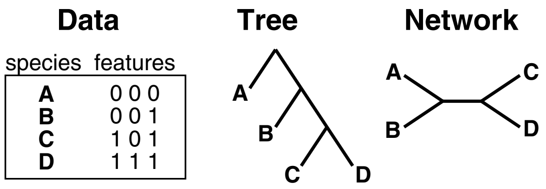
# Gaphyl: An Evolutionary Algorithms Approach for the Study of Natural Evolution

**Clare Bates Congdon**
Computer Science Department
Colby College
5846 Mayflower Hill Drive
Waterville, ME 04901
ccongdon@colby.edu

## Abstract

Gaphyl is an application of genetic algorithms (GA's) to phylogenetics, an approach used by biologists to investigate the evolutionary relationships among organisms. Typical phylogenetic software packages use heuristic search methods to navigate through a space of possible trees in an attempt to find the most plausible evolutionary hypotheses, as exhaustive search is not practical in this domain. Gaphyl substitutes an evolutionary search mechanism, with the result that on a complex problem from the literature (the major clades of the angiosperms), Gaphyl is able to find a more complete solution (more equally plausible hypotheses) in less time than the standard approach. Contributions of GA operators are investigated, as are some possibilities for hybrid systems.



| species | features |
|---|---|
| A | 0 0 0 |
| B | 0 0 1 |
| C | 1 0 1 |
| D | 1 1 1 |

Figure 1: A toy example data set, sample phylogeny, and sample network. In this example, there are four species and three features. The tree formed shows the hypothesis that species B is related to species A, gaining the third feature. Similarly, C and D are more closely related to B than to A, also acquiring new features.

## 1 INTRODUCTION

The human genome project and similar projects in biology have led to a wealth of data and the rapid growth of the emerging field of bioinformatics, a hybrid discipline between biology and computer science that uses the tools and techniques of computer science to help manage, visualize, and find patterns in this data. The work reported here is an application to biology, and indicates gains from using genetic algorithms (GA's) as the search mechanism for the task.

Phylogenetics [6] is a method widely used by biologists to reconstruct hypothesized evolutionary pathways followed by species currently or previously inhabiting the Earth. Given a dataset that contains a number of different species, each with a number of attribute-values, phylogenetics software constructs phylogenies, which are representations of the possible evolutionary relationships among the given species. A typical phylogeny is a tree structure: The species nearest the root of a tree can be viewed as the common ancestor, the leaves of a tree are the species, and subtrees are subsets of species that share a common ancestor. Each branching of a parent node into offspring represents a divergence in one or more attribute-values of the species within the two subtrees. In an alternate approach, sometimes called "unrooted trees" or "networks", the root of the tree is not assumed to be an ancestral species, although these hypotheses are often drawn as trees as a convenience. Unrooted trees represent hypothetical relationships between species, but do not attempt to model ancestral relationships.

An example phylogeny for a toy data set is shown in Figure 1. In this example, species A is the common ancestor in the tree, and B is the common ancestor of the subtree below A (assuming the tree is rooted). The relationships between species is also shown in the network representation, to better understand the unrooted tree.

Phylogenies are evaluated using metrics such as parsimony: A tree with fewer evolutionary changes is considered better than one with more evolutionary changes. The work reported here used Wagner parsimony. Wagner parsimony is straightforward to compute (requiring only a single pass through the tree) and incorporates few constraints on the evolutionary changes that will be considered. For example, some parsimony approaches require the assumption that species will only grow more complex via evolution — that features will be gained, but not lost in the process.

The typical phylogenetics approach uses a deterministic hillclimbing methodology to find a phylogeny for a given dataset, saving one or more "most parsimonious" trees as the result of the process. The most parsimonious trees are the ones with a minimum number of evolutionary changes connecting the species in the tree. Multiple "bests" correspond to equally plausible evolutionary hypotheses, and finding more of these competing hypotheses is an important part of the task. The tree-building approach adds each species into the tree in sequence, searching for the best place to add the new species. The search process is deterministic, but multiple trees may be found in the process of the search, and different trees may be found by running the algorithm with different random "jumbles" of the order of the species in the dataset.

This research is an investigation into the utility of using evolutionary algorithms on the problem of finding parsimonious phylogenies.

## 2 DESIGN DECISIONS

To hasten the development of our system, we used parts of two existing software packages. Phylip [5] is a phylogenetics system widely used by biologists. In particular, this system contains code for evaluating the parsimony of the phylogenies (as well as some helpful utilities for working with the trees). Using the Phylip source code rather than writing our own tree-evaluation modules also helps to ensure that our trees are properly comparable to the Phylip trees. Genesis [7] is a genetic algorithms (GA) package intended to aid the development and experimentation with variations on the GA. In particular, the basic mechanisms for managing populations of solutions and the modular design of the code facilitate implementing a GA for a specific problem. We named our new system Gaphyl, a reflection of the combination of GA and Phylip source code.

The research described here was conducted using published datasets available over the internet [4]. The first dataset used is the families of the superorder of Lamiiflorae dataset [1], consisting of 23 species and 29 attributes. This dataset was chosen as being large enough to be interesting, but small enough to be manageable. A second dataset, the major clades of the angiosperms [3], consisting of 49 species and 61 attributes, was used for further experimentation. These datasets were selected because the attributes are binary, which simplified the development of the system. As a preliminary step in evaluating the GA as a search mechanism for phylogenetics, "unknown" values for the attributes were replaced with 1's to make the data fully binary. This minor alteration to the data does impact the meaningfulness of the resulting phylogenies as evolutionary hypotheses, but does not affect the comparison of Gaphyl and Phylip as search mechanisms.

The typical GA approach to doing "crossover" with two parent solutions with a tree representation is to pick a subtree (an interior or root node) in both parents at random and then swap the subtrees to form the offspring solution. The typical mutation operator would select a point in the tree and mutate it to any one of the possible legal values (here, any one of the species). However, these approaches do not work with the phylogenies because each species must be represented in the tree exactly once.

Operators designed specifically for this task are described in the following sections and in more detail in [2].

### 2.1 CROSSOVER OPERATOR

The needs for our crossover operator bear some similarity to traveling salesperson problems (TSP's), where each city is to be visited exactly once on a tour. There are several approaches in the literature for working on this type of problem with a GA, however, the TSP naturally calls for a string representation, not a tree. In designing our own operator, we studied TSP approaches for inspiration, but ultimately devised our own. We wanted our operator to attempt to preserve some of the species relationships from the parents. In other words, a given tree contains species in a particular relationship to each other, and we would like to retain a large degree of this structure via the crossover process.

Our crossover operator proceeds as follows:

1. Choose a species at random from one of the parent trees. Select a subtree at random that includes this node, excluding the subtree that is only the leaf node and the subtree that is the entire tree. (The exclusions prevent crossovers where no information is gained from the operation.)
2. In the second parent tree, find the smallest subtree containing all the species from the first parent's subtree.

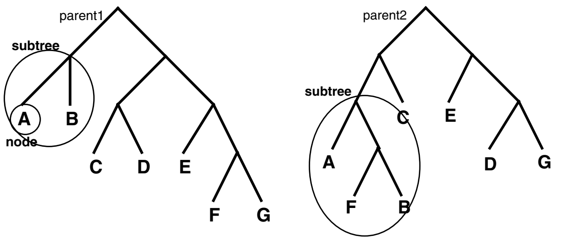


Figure 2: Two example parent trees for a phylogenetics problem with seven species. A subtree for crossover has been identified for each tree.

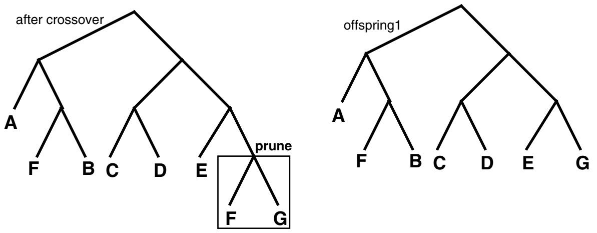


Figure 3: At the left, the offspring initially formed by replacing the subtree from parent1 with the subtree from parent2; on the right, the offspring tree has been pruned to remove the duplicate species F.

3. To form an offspring tree, replace the subtree from the first parent with the subtree from the second parent. The offspring must then be pruned (from the "older" branches) to remove any duplicate species.
4. Repeat the process using the other parent as the starting point, so that this process results in two offspring trees from two parent trees.

This process results in offspring trees that retain some of the species relationships from the two parents, and combine them in new ways.

An example crossover is illustrated in Figures 2 and 3. (Note that in the phylogenies, swapping the left and right children does not affect the meaning of the phylogeny.)

## 2.2 CANONICAL FORM

Trees are put into a canonical form when saving the best trees found in each generation, to ensure that no equivalent trees are saved among the best ones. Canonical form is illustrated in Figure 4.

## 2.3 MUTATION OPERATORS

One of our mutation operators selects two leaf nodes (species) at random, and swaps their positions in the tree. This operator allows the GA to investigate slight variations on a parent tree.

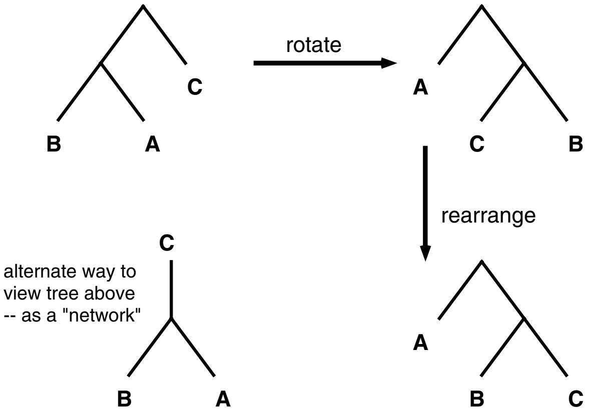


Figure 4: An illustration of putting a tree into canonical form. The tree starts as in the top left; an alternate representation of the tree as a "network" is shown at the bottom left. First, the tree is rotated, so that the first species in the dataset is an offspring of the root. Second, subtrees are rearranged so that smaller trees are on the left and alphabetically lower species are on the left.

A second mutation operator picks a random subtree and a random species within the subtree. The subtree is rotated to have the species as the left child of the root and reconnected to the parent. The idea behind this operator is that within a subtree, the species might be connected to each other in a promising manner, but not well connected to the rest of the tree.

## 2.4 IMMIGRATION

The population is subdivided into a specified number of subpopulations which, in most generations, are distinct from each other (crossovers happen only within a given subpopulation). After a number of generations have passed, each population migrates a number of its individuals into other populations; each emigrant determines at random which population it will move to and which tree within that population it will uproot. The uprooted tree replaces the emigrant in the emigrant's original population. The number of populations, the number of generations to pass between migrations, and the number of individuals from each population to migrate at each migration event are determined by parameters to the system. Immigration was added due to problems with premature convergence identified in early stages of development.

# 3 EXPERIMENTAL RESULTS

Recall that both Gaphyl and Phylip have a stochastic component, which means that evaluating each system requires doing a number of runs. In Phylip, each distinct run first "jumbles" the species list into a different random order. In Gaphyl, there are many different effects of random number generation: the construction of the initial population, parent selection, and the selection of crossover and mutation points. For both systems, a number of different runs must be done to evaluate the approach.

### 3.1 COMPARISON OF GAPHYL AND PHYLIP

1. With the Lamiiflorae data set, the performance of Gaphyl and Phylip is comparable. Phylip is more expedient in finding a single tree with the best parsimony (72), but both Gaphyl and Phylip find 45 most parsimonious phylogenies in about twenty minutes of run time.
2. With the angiosperm dataset, a similar pattern emerges: Phylip is able to find one tree with the best fitness (279) quite quickly, while Gaphyl needs more run time to first discover a tree of fitness 279. However, in a comparable amount of runtime, Gaphyl is able to find 250 different most parsimonious trees of length 279 (approximately 24 hours of runtime). Phylip runs for comparable periods of time have not found more than 75 distinct trees with a parsimony of 279, and runs of nearly 3 days have not turned up more than 95 distinct trees. Furthermore, the trees found by Phylip are a proper subset of the trees found by Gaphyl.

In other words, Gaphyl is more successful than Phylip in finding more trees (more equally plausible evolutionary hypotheses) in the same time period. This represents a more complete solution to the problem.

The Lamiiflorae task is considerably easier to solve than the angiosperm task. Example parameter settings are a single population of 500, 500 generations, 50% elitism (the 250 best trees are preserved into the next generation), 100% crossover, 10% first mutation, and 100% second mutation. Empirically, it appears that 72 is the best possible parsimony for this dataset, and that there are not more than 45 different trees of length 72.

The angiosperm task seems to benefit from immigration in order for Gaphyl to find the best known trees (fitness 279). Successful parameter settings are 5 populations, population size of 500 (in each subpopulation), 2000 generations, immigration of 5% (25 trees) after every 500 generations, 50% elitism (the 250 best trees are preserved into the next generation), 100% crossover, 10% first mutation, and 100% second mutation. (Immigration does not happen following the final generation.) We have not yet done enough runs with either Phylip or Gaphyl to estimate the maximum number of trees at this fitness, nor a more concise estimate of how long Phylip would have to run to find 250 distinct trees, nor whether 279 is even the best possible parsimony for this dataset.

### 3.2 BIG PICTURE: THE ROLE OF THE GA IN THIS TASK

In constructing Gaphyl, we used the code from Phylip's evaluation metric, but the search mechanisms are those of the GA described in this section. In other words, we are investigating the use of the GA as an alternate search method for this already established task. There is an immediate gain to our approach: Our search for trees increases in complexity with the number of species in the dataset. The number of attributes, however, does not affect the search. Conversely, the complexity of the search in Phylip increases relative to the number of attributes as well as the number of species in the dataset. Biologists frequently run phylogeny software for weeks at a time, so a savings in speed has a measurable impact.

Both systems are far from optimized, so strong conclusions cannot be drawn from runtime alone. However, the pattern that is emerging is that as the problems get more complex, Gaphyl is able to find a more complete set of trees with less work than what Phylip is able to find. The work done to date illustrates that Gaphyl is a promising approach for phylogenetics work, as Gaphyl finds a wider variety of trees on this problem than Phylip does. This further suggests that Gaphyl may be able to find solutions better than those Phylip is able to find on datasets with a larger number of species and attributes, because it appears to be searching more successful regions of the search space.

While it is true that one cannot compare software on runtime alone, recall that Gaphyl was constructed from existing systems, neither one of which was optimized for speed. In particular, Genesis was designed to simplify GA experimentation and modifications (much like the project here). It is possible to make some comparisons of operations done by the two systems in their search, but these are apples and oranges, since the work done to get from one tree to the next varies between the systems. In Phylip, each jumble corresponds to a hillclimbing search, which (with the Angiosperms dataset) investigates on the order of 10,000 trees for each random ordering of the species list, and 40,000 jumbles in the 24 hours, or on the order of 400 million trees. In Gaphyl, 10 experiments (using different seeds to the random number generator) with 2500 total trees and 2000 generations investigates on the order of 50 million trees in 24 hours, although the number should be halved due to the 50% elitism.

### 3.3 CONTRIBUTION OF OPERATORS

To evaluate the contributions of the GA operators to the search, additional runs were done with the first data set (and a single population). Empirically, crossover and the second mutation operator had been found to be the largest contributors to successful search, so attention was focused on the contributions of these operators.

In the first set of experiments, the first mutation rate was set to be 0%. First, the crossover rate was varied from 0% to 100% at increments of 10% while the second mutation rate was held constant at 100%. Second, the second mutation rate was varied from 0% to 100% at increments of 10% while the crossover rate was held constant at 100%. 20 experiments were run at each parameter setting; 500 generations were run.

Figure 5 illustrates the effects of varying the crossover rate (solid line) and second mutation rate (dashed line) on the average number of generations taken to find at least one tree of the known best fitness (72). Experiments that did not discover a tree of fitness 72 are averaged in as taking 500 generations. For example, 0% crossover was unable to find any trees of the best fitness in all 20 experiments, and so its average is 500 generations. This first experiment illustrates that in general, higher crossover rates are better. There is not a clear preference, however, for high rates of the second form of mutation. To look at this operator more closely, the final populations of the 20 experiments were looked at to determine how many of the best trees were found in each run.

Figure 6 illustrates the effects of varying the crossover rate (solid line) and second mutation rate (dashed line) on the average number of best trees found. Experiments that did not discover a tree of fitness 72 are averaged in as finding 0 trees. For example, 0% crossover was unable to find any trees of the best fitness in all 20 experiments, and so its average is 0 of the best trees. As Figure 6 illustrates, runs with a higher second mutation rate tend to find more of the best trees than runs with a lower second mutation rate.

The impact of the first mutation operator had seemed to be low based on empirical evidence. So another set of experiments was done to assess the contribution of this operator. In both, the crossover rate was set at 100%; in one, the second mutation rate was set at 0% and in the other, the second mutation rate was set at 100%. The results of this experiment clearly indicate that higher rates of this form of mutation are not beneficial. Furthermore, this operator is not clearly contributing to the search. The results are illustrated in Figure 7.

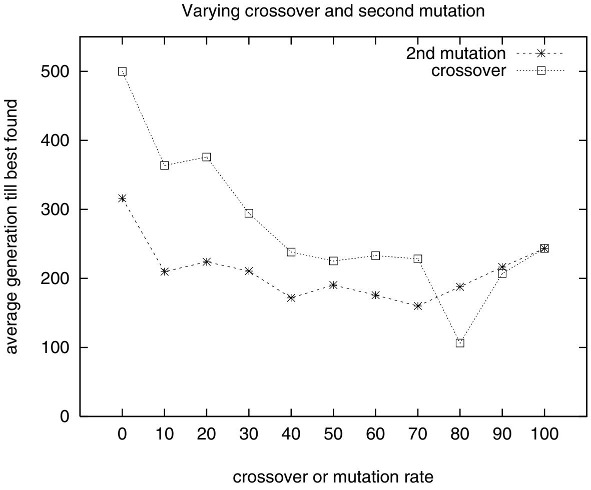


Figure 5: The effect of varying crossover rate while holding second mutation constant and of varying the second mutation rate while holding the crossover rate constant. The average generation at which the best fitness (72) was found is illustrated.

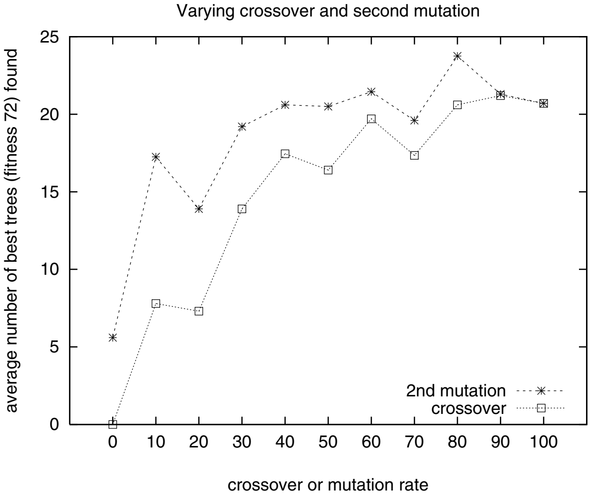


Figure 6: The effects of varying crossover rate while holding second mutation constant and of varying the second mutation rate while holding the crossover rate constant. The average number of best trees (45 max) found by each parameter setting is illustrated.

In the final set of experiments, the first experiments of varying crossover rate while holding second mutation rate constant and vice versa were repeated, but this time with a first mutation rate of 10%. The results are illustrated in Figure 8.

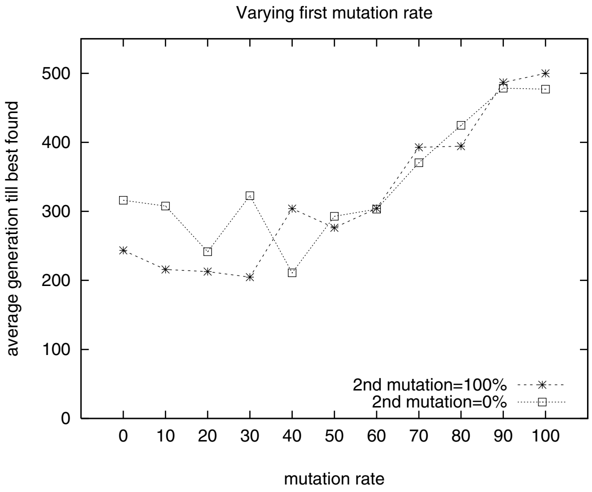


Figure 7: The effect of varying the first mutation rate while holding crossover and second mutation constant. The crossover rate is 100% for both graphs; second mutation rates of 100% and 0% are shown. The average generation at which the best fitness (72) was found is illustrated.

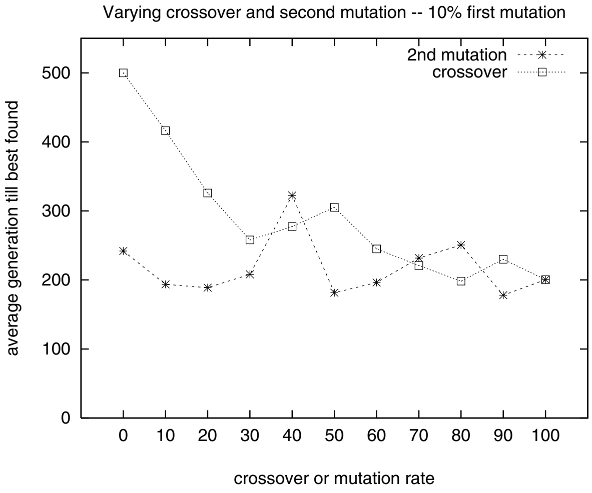


Figure 8: The effect of varying crossover rate while holding second mutation constant and of varying the second mutation rate while holding the crossover rate constant, this time with a first mutation rate of 10%. The average generation at which the best fitness (72) was found is illustrated.

### 3.4 CONTRIBUTION OF OTHER PARAMETERS

Population sizes for each experiment

| Gens | Number of Populations | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 4 | 8 | 16 |
| 1600 | 1024 | 512 | 256 | 128 | 64 |
| 800 | 2048 | 1024 | 512 | 256 | 128 |
| 400 | 4096 | 2048 | 1024 | 512 | 256 |

Table 1: The population size for each experiment described in Section 3.4. When there are multiple populations, the number shown refers to the number of trees in each distinct population.

An additional set of experiments was designed to assess tradeoffs in terms of putting a fixed number of trees in one population or distributing them across a number of populations and tradeoffs between having larger population sizes or doing more generations, for a fixed number of evaluations in all cases. These experiments were done using the angiosperms dataset.

The base case may be thought of as 1 population of 1024 individuals, and 1600 generations. Then, along one dimension, the population is divided across 2, 4, 8, and 16 populations, a total of five variations. Along the other dimension the number of generations is halved as the population size is doubled, for a total of three variations. This creates an array of 15 parameter settings, illustrated in Table 1. The horizontal axis shows the number of populations, the vertical axis shows the number of generations, and each interior cell shows the population size.

Twenty experiments, with different seeds to the random number generator, were done for each setting. When multiple populations are used, five percent of the population immigrates after 25%, 50%, and 75% of the generations have completed.

The results of these experiments, illustrated in Table 2, show the best results with 2 populations of 1024 trees run for 800 generations, with a total of 7 out of the 20 runs finding trees of the best known fitness of 279. In general, it appears that two populations are better than one, but that there might not be great gains from more than two populations. Further, it appears that the system benefits from a balance between a large population size and a large number of generations.

### 3.5 EXPLORATION OF HYBRID POSSIBILITIES

We have noted that Phylip is relatively quick to find at least one of the best solutions, but that over a span of time, does not find as many of the best solutions as Gaphyl does. Therefore, it seems that investigating the possibility of a hybrid system would be beneficial. The hybrid variation explored here is to use Phylip runs to seed the initial population of the GA run.

In these experiments, the point of comparison is the

Number of runs that found a "best"

| Gens | Number of Populations | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 4 | 8 | 16 | sum |
| 1600 | 1 | 2 | 1 | 2 | 1 | 7 |
| 800 | 5 | 7 | 2 | 5 | 2 | 21 |
| 400 | 3 | 3 | 4 | 0 | 0 | 10 |
| sum | 9 | 12 | 7 | 7 | 3 | |

Average fitness of final populations

| Gens | Number of Populations | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 4 | 8 | 16 |
| 1600 | 281.70 | 281.55 | 281.10 | 280.85 | 280.95 |
| 800 | 280.60 | 279.95 | 280.25 | 279.95 | 280.15 |
| 400 | 280.45 | 280.15 | 280.45 | 281.15 | 281.95 |

Table 2: The number of runs that found the best solution and the average best solution found across 20 runs, varying the number of populations and number of generations, with a constant 1024 trees (split across the specified number of populations).

starting point for the system. Four variations were explored, using the angiosperms dataset:

1. Starting with an entirely random initial population.
2. Starting with an initial population comprised of a random selection of trees found by running one Phylip jumble.
3. Starting with an initial population comprised of half Phylip trees from one jumble and half random trees.
4. Starting with an initial population comprised of 20 Phylip trees, one of the best from each of 20 different jumbles, and the remainder random trees.

25 experiments were run for each variation. One population was used, so as not to confound the effects of multiple populations. The population size was 2000 trees, run for 1000 generations. Other parameters are as reported previously.

Of these runs, the 4th variation fared the best, finding at least one tree with the 279 fitness in 14 of the 25 runs. Secondly, the first variation found at least one tree with 279 fitness in 5 of the 25 runs. The second and third variations did not find any trees of 279 fitness in the 25 runs. Trajectories of average fitnesses across all runs are shown in Figure 9.

These experiments suggest that while seeding from Phylip runs may help the progress of the GA, the initial seeds must be sufficiently diverse for this "jump start" to be helpful. It appears that choosing the seed trees from a single Phylip jumble is comparable to starting the GA with a population that has already converged. (Note: This experiment was repeated with five distinct Phylip jumbles, always with similar results.)

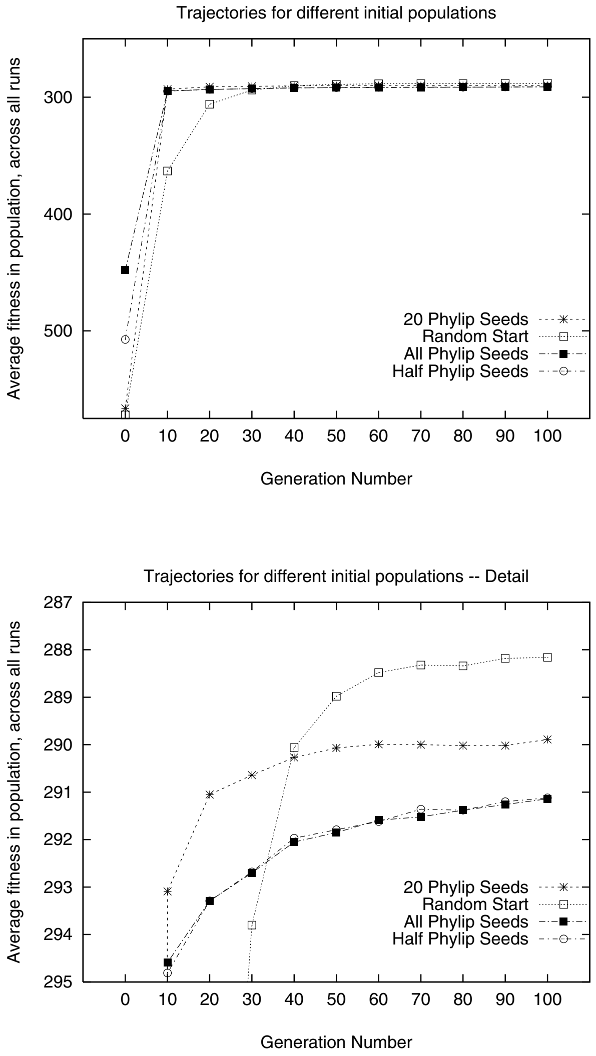


Figure 9: Trajectories for the four experiments with seeding the initial population. The second graph shows more detail of the lower fitness values.

## 4 CONCLUSIONS AND FUTURE WORK

The GA search process as implemented in Gaphyl represents a gain for phylogenetics in its ability to find more equally plausible trees than Phylip in the same runtime. Furthermore, as the datasets get larger in the number of species and attributes, the effectiveness of Gaphyl over Phylip appears to increase. One possible facet of this success is that the Gaphyl search process is independent of the number of attributes (and attribute-values); the complexity of the search varies with the number of species (which determines the number of leaf nodes in the tree). Phylip uses attribute information in its search process.

The first mutation operator is perhaps the "obvious" form of mutation to implement for this problem, and yet, its use (at high levels) appears to detract from the success of the search. While multiple populations appear to help the system avoid premature convergence, too many populations are not helpful.

The creation of a hybrid system that uses Phylip's relatively fast but limited search strategy to seed the initial population is a promising approach, as long as care is taken that the seeds are diverse.

There is obviously a wealth of possible extensions to the work reported here. First, more extensive evaluations of the capabilities of the two systems must be done on the angiosperms data set, including an estimate of the maximum number of trees of fitness 279 (and, indeed, whether 279 is the most parsimonious tree possible). This would entail more extensive runs with both approaches.

Second, more work must be done with a wider range of datasets to evaluate whether Gaphyl is consistently able to find a broader variety of trees than Phylip, and perhaps able to find trees better than Phylip is able to find.

Third, Gaphyl should be extended to work with non-binary attributes. This is particularly important in that phylogenetic trees are increasingly used by biologists primarily with the A, C, G, T markers of genetic data.

Finally, we need to compare the work reported here to other projects that use GA approaches with different forms of phylogenetics, including [8] and [9]. Both of these projects use maximum likelihood for constructing and evaluating the phylogenies. The maximum likelihood approach (which is known as a "distance-based method") is not directly comparable to the Wagner parsimony approach (which is known as a "maximum parsimony" approach).

## Acknowledgments

I would like to thank Emily F. Greenfest, who worked with me in the initial design and implementation of this project. Thanks also to Judy L. Stone and Randall Downer for sharing their knowledge of phylogenetic theory and software. Thanks to Randolph M. Jones, Joshua R. Ladieu, and the anonymous reviewers for comments on previous versions of this paper. The Colby College Natural Science Grant Program provided travel and equipment support for the author.

## References

[1] L. An-Ming. A preliminary cladistic study of the families of the superorder lamiiflorae. *Biol. J. Linn. Soc.*, 103:39–57, 1990.

[2] C. B. Congdon. Gaphyl: A genetic algorithms approach to cladistics. In L. De. Raedt and A. Siebes, editors, *Principles of Data Mining and Knowledge Dicovery (PKDD 2001)*, Lecture notes in artificial intelligence 2168, pages 67–78, New York, 2001. Springer.

[3] R. Dahlgren and K. Bremer. Major clades of the angiosperms. *Cladistics*, 1:349–368, 1985.

[4] M. J. Donaghue. Treebase: A database of phylogenetic knowledge. web-based data repository, 2000. `http://phylogeny.harvard.edu/treebase`.

[5] J. Felsenstein. Phylip source code and documentation, 1995. Available via the web at `http://evolution.genetics.washington.edu/phylip.html`.

[6] P. L. Forey, C. J. Humphries, I. L. Kitching, R. W. Scotland, D. J. Siebert, and D. M. Williams. *Cladistics: A Practical Course in Systematics*. Number 10 in The Systematics Association Series. Clarendon Press, Oxford, 1993.

[7] J. J. Grefenstette. A user's guide to GENESIS. Technical report, Navy Center for Applied Research in AI, Washington, DC, 1987. Source code updated 1990; available via the web at `http://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/genetic/ga/systems/genesis/`.

[8] P. O. Lewis. A genetic algorithm for maximum-likelihood phylogeny inference using nucleotide sequence data. *Mol. Biol. Evol.*, 15(3):277–283, 1998.

[9] H. Matsuda. Protein phylogenetic inference using maximum likelihood with a genetic algorithm. In L. Hunter and T. E. Klein, editors, *Pacific Symposium on Biocomputing '96*, pages 512–523. World Scientific, London, 1996.