# A Self-Adaptive Mate Selection Model for Genetic Programming

**Rodney Fry, Stephen L. Smith and Andy M. Tyrrell**
Department of Electronics.
The University of York
Heslington, York YO10 5DD, UK
{sls5,amt}@ohm.york.ac.uk

**Abstract- This paper documents new extensions to the selection model in genetic programming, designed to be analogous to the more complex behaviour of selection in natural evolution. Specifically, a negative assortative mating scheme is presented in conjunction with a model of psychological evolution, allowing the mating strategy to change throughout the evolutionary process. Results show that self-adaptive mate selection accelerates evolution for several well known test problems.**

## 1 Introduction

The theory of natural selection is now well-accepted in the scientific community. If there is variation amongst members of the population, there will be a differential rate of survival between those individuals. Provided that the variation is directly related to an underlying genetic description, rather than being caused by environmental forces, then in each successive generation unwanted characteristics will be lost and favourable characteristics will be retained, given that the genetic description has some degree of heritability. The process of natural selection changes the genetic makeup of the population.

Newer theories are now becoming more widely accepted. Prominent amongst them is the change which cognitive sexual selection brings about. Several new ideas must be considered. Both male and female must choose to mate with one another: random mating no longer occurs. Given that mental traits are heritable in exactly the same way as physical traits, the population can evolve to perceive the beneficial characteristics in potential mates to be attractive qualities. Mate choice is a cognitive process which is subject to selective pressure, but which also exerts its own selective pressure. Sexual selection has been shown to accelerate the process of adaptation in animal and plant populations.

Evolutionary theory is still an ongoing field of research. Genetic programming, therefore, finds itself in an advantageous position since it is based upon a model of evolution. Just as Darwin's theories of evolution by natural selection have now been qualified using more modern knowledge of population genetics, so Darwin's observations of sexual selection are now being considered with more than mere passing interest. The field of evolutionary psychology is a new and promising area of research. Just as models of evolution have led to the emergence of technologies such as genetic programming, it is obvious that those same technologies can now adopt new additions to the model – specifically, the extension of the model of natural selection to include the effect of mate choice.

This work follows from the notion that genetic programming is based around a basic model of the process of evolution by natural selection. By using more detailed model, the performance of the system may be enhanced. More specifically, it is asserted that mate choice plays a significant role in the evolution of sexually reproducing animals and plants.

The psychological evolution of sexual selection provides new aspects to include in the model of evolution. Following from these assertations, it is hypothesised that extending the model of selection used in genetic programming to incorporate mate choice will enhance the performance of the system.

## 2 Biological Inspiration

The importance of mate choice in computational evolutionary models was not identified very long ago [12]. Indeed, it is only fairly recent that the opinion of the biological research community has shifted towards the notion of sexual selection being of high evolutionary importance [13,4]. Just as Darwin was not aware of the particulate nature of inheritance, evolutionary theorists, until recently, were not aware of the magnitude of the effect of sexual selection. The models of evolution used in evolutionary computation have demonstrated where our knowledge of the process of evolution was lacking. Todd and Miller [17] used a simple model of sexual selection to show how diversity is maintained, and how speciation and peak-hopping take place in natural systems.

The attractiveness of a sexually-reproducing individual determines how reproductively successful it will be. For example, flowering plants which are not fragrant and brightly coloured will not attract the insect pollinators required for reproduction. The fitness of the individual, therefore, incorporates a measure of attractiveness to possible mates. One can envisage a situation whereby a group of individuals of a lower survival rate are considered to be of high attractiveness, simply because they exhibit the correct phenotypic qualities. This concept can result in advantageous peak-hopping – something

0-7803-9363-5/05/$20.00 ©2005 IEEE.

which would be far less probable were it not for sexual selection.

The evolution of mate preference and the evolution of preferential features can be viewed as a feedback loop. If a particular phenotypic feature causes a group of individuals to have a higher rate of survival, it follows that a cognitive process within animals of the same species should evolve which makes that feature more attractive to them. Therefore, individuals which exhibit this feature more prominently will have a higher rate of reproduction. However, it should be noted that variations of the selection model exist; Zahavi considers the peacock's tail be a physical handicap, which the genetic fitness of the male has to overcome in order to attract a mate and produce superior offspring [16].

## 3 Previous Work

Due to the evolutionary theories underpinning sexual selection being relatively new, not much research has been done in the field of mate selection in evolutionary computation, and less so in genetic programming. Recent studies, however, have shown mate selection to be beneficial in genetic algorithms (GAs) [10]. It is not a great conceptual jump to assume that the same would be true of genetic programming. Previous work in mate selection schemes can be categorised in two distinct ways: as using computation- or memory-based mating. This categorization arises from the way which the similarity between two individuals is evaluated. There exists in nature a parallel to this categorisation. Choosing mates based on ancestral similarity or dissimilarity, inbreeding and outbreeding respectively, is an analogue of memory-based mating. Choosing mates based on phenotypic similarity or dissimilarity, positive and negative assortative mating respectively, is an analogue of computation-based mating. The work presented in this paper concerns the latter case.

The process of computation-based mating involves preferentially choosing mates based on how similar the two solutions are. In nature, an estimation is made, based on phenotype, as to the compatibility of the genetic makeup of an individual. In evolutionary algorithms, the same comparison can be made directly. The Hamming distance neatly quantifies how similar two fixed-length binary strings are. The earliest study with regard to using distance information in selection is that of incest prevention [6]. Individuals whose Hamming distance is small are regarded as being related and prevented from mating. Somewhat misleadingly, this scheme is computation based since incestuous matings are not identified by the use of ancestry information.

In a similar vein, and also for GAs, there exists a selection scheme whereby mates are selected that have the maximum Hamming distance [7,8]. The results show that, for a selection scheme based on dissimilarity between mates, the performance of the evolutionary algorithm may be enhanced.

Ratford et al. [14,15] also founded a selection scheme for GAs where mates are chosen for individuals based on the Hamming distance between the solutions. They define, through the use of a "seduction function", an optimal distance at which matings should occur. The parameter which defines the optimal distance is set before the run begins, although rudimentary provision for the optimal distance parameter to change throughout the run is included: the distance changes by a small step every generation until a predefined limit is reached.

A recent study into the importance of diversity measures in genetic programming [3] has introduced a number of key points. Using phenotypic entropy and edit distance as diversity measurements, a correlation between fitness and both phenotypic and structural diversity was observed. Populations were found to able to be structurally similar while maintaining high behavioural diversity. Although better performance is seen with higher levels of diversity, high diversity does necessarily cause better performance. The results acquired while testing the mate selection schemes certainly show an increase in performance. Equally, the results also show an increase in diversity, coupled with a decrease in the variance of diversity. Populations are more diverse more of the time, and this gives the system more of a opportunity to generate suitable solutions. However, the results do not prove a causal link between the observed increase in diversity and success rate.

## 4 Implementation

When implementing a GA or evolutionary strategy (ES) system, one expects to have a fairly small population, and for the number of generations required to find a suitable solution to be quite large. The implementation of GP, however, is different. Normally, a GP system has a larger population, but requires fewer generations to discover a suitable solution. A GP system, as a result of the variability of solution size, can solve a far greater number of problems than its fixed-length counterparts. The solutions are far larger and the operations required to build each new generation are more complicated. More must be done by the selection and recombination mechanisms enabling new selection schemes developed here to be evaluated effectively. Additionally, using a GP representation allows the tree-to-tree edit distance to be investigated. As such, the processing power and memory requirements of a GP system are far greater than a GA or ES system.

Using the two similarity-based mate selection schemes described in this paper undoubtedly require additional computer resources. However, the additional processor time used to extend the selection model, in accordance with observations of how natural evolution works, is justified by in the increase in the probability of success it provides.

The motivation behind this work is to produce an abstract similarity based mate selection scheme that can be applied to any optimisation scheme which adapts a population by recombining solutions. Since all evolutionary computations are based around a model of evolution of varying complexity, the selection scheme can simply be 'plugged in' to any evolving system to increase its performance. The work described is an extension of the model of selection, and is, therefore, an implicitly transferable concept, rather than a representation-specific formulation.

### 4.1 Modified Tournament Selection

The chance of a mating occurring is always related to fitness, since the greater chance an individual has of survival, the greater chance the individual has of reproducing. However, when an individual considers whether to mate with another specific individual, there is an additional underlying force which aids the decision. There is a tendency for animals to find genetically diverse individuals more attractive and a tendency for them to find relations less attractive. The probability of a mating occurring, therefore, is partly due to fitness and partly due to how genetically diverse or unrelated the two potential mates are.

The genetic diversity between, or relationship between, the two individuals may be generically termed their 'similarity'. For example, the less similar the two individuals are, the more likely they are to mate. A simple model of this observation would, therefore, require a measure of similarity between two individuals.

$$f'_{ij} = f_i(1 - s_{ij}) \qquad (1)$$

Equation 1 describes a model of the relationship between the standardised fitness of individual $i, f_i$, and the apparent fitness of individual $i$ from the perspective of individual $j$, $f'_{ij}$, with respect to the similarity between individuals $i$ and $j$, $s_{ij}$. The similarity is measured such that it is in the range $0 \leq s_{ij} \leq 1$. Therefore, for individuals that are completely dissimilar, the apparent fitness remains the same as the standardised fitness. However, for individuals that are identical, the apparent fitness drops to zero.

To integrate the apparent fitness metric into a standard evolutionary framework, the tournament selection mechanism must be modified slightly. The first of the two individuals which will undergo crossover is selected via a standard tournament: the probability of selection is based purely on fitness. Another tournament of identical size is initiated, except that the second individual is selected using the apparent fitness metric. The outcome is that the first individual chooses a mate for itself based on a combination of fitness and dissimilarity.

The rationale behind the modification of the tournament selection is simple. In order for the algorithm to promote genetic diversity within the population, individuals must be selected depending on how different they are from the fittest individuals. Similar to fitness sharing, where individuals occupying different niches enjoy a fitness payoff, individuals that occupy the same region of the problem space have a lower apparent fitness. This effect is achieved by selecting the first parent based solely on fitness. If the first parent is more likely to be of high fitness, then, due to the modified tournament selection scheme, the second parent will be more likely to be not just of high fitness, but of different genetic constitution.

The modified tournament selection scheme is similar to fitness sharing. They both alter the fitness function in such a way that helps to maintain diversity in the population, and to increase the success rate of the evolutionary algorithm. There are, however, a number of differences between the two. Explicit fitness sharing alters the fitness function on a population-wide level depending on how the individuals are distributed around the fitness landscape. The apparent fitness, however, is calculated at an individual level. The fitness function is altered with reference to how suitable two potential mates might be. The reason for the differences between the two schemes is that they are modelled on different biological occurrences: fitness sharing is a model of environmental niching and the modified tournament selection is a model of mate preference in sexual selection. The similarities between the two arise from their common purpose: the maintenance of population diversity.

Given that the probability of a mating taking place is related to genetic diversity and to their consanguinity, there are two ways of quantifying the similarity, $s_{ij}$ : either based directly on the genetic similarity between the two potential mates, or based on common ancestry. The work presented here considers the former.

### 4.2 Edit Distance Similarity

In order to quantify the genetic similarity between two parse trees, one must measure the edit distance between them. The edit distance calculation, however, is notoriously processor intensive. However, inspired by Ekárt and Németh [5] and the description of their implementation of a more efficient edit distance calculation, the following genetic similarity metric was formulated.

Differences between each of the nodes of the parse tree may be counted by simply recursively traversing down the two trees simultaneously, counting the number of nodes compared, $n_{ij}$ , and the edit operations necessary, $d_{ij}$ . The number of edit operations is incremented when the functions between the two nodes are different or when a node must be inserted or deleted.

Figure 1 illustrates this process using two fairly similar parse trees as an example. In this case, the number of nodes compared, $n_{ij}$ = 12, and the number of edit operations necessary to transform one tree into the other, the edit distance, $d_{ij}$ = 5. The number of edit operations can be used to calculate how similar two parse trees are. In this study, two different methods of calculating the similarity are defined: the relative similarity and the absolute similarity.

### 4.3 Relative Similarity

Counting the number of nodes compared, $n_{ij}$, provides a useful quantity when calculating the similarity between the two trees. The value of $n_{ij}$ represents the maximum number of edit operations that could have occurred. For example, if two parse trees are entirely different and require an edit at every node, then the number of edit operations required would be equal to the number of nodes compared, therefore $n_{ij} = d_{ij}$. Conversely, if the two parse trees are identical, then the number of edit operations is zero, no matter how many nodes had to be compared. Therefore, the ratio of edit operations to nodes compared represents the relative edit distance between the two parse trees, $r_{ij}$.

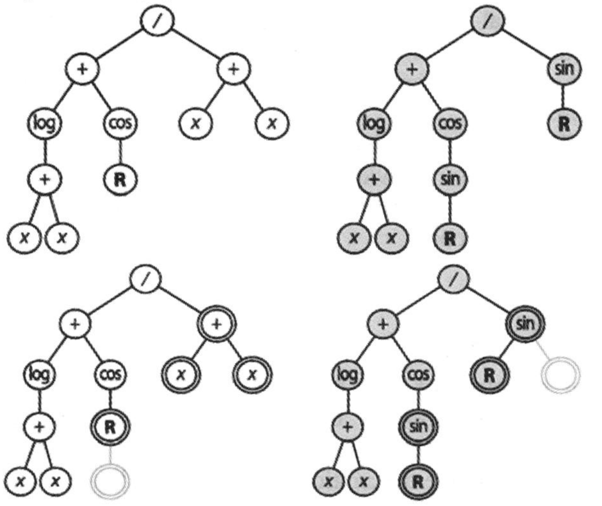


**Figure 1. Diagrammatic representation of the calculation of the edit distance between two parse trees.**

The relative similarity may be calculated as described in Equation 3.

$$r_{ij} = \frac{d_{ij}}{n_{ij}} \tag{2}$$

$$s_{ij} = 1 - r_{ij} \tag{3}$$

The relative similarity is a measure of how different the two trees could have been; the maximum difference changes, depending on which trees are being measured.

### 4.4 Absolute Similarity

In order to measure the absolute difference between two trees, the combined tree size measure must be discarded. In the absence of a measure of the limit of the difference between the two trees, the difference between the two trees must be compared against the maximum distance possible for any tree. The greatest number of edit operations required would be the greatest number of nodes a parse tree of a given depth limit, $D_p$, could have. This represents a situation where a tree of maximum size requires all of its constituent nodes to be edited. If the maximum number of operands that any one function accepts is two, as is often the case, then the maximum parse tree size, $T_p$, for a binary tree can be calculated using Equation 4.

$$T = 2^{D_p+1} - 1 \tag{4}$$

The absolute similarity is calculated in the same manner as the relative similarity, except that, in this case, the number of nodes compared, $n_{ij}$ , is equal to the size, $T_p$, of the largest possible tree. Therefore, the absolute similarity between two parse trees may be calculated by substituting Equations 3 and 4, as shown in Equation 5.

$$s_{ij} = 1 - \frac{d_{ij}}{T_p} \tag{5}$$

The edit distance similarity metric, whether absolute or relative, forms the basis of a mate selection scheme. Mates are chosen with direct reference to how genotypically dissimilar they are.

### 4.5 Self-Adaption

In order to emulate the success of sexual selection observed in natural evolution, the mate selection algorithm itself must be able to evolve during the run. It must be able to change its behaviour depending on past evolutionary experience. Therefore, there must be a variable within the algorithm which can be adapted during a run. This variable should directly decide how much of an effect the mate selection algorithm has upon the modified tournament.

The mate choice mechanism only affects the crossover operator: it is the only operator which requires two parent parse trees. When a crossover operation is to be carried out, a choice must be made: whether to use the standard or modified tournament selection schemes. Thus, the similarity-based mate choice algorithm is used with a given probability. The probability of using the mate selection scheme is defined as $p_{mate}$. Changing the value of $p_{mate}$ during a run of the system changes the balance between the exploration of the search space and the exploitation of the fittest individuals. When the value of $p_{mate}$ is low, the system is more likely to use the regular tournament selection scheme. This scheme, which chooses mates purely based on fitness, exploits the fittest individuals in the population to find the adaptive peak around which the population has converged. Conversely, when the value of $p_{mate}$ is high, the system is more likely to use the modified tournament selection scheme. The modification to the tournament selection scheme is such that mates have a greater probability of selection if they do not occupy the same region of the search space.

Self-adaption can be implemented at a number of levels and by using a number of different implementations. Three such adaptive levels were defined by Angeline [1]. For this study, two different self-adaption methodologies are presented.

#### 4.5.1 Population-Level Dynamic Probability

The self-adaption of the mate selection probability at the population level corresponds to the mate selection requirements of the average individual in the population. As the simulated evolution progresses, there will be a change in the parse tree similarity requirement. This requirement can be quantified by observing the success rate of the crossover operations. In one generation, many crossover operations will take place. Depending on the value of $p_{mate}$, the crossover operator will either use the standard tournament or the mate choice tournament to select the second parent parse tree. If the crossover operations that use the mate choice tournament are more successful than the crossover operations that use the standard tournament, then the value of $p_{mate}$ should logically increase. Consequently, if the mate selection algorithm is relatively unsuccessful, the value of $p_{mate}$ should decrease.

A successful crossover operation is one which results in an increase in fitness. The success rate of the crossover operation, however, is very low. This problem is addressed by defining the dynamic probability to follow the square of the number of successful operations. A minimum probability, $p_{min}$, must also be defined to prevent the probability from dropping to zero and being excluded from the remainder of the run. Equation 7 shows how the probability of using the mate selection is calculated. The symbol $C_{suc}$ is the successful crossover operations (squared to amplify the small differences in success rates), $C_{tot}$ is the total number of crossover operations and $R$ is the rate of squared success. The *mate* subscript refers to the mate selection scheme, and the *std* subscript refers to the standard selection scheme.

$$R = \frac{C_{suc}^2}{C_{tot}} \qquad (6)$$

$$p_{mate} = p_{\min} + \frac{(1-2p_{\min})R_{mate}}{R_{mate} + R_{std}} \qquad (7)$$

Equation 7 returns a super-linear increase in probability as the number of successful operations increases, due to the square term in the calculation of the rate of squared success (rather than using a standard success rate calculation).

Using this method, the population changes the preference with which it selects mates. The other way in which self-adaption may be applied to the problem of mate selection probability is at the level of the individual.

### 4.5.2 Individual-Level Dynamic Probability

Rather than force the entire population to act in the best interests of the average individual, the freedom of individual evolution must be granted. Based on Angeline's parameter tree adaption [1], each individual must have its own locally stored $p_{mate}$ variable, allowing mate preference to evolve to suit the needs of that individual. Adapting the value of $p_{mate}$ based on success rate makes less sense at the level of the individual: there is only one crossover operation with which to judge success by. Instead, the mate selection probability is adapted using a numeric evolutionary method. Before creating each new generation, random noise is applied to every individual's mate selection probability. The probability after noise has been applied, $p'_{mate}$, is given in Equation 8, where $N$ is a Gaussian random variable of mean 0 and variance 1, and $\alpha$ is a scaling constant, which, for this study, was set to $\alpha = 0.1$.

$$p'_{mate} = p_{mate} + \alpha N(0,1) \qquad (8)$$

There are a few differences between the way in which the value changes, partly because $p_{mate}$ is a probability rather than an unbounded strategy parameter, and partly because it is not success-based. Firstly, it does not matter if the value of $p_{mate}$ drops to zero because the future value does not rely on any measurement of success: the parameter is still able to escape from being zero-valued. Secondly, the value must be in the range $0 \le p_{mate} \le 1$ to be valid probability. If $p_{mate}$ falls below zero, it is simply assigned the value of zero. Similarly, the value is also forbidden to rise above the value of unity. Finally, due to the range of $p_{mate}$, and the fact that it is perfectly valid for it to be zero-valued, the value is adapted linearly with respect to its previous amplitude. This is because the probabilities must lie within a certain range, in contrast to Angeline's implementation, which was unbounded and exponential (since they were just arbitrary parameters). Additionally, it is unlike the subtree crossover parameter tree adaption described in Equation 8, which had no upper limit.

This form of probability adaption relies on selection retaining those individuals who have a favourable mate preference parameter whilst allowing that value to evolve over generations.

## 5 Results

The performance and behaviour of the mate selection schemes described in this paper were tested using a standard GP system. The simulation software was written in C++ using the description of the GP paradigm by Koza [11] and Banzhaf et al. [2].

A common set of parameters was used for each of the test problems. To ensure the maximum variety of program shape and size, the method used to generate the initial population was the 'ramped half-and-half' method described by Koza [11]. The selection mechanism used was tournament selection in either its standard form, or in the modified form to allow for mate selection. Single point crossover and subtree mutation were used, together with a standard set of operator probabilities: the crossover probability is high, the reproduction probability is low, and the mutation probability is very low. If a suitable solution was not found, a run was terminated after a given number of generations. Since the GP system is based purely on random numbers, the performance of the system must be assessed over a number of runs. To ensure quality results, each parameter set was averaged over 150 runs. The default parameter set is shown in Table 1.

| | |
|---|---|
| Generative method | Ramped half-and-half |
| Selection type | Tournament selection |
| Tournament size | 6 |
| Reproduction rate | 10% |
| Crossover type | Single point crossover |
| Crossover rate | 88% |
| Mutation type | Subtree mutation |
| Mutation rate | 2% |
| Termination generation | 100 |
| Run repetitions | 150 |

**Table 1. Summary of the default parameters used.**

### 5.1 Test Problems

The performance GP system was tested using a collection of well-known problems. The problems were chosen for their varying difficulty, although they are all under the general class of symbolic regression.

#### 5.1.1 Quartic and Heptic Polynomials

The two test problems described in this section represent standard algebraic symbolic regression problems of two different difficulty levels. The target function for the quartic problem is $f(x) = x^4+x^3+x^2+x$ and the target function for the heptic problem is $f(x) = x^7+x^6+x^5+x^4+x^3+x^2+x$. These problems are similar to the algebraic symbolic regression problems described by Koza [10] and use similar function and terminal sets. The function set consists of a general toolbox of algebraic functions, and the terminal set consists solely of the independent variable, $x$. The population size is a standard value for this type of problem. Similarly, the parse tree depth limit has been set to a sufficiently large value.

To protect the division function from division-by-zero errors, if the denominator of a division is zero, the output is simply assigned the value of unity. The log function, like the division function, is also protected. To avoid a complex output, the absolute value is fed into the log function. Additionally, since $\log(0) = \infty$, if the input value is zero, the output is simply assigned the value of unity to avoid errors. Fitness is evaluated over twenty test cases over a given range. A test case is regarded as a 'hit' if, when compared against the target function, it is within a certain small threshold. A successful solution scores a hit for each of the test cases. The parameter sets for the quartic and heptic problems are summarized in Table 2.

#### 5.1.2 Quadratic Polynomial with Irrational Coefficients

The quadratic test problem is an example of a simple algebraic symbolic regression problem. The target function is a quadratic polynomial with irrational coefficients: $f(x) = ex^2 + \pi x$.

| | |
|---|---|
| Target functions | $f(x) = x^4 + x^3 + x^2 + x$<br>$f(x) = x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x$ |
| Population size | 500 |
| Depth limit | 12 |
| Function set | {+,−, ∗, /, sin, cos, exp, log} |
| Terminal set | {$x$ } |
| Fitness cases | 20 random points; $-1 \leq x \leq 1$ |
| Hit threshold | ±0.02 |
| Success predicate | 20 Hits |

**Table 2. Parameters for quartic and heptic problems.**

Since the target function is only of second order, the GP system should find this the easiest of the test problems. In order to test the behaviour of the GP system in a different situation, the problem is given a parameter set which makes the problem significantly harder.

Firstly, the population size is reduced to half of the standard value for this type of problem. Secondly, the depth limit is reduced to two-thirds that of the other algebraic symbolic regression problems in this study. Thirdly, the function set used for this problem is quite small; only simple algebraic functions are included. Algebraic functions that display high-order behaviours, such as trigonometric functions, have been removed. The terminal set for this problem consists of the independent variable, $x$, an ephemeral random constant, $R$, which is in the range $-1 \leq R \leq 1$.

Fitness is calculated in exactly the same way as for the quartic and heptic problems: there are twenty comparisons made against the target function in a given range. Similarly, a successful solution is qualified in exactly the same manner: that the evolved algebraic function has a 100% hit rate. The parameter set for this problem is summarised in Table 3.

| | |
|---|---|
| Target function | $f(x) = ex^2 + \pi x$ |
| Population size | 250 |
| Depth limit | 8 |
| Function set | {+,−, ∗, /} |
| Terminal set | {$x$,$R$} |
| Fitness cases | 20 random points; $-1 \leq x \leq 1$ |
| Hit threshold | ±0.02 |
| Success predicate | 20 Hits |

**Table 3. Parameters used for the quadratic problem**

#### 5.1.3 The MAX Problem

The MAX problem [9] is significantly different from the previously described symbolic regression problems. The objective is to evolve a program which returns the largest value for a given function set and depth limit. The depth of the root node is defined as depth zero. Unlike the problems described so far, there is no independent variable for this problem, only a constant value. Therefore, only one fitness case is required to evaluate each candidate solution.

The maximum value of a tree is entirely dependent on the function set, the terminal set, and the maximum depth of the parse tree, $D_p$. For this problem, the function set consists only addition and multiplication operators, and the terminal set consists solely of the constant value 0.5. Using these function and terminal sets, there are a limited number of solutions to the problem. For example, consider a tree of a maximum depth of four nodes. In order to construct a tree which evaluates to the maximum value, the lower two levels of the tree must be configured such that each of the eight subtrees add up the terminal 0.5 values. In the level above, each of the four subtrees must be configured to add together the lower levels. This produces a value of 2.0 at each of the four lower subtrees. In the next level above either multiplication or addition may be used, since $2 \times 2 \equiv 2 + 2$, making the final two subtrees evaluate to 4.0. The top-level node (at depth zero) must be a multiplication operator to produce the maximum value. Since the only variation possible is at depth one, there are four possible solutions to the MAX problem in this case.

The maximum value for these function and terminal sets is given by $4^{2D_p-3}$ For the test problem in this study, the maximum depth was set to $D_p = 6$, so the maximum attainable value is 65536. Table 4 summarises the parameter set used for the MAX problem.

| | |
|---|---|
| Population size | 250 |
| Depth limit | 6 |
| Function set | $\{+, *\}$ |
| Terminal set | $\{0.5\}$ |
| Success predicate | Evaluates to 65536 |

**Table 4. Parameters used for the MAX problem.**

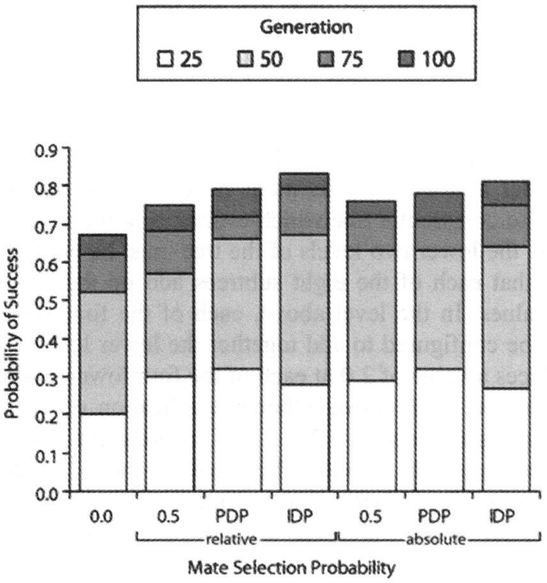
### 5.2 Mate Selection based on Edit Distance

Two ways in which the edit distance can be calculated have been defined. This section documents the behaviour of both, and compares them against the standard tournament selection mechanism. The behaviour of the new systems using self-adaptive methods is also assessed.

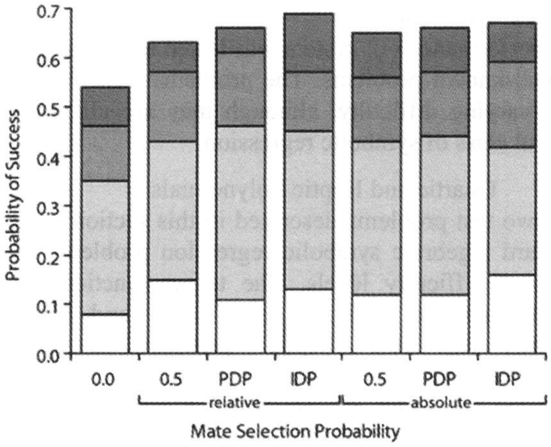
To observe the behaviour, three tests were considered: using the relative edit distance, using the absolute edit distance, and not using mate selection. For the two cases where the edit distance was being tested, the mate selection probability was set in three different ways: to a constant value of $p_{mate} = 0.5$; to a value starting at $p_{mate} = 0.5$, but being adapted on a population level based on success (PDP); and to a value starting at $p_{mate} = 0.5$, but being adapted on an individual level using a Gaussian random variable (IDP). The control case where no mate selection is used is the equivalent of using a mate selection probability of $p_{mate} = 0.0$.

The results in Figure 2 show that the system's performance is greater when mate selection is used. In general, using the relative edit distance measurement gives a greater performance boost.

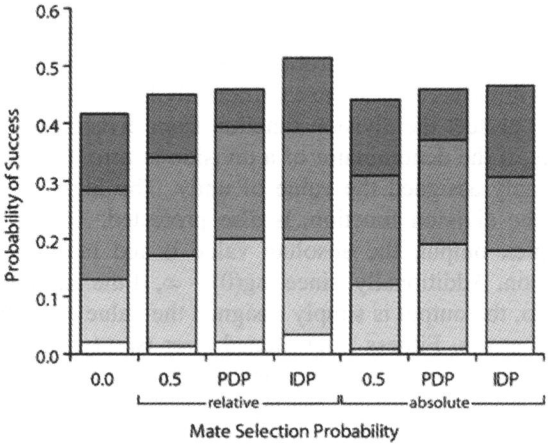
Of all the different ways in which mate selection has been implemented, the IDP self-adaptive method using relative edit distance is the most successful.



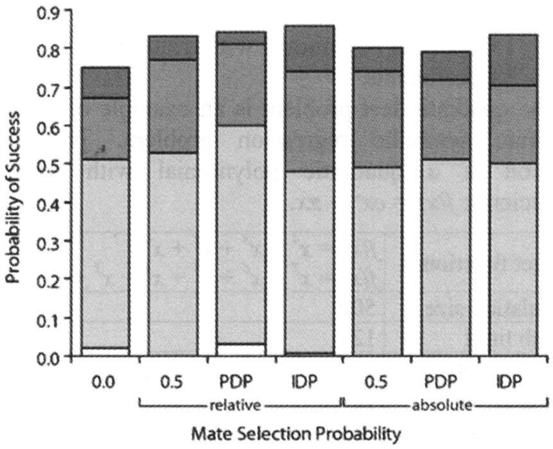
**(a) Quartic problem.**



**(b) Heptic problem.**



**(c) Quadratic problem.**



**(d) MAX problem.**

**Figure 2. Graphs showing how the performance of the system is affected by using edit distance**

Figure 2 presents results which provide clues to the way in which the different methods of self-adaption boost evolutionary performance. In general, since the population-level self-adaption is success-based, more of the successful runs produce solutions earlier on in the evolution.

Moreover, there is less chance of a successful solution occurring later on in the run. When contrasted with the individual-level self-adaption, the behaviour is different: since self-adaption occurs without regard to rate of improvement due to crossover, more successful solutions are generated later on in the evolution. Also, in the IDP test case the success rate earlier on is lower than in the PDP test case.

## 6 Conclusions

The results presented show that mate selection accelerates evolution. Using a direct measure of genetic similarity, the acceleration effect was shown to be enhanced by self-adaptive methods. These results support the claim that the overall performance of GP can be enhanced if the evolutionary model is extended to incorporate advanced features of the process of natural evolution.

Two different ways of defining the similarity were presented: relative and absolute. The relative distance represents how different two specific trees are, whereas the absolute distance represents how different any two trees are. The subtle difference in the way in which the similarity is calculated produces a significant difference in the results. The way in which the two mechanisms work must, therefore, have a bearing on the direction the evolution takes. As the population converges, the parse trees become more similar. The way in which the relative similarity is calculated pronounces slight differences between the trees. The absolute similarity measurement, however, treats all differences between trees as being the same. By definition, the relative distance lies in the range $0 \leq r_{ij} \leq 1$, whereas the absolute distance lies in the range $0 \leq d_{ij} \leq T_p$ . So, the same relative distance may be measured for two randomly generated trees in the initial population and two syntactically similar trees much later on in the evolution. The relative similarity between two individuals is calculated with reference to how different they could potentially be. As such, as the state of the population changes, the way in which the similarity is calculated changes with it.

Self-adaption has proved to be a valuable performance-enhancing addition to the distance-based mate choice algorithm. The adaption of the mate selection probability represents an evolutionary strategy relevant to what level of similarity is required at that time. As the value of $p_{mate}$ changes throughout the run, there is a change of evolutionary emphasis. The higher the mate selection probability, the more exploration takes place. Conversely, the lower the mate selection probability, the more natural selection exploits the fittest individuals in the population.

Further work is in progress to evaluate the diversity of the populations generated and how the computational edit-based similarity technique reported here, compares with a memory-based ancestry similarity technique.

## Bibliography

[1] Angeline, P.J. Adaptive and self-adaptive evolutionary computations. In M. Palaniswami, Y. Attikiouzel, R. Marks, D. Fogel, and T. Fukuda, editors, *Computational Intelligence: A Dynamic Systems Perspective*, pages 152–163. IEEE Press, Piscataway, NJ, 1995a.

[2] Banzhaf, W., Nordin, P., Keller, R.E., and Francone, F.R. *Genetic Programming: An Introduction*. Morgan Kaufmann, 1998.

[3] Burke, E., Gustafson, S. and G. Kendall. Diversity in genetic programming: an analysis of measures and correlation with fitness. *IEEE Transactions on Evolutionary Computation*, 8(1):47–62, Feb 2004.

[4] Cronin, H. *The Ant and the Peacock: Altruism and Sexual Selection from Darwin toTtoday*. Cambridge University Press, 1991.

[5] Ekárt, A. and Németh, S.Z. Maintaining the diversity of genetic programs. In J. A. Foster, E. Lutton, J. Miller, C. Ryan, and A.G.B. Tettamanzi, editors, *Genetic Programming, Proceedings of the 5th European Conference, EuroGP 2002, volume 2278 of Lecture Notes in Computer Science*, (Kinsale, Ireland, 3-5 April 2002). Springer-Verlag, 2002, 162–171.

[6] Eshelman, L.J. and Schaffer, J.D. Preventing premature convergence in genetic algorithms by preventing incest. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, (San Diego, CA, 1991). Morgan Kaufmann, 1991, 115–122.

[7] Fernandes, C., Tavares, R., Munteanu, C., and Rosa, A. Assortative mating in genetic algorithms for vector quantization problems. In *Proceedings of the 2001 ACM Symposium of Apllied Computing*, 2001, 361–365.

[8] Fernandes, C. and Rosa, A. A study on non-random mating and varying population size in genetic algorithms using a royal road function. In *Proceedings of the IEEE Congress on Evolutionary Computation*, 2001.

[9] Gathercole, C. and Ross, P. An adverse interaction between crossover and restricted tree depth in genetic programming. In J. R. Koza, D.E. Goldberg, D.B. Fogel, and R.L. Riolo, editors, *Genetic Programming 1996: Proceedings of the First Annual Conference*, (Stanford University, CA, USA, 28–31 July 1996). MIT Press, 1996, 291–296.

[10] Huang, C.F. *A study of mate selection schemes in genetic algorithms – Part I.* Internal Report LAUR 03-1809, Los Alamos National Laboratory, 2003.

[11] Koza, J.R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* MIT Press, 1992.

[12] Miller, G.F. and Todd, P.M. The role of mate choice in biocomputation: Sexual selection as a process of search, optimization, and diversification. In W. Banzhaf and F.H. Eeckman, eds, *Evolution and Biocomputation: Computational Models of Evolution*, 169–204. Springer, 1995.

[13] O'Donald, P. *Genetic Models of Sexual Selection.* Cambridge University Press, 1980.

[14] Ratford, M., Tuson, A., and Thompson, H. An investigation of sexual selection as a mechanism for obtaining multiple distinct solutions. In *Emerging Technologies*, 1997.

[15] Ratford, M., Tuson, A. and Thompson, H. The single chromosome's guide to dating. In *Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms*, 1997.

[16] Ridley, M. *Evolution,* Blackwell Science Inc. Oxford, 2003.

[17] Todd, P.M. and Miller, G.F. Biodiversity through sexual selection. In C. G. Langton and K. Shimohara, editors, *Artificial Life V: Proceedings of the Fifth International Workshop on the Synthesis and Simulation of Living Systems*, (Cambridge, MA, 1997). MIT Press/Bradford Books, 1997, 289–299.