fitness values across generations. If there is a correlation between fitness values across generations, then it would be possible to focus only on those chromosomes with a good pedigree. This could lead to more accurate GP solutions requiring less training time.

Figure 2 illustrates the tracking of fitness values across multiple generations. In the figure above, two of the most fit chromosomes (1 and 2 in generation *n*) produce fitter children in generation *n + 1*.

It is possible to trace lineage back several generations (e.g. grandparents, great-grandparents). However, this research only considers the previous generation in the lineage assessment process.

**GP Experiments**

Five experiments have been conducted using synthetic data sets where the dependent values are clearly defined. Using synthetic data reduces the fuzziness of problems and makes it possible to adequately test the theory. The Genetic Program must not solve the equations easily, since this experiment requires a large number of samples. However, the Genetic Program needs the ability to model the equation easily. A difficult problem causes a Genetic Program to get trapped in



**Figure 2. Tracking Fitness values For Two Generations**

local minima. In this case, the Genetic Program may not grow closer to the solution and the differences in fitness values over time may not be clearly illustrated. Therefore, the five equations chosen may be solved easily by the Genetic Program within a few generations. To keep the Genetic Program from solving the programs, a very small random number is added to each dependent variable for all instances. This makes it impossible for the Genetic Program to solve the problem, while allowing the Genetic Program to come very close to the actual solution.

Each of the five datasets is based on one of the following five equations:

$$Z = W + X + Y \tag{2}$$

$$Z = 2 \cdot X + Y - W \tag{3}$$

$$Z = X / Y \tag{4}$$

$$Z = X^3 \tag{5}$$

$$Z = W^2 + W \cdot X - Y \tag{6}$$

The complexity of the equations becomes progressively complicated for each experiment.

All experiments define a generation as 1,000 chromosomes. The selection method pairs chromosomes based on their fitness rank, with the top two fittest individuals mating, then the next two, etc.

Each experiment runs for 50 generations. For every generation, the 1000 chromosomes are sorted by fitness values then divided into five distinct groups of 200 chromosomes each. The fitness values of the offspring are recorded for each pair of parent chromosomes and the average of all fitness values within a group is calculated.

The next step compares the offspring’s fitness values for those parents who had the best 200 chromosomes (the best-class) with the offspring’s fitness values of those parents who had the middle 200 chromosomes (the middle class), along

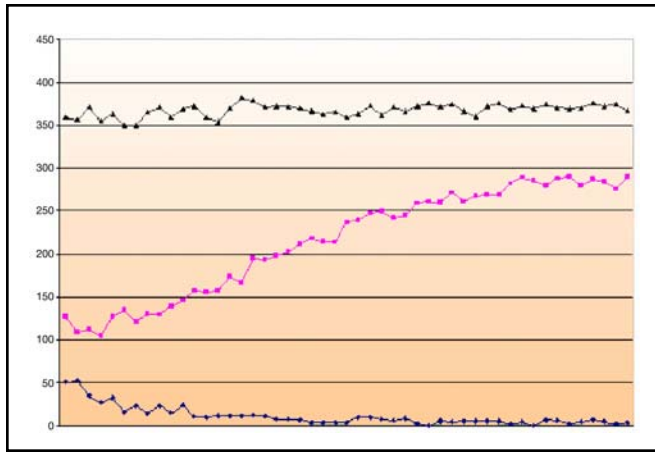


Figure 3. Results from the First Experiment

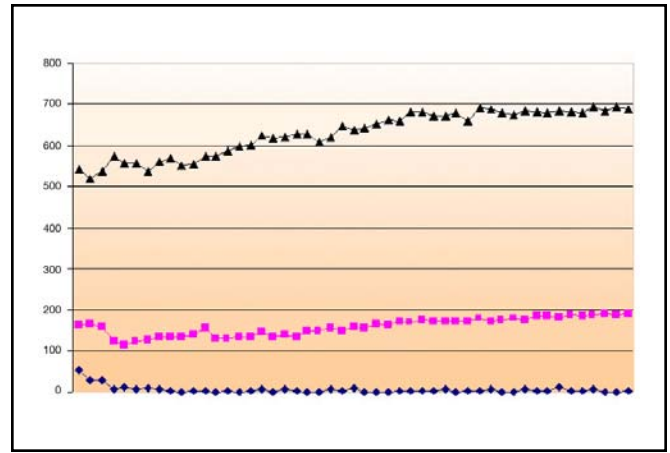


Figure 4. Results from the Second Experiment

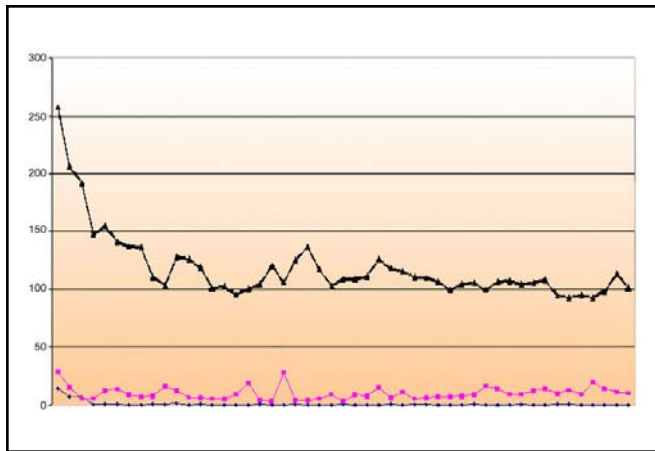


Figure 5. Results from the Third Experiment

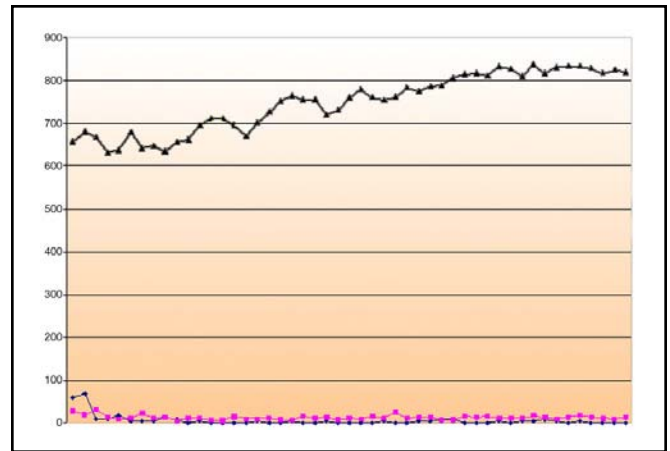


Figure 6. Results from the Fourth Experiment

with those offspring whose parents had the lowest 200 fitness values (the worst-class).

### Experimental Results

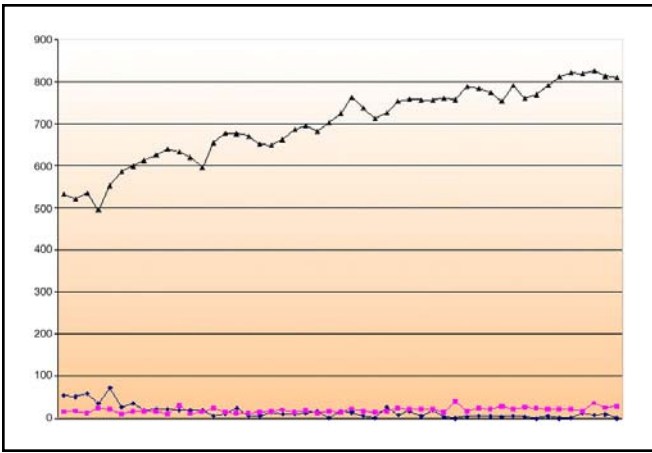
Figure 3 depicts the results from running the Genetic Program against equation 2. The  $x$ -axis represents the number of generations (1 through 50). The  $y$ -axis shows the average fitness values for the best-class, middle-class, and worst-class groups. The top line (the one in the 300-400 range) represents average fitness values of the offspring for the best-class parents. The middle line is the average fitness values of the offspring for the middle-class parent chromosomes. The bottom line shows the average fitness values of the offspring for the worst-class parents. Inspecting Fig. 3, it is clear that there is a distinction between best, middle, and worst-class groups. At no time do any of the group averages intersect. A  $t$ -test reveals these differences as statistically significant.

Figure 4 shows the results for the GP modeling equation 3. The  $x$ -axis represents the number of generations (1 through 50). The  $y$ -axis shows the average fitness values for the best-class, middle-class, and worst-class groups. The top line (the one in the 500-700 range) represents average fitness values of the offspring for the upper-class parents. The middle line rep-

resents the average fitness values of the offspring for the middle-class parent chromosomes. The bottom line shows the average fitness values of the offspring for the worst-class parents. Once again, it is clear that there is a distinction between best, middle, and worst-class groups. At no time do any of the group averages intersect. A  $t$ -test reveals the differences as statistically significant.

Figure 5 illustrates the differences between the three groups of chromosomes for the third experiment. The  $x$ -axis represents the number of generations (1 through 50). The  $y$ -axis shows the average fitness values for all three classes. The top line represents the best-class group. The middle line corresponds to the average fitness values of the offspring for the middle set of parent chromosomes. The bottom line shows the average fitness values of the offspring of the worst class parents. Once again, there is a distinct difference between the best-class and the other two groups. The fitness values of the middle group are generally higher than the fitness values of the offspring of the worst-class parents.  $T$ -tests indicate that the differences between these groups are significant in all cases.

Figure 6 shows the results for the fourth experiment which model equation 5. The average fitness values of the offspring of the best chromosomes are far superior than the other two



**Figure 7. Results from the Fifth Experiment**

groups. Compared to the previous experiments, these middle and worst class parents are not as distinguishable. The *t*-test analysis indicates that the differences between these groups are significant in all cases where the highest fitness values differ from the other groups.

Figure 7 depicts the results for the fifth experiment (equation 6). The top line shows average fitness values of the offspring of the set of the best 200 chromosomes. These offspring are much higher than the other (middle and worst class) lines. The *t*-tests indicate that the superiority of the best-class parents is statistically significant to each of the other classes. Middle-class and worst-class results are comparable.

### Discussion

In all five experiments, the offspring of the best-class parents are statistically significantly superior to the other two classes. As the complexity increases over the progression of experiments, the gulf widens between the best-class parents and the other two classes. It may be inferred that this gulf would be greater for more complex datasets. Clearly the lineage of the best class parents is superior to the lineage of the middle and worst class parents.

This utility of this observation may be exploited by reconsidering how chromosomes are selected for breeding. The traditional selection process uses either rank (as presented in this paper), roulette (assign greater probabilities to more fit chromosomes), or stochastic universal sampling (assign equal probabilities to all chromosomes). Based on the results from the five experiments, an alternative approach would be to initially breed all *n* chromosomes, sort by groups, divide into *k* groups, and use the best *n/k* chromosomes for breeding. These best-class chromosomes may propagate *k* times in order to maintain *n* chromosomes within the population. Essentially, this breeding process eliminates the middle and worst class parent chromosomes. Considering that never in the case for any generation for any of the five experiments did the middle or worst-case produce an offspring average superior to the best-class parents, this alternative approach is quite plausible.

### Conclusions

Five experiments were conducted that examined lineage patterns across generations in GP modeling. In all five experiments the best-class of parents (top 20 percent of the population) produced statistically significantly superior offspring compared to middle-class parents (chromosomes ranked in the 40 to 60 range) and worst-class parents (bottom 20 percent of the population). This research lays a foundation for developing a new method for selecting chromosomes for GP modeling where best-class chromosomes are allowed to breed multiple times while middle and worst-class chromosomes are discarded.

### Future Directions

Future research efforts may assume several directions. A set of experiments can be designed to demonstrate the viability of using focused chromosome selection. These experiments can assess whether GPs train faster and/or produce better models. This experimental process could assess whether the number of groups has an impact upon the GP performance.

It may be interesting to explore two or more generations back to determine whether the chromosomal relationship decays over subsequent generations. Exploring the degree of decay would help determine the impact of ancestors upon their descendants.

### Acknowledgments

Financial support of this work was provided by the Institute for Space Systems Operations (ISSO).

### References

- <sup>1</sup>P. J. Angeline, "Genetic Programming and Emergent Intelligence," in *Advances in Genetic Programming*. Ed. K. E. Kinneer, Jr. MIT Press, 1994. Chp. 4, 75-98.
- <sup>2</sup>P. J. Angeline, "Genetic Programming's Continued Evolution," in *Advances in Genetic Programming*, Vol. 2. Ed. K. E. Kinneer, Jr. MIT Press, 1996. Chp. 1, 1-20.
- <sup>3</sup>J. Koza, "Genetic Programming," in *Encyclopedia of Computer Science and Technology* Eds. A. Kent and J. G. Williams. N.Y.: Marcel Decker, Inc., 1997.
- <sup>4</sup>W. B. Langdon and W. Banzhaf, "Genetic Programming Bloat without Semantics," 6th Internat'l Conference on Parallel Problem Solving from Nature, Paris, France, 2000. 201-10.
- <sup>5</sup>N. F. McPhee and N. J. Hopper, "Analysis of Genetic Diversity through Population History," *Proc.*, Genetic and Evolutionary Computation Conference, July 1999.
- <sup>6</sup>E. Burke, S. Gustafson, G. Kendall, and N. Krasnogor, "Is Increased Diversity Beneficial in Genetic Programming: An Analysis of the Effects on Fitness," *Proc.*, Congress on Evolutionary Computation, Australia. IEEE Press, 2003. 1398-405.
- <sup>7</sup>J. J. Grefenstette, "Incorporating Problem Specific Knowledge into Genetic Algorithms," in *Genetic Algorithms and Simulated Annealing*. Ed. L. Davis. Morgan Kaufmann Publishers, Inc., 1987. Chp. 4, 42-60.
- <sup>8</sup>D. Whitley, "A Genetic Algorithm Tutorial," Department of Computer Science, Colorado State University, Technical Report CS-93-103, Nov. 10, 1993.

<sup>9</sup>D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, USA, 1989.

### **Publications**

- Boetticher, G. and K. Kaminsky. "Optimization of Genetically Engineerable Evolvable Programs (GEEP) for Improved Data Understanding," *International Space Systems Annual Report*, 2003.
- Boetticher, G., H. Al-Mubaid, and K. Frasier-Scott. "Automated Hybridization of Machine Learners for Recursive Spot Identification, Optimization, and Gel Matching of 2-Dimensional Gel Electrophoresis," *International Space Systems Annual Report*, 2003.

### **Presentations**

- Boetticher, G. "The GDB Cup: Applying 'Real World' Financial Data Mining in an Academic Setting," 8th World Multi-Conference on Systemics, Cybernetics and Informatics, Orlando, FL, July 18–21, 2004.
- Boetticher, G., W. Ding, C. Moen, and K.-B. Yue. "Using a Pre-Assessment Exam to Construct an Effective Concept-Based Grade Predicting Genetic Program," ACM SIGCSE, St. Louis, MO, Feb. 23–27, 2005.
- Davis, G. and G. Boetticher. "Associating Software Metrics with Software Coding Standards," Computer Applications Conference 2004, Houston, TX, May, 2004.
- Kaminsky, K. and G. Boetticher. "Better Software Defect Prediction Using Equalized Learning with Machine Learners," IASTED International Conference on Knowledge Sharing and Collaborative Engineering, St. Thomas, U.S. Virgin Islands, Nov. 22–24, 2004.
- Kaminsky, K. and G. Boetticher. "Building a Genetically Engineerable Evolvable Program (GEEP) Using Breadth-Based Explicit Knowledge for Predicting Software Defects," North American Fuzzy Information Processing Society (NAFIPS), Banff, Canada, June 27–30, 2004.
- Kaminsky, K. and G. Boetticher. "How to Predict More with Less, Defect Prediction Using Machine Learners in an Implicitly Data Starved Domain," 8th World Multi-Conference on Systemics, Cybernetics and Informatics, Orlando, FL, July 18–21, 2004.
- Kaminsky, K. and G. Boetticher. "Improving Software Quality Prediction through Equalized Learning Using Machine Learners," Computer Applications Conference 2004, Houston, TX, May, 2004.

### **Funding and Proposals**

- Boetticher, G. "Using Machine Learners To Predict Infant RSV Infections," National Heart Lung Blood Institute (NHLBI), Clinical Proteomics Programs, RFA-HL-04-019, National Institutes of Health, (involves UTMB), requested for 4 years, \$199,282 (*submitted*).
- Boetticher, G. "Mucosal Biomarkers of Viral Induced CAP, SEPSIS and CAP: Partnerships for Diagnostics Development," National Institute of Allergy and Infectious Diseases, National Institutes of Health, (involves UTMB), requested for 5 years, \$251,362 (*submitted*).