CSSE Technical Report UWA-CSSE-07-002
February 2007
Statistical Exploratory Analysis of Genetic Algorithms: The Detrimentality of Crossover

Andrew Czarn and Cara MacNish and Kaipillil Vijayan and Berwin Turlach
School of Computer Science & Software Engineering
The University of Western Australia
Crawley WA 6009 Australia
email: aczarn@csse.uwa.edu.au
tel: +61 8 6488 1945
fax: +61 8 6488 1089

Abstract

The traditional concept of a genetic algorithm (GA) is that of selection, crossover and mutation. However, a limited amount of data from the literature has suggested that the niche for the beneficial effect of crossover upon GA performance may be smaller than has traditionally been held. Based upon previous results on not-linear-separable problems we decided to explore this by comparing two test problem suites, one comprising non-rotated functions and the other comprising the same functions rotated by 45 degrees rendering them not-linear-separable.

We find that for the *difficult* rotated functions the crossover operator was detrimental to the performance of the GA. We conjecture that what makes a problem *difficult* for the GA is complex and involves factors such as the degree of optimization at local minima due to crossover, the bias associated with the mutation operator and the Hamming Distances present in the individual problems due to the encoding.

Finally, we tested our GA on a real world landscape minimization problem to see if the results obtained would match those from the *difficult* rotated functions. We find that they match and that the features which make certain of the test functions difficult are also present in the real world problem.

Keywords: detrimentality, crossover, rotated functions, linear-separable, not-linear-separable

1 Introduction

It has been traditionally maintained that the crossover operator is an integral component of a genetic algorithm. This has been held to the extent that many GA practitioners believe that it is the inclusion of the crossover operator that distinguishes GAs from all other optimization algorithms [6].

Despite this, work by Eshelman and Schaffer, entitled *Crossover's Niche*, suggested that there exists a unique niche for which crossover is advantageous and this is smaller than has been traditionally held by the GA community. As such, it is an open question as to how important the crossover operator is for many real-world problems [9].

In our earlier work [5] we observed that the crossover operator is advantageous for linear-separable real-valued optimization problems such as De Jong's *F1* and *F3* but detrimental for not-linear-separable problems such as De Jong's *F2* and Schaffer's *F6* [5]. We define linear-separable problems as those where the objective function can be written as a sum of univariate functions, which are allowed to be non-linear, where each of the functions can take one component of the input vector as an argument.

Our observation prompted us to explore instances where crossover may be beneficial or detrimental by comparing two test problem suites that we formulated. The first suite comprises linear-separable problems. The second suite comprises the same suite of problems but having been rotated by 45 degrees in the solution space rendering them not-linear-separable.

In this paper the literature that the crossover operator may not always be useful in improving GA performance is reviewed in Section 2. This is followed in Section 3 by a discussion of the observations we have made in our previous work which prompted us to embark upon the research described in the following sections. In Section 4 we describe our experimental set-up including refinements to our previously published statistical methodology. Section 5 describes the results of our experimental work. Section 6 reviews the factors that influence the detrimentality of crossover. Section 7 demonstrates that the detrimentality of crossover is a real world phenomenon. A discussion in Section 8 concludes the paper and suggests areas for future research.

2 Review of the Usefulness of Crossover

A genetic algorithm works by encoding potential solutions to a problem as a series of bits or *genes* on a bit-string or *chromosome*. The mechanics of a genetic algorithm are straightforward: in its simplest form new solutions are generated using *crossover*, where genes are swapped over between pairs of chromosomes, and *mutation*, where the binary value of a gene is inverted.

From a traditional perspective it has been maintained that crossover is a necessary inclusion in a GA. This has been sustained to the extent that some practitioners hold that the resultant algorithm is no longer a GA if the crossover operator is omitted [6]. Mutation, on the other hand, has been traditionally seen as a background operator with the unique role, as described by Holland, of ensuring that no allele or value of a bit character (1 or 0) permanently disappears from the population [11].

However, over time this traditional stance has been an issue of contention, with some GA practitioners claiming that a GA without crossover, known as *naive evolution* (which comprises of selection and mutation only), may be a much more powerful search algorithm than the GA community has previously held [8].

An initial example of research in this area is that by Eshelman and Schaffer in their paper entitled *Crossover's Niche*. The authors argued that what distinguishes the GA among population-based hillclimbers is pair-wise mating and that problems can be devised where crossover gives a competitive advantage. However, it was conjectured that many problems do not have these features and it remains an open question as to how important crossover may be for real world problems. In addition, because GAs are susceptible to premature convergence the niche for which crossover is beneficial to GA performance may be smaller than most GA practitioners maintain [9].

Jones [14] added to this by showing that a macromutational hillclimber (one that involves large scale mutations) easily outperforms a standard GA on Hollands's Royal Road problem [18] which has the properties that Eshelman and Schaffer ascribe to problems residing in *Crossover's Niche*. Thus the *niche* may be even smaller than Eshelman and Schaffer had anticipated.

Interestingly, Salomon conjectured that *Crossover's Niche* is in fact linear-separable problems [23]. From his work with Rastrigin-like functions he conjectured that crossover implicitly exploits the decomposability property of the fitness function: the optimization is decomposable into n independent one-dimensional (one bit string) sub-problems [23]. Salomon's conjectures resulted from observations of empirical data and have yet to be confirmed by statistical analysis.

The above papers drew attention to the idea that under particular circumstances the crossover operator may not prove useful in improving GA performance. However, limited research has taken place exploring this issue.

One study which has addressed this topic to a limited degree is that by Schaffer et al [24]. These researchers conducted a factorial design study using the analysis of variance (ANOVA) studying the De Jong suite plus an additional five problems. Close examination of the best online pools suggested a relative insensitivity to the crossover operator when using Gray encoding. It was suggested that this is because the use of Gray encoding makes searching the solution space much less susceptible to Hamming cliffs [24, 8]. However, Schaffer et al did not conjecture the general type of problems that would demonstrate this behaviour.

Further debate about the usefulness of the crossover operator was made by Fogel and Atmar [10]. These researchers conducted several experiments requiring solving systems of linear equations. The authors concluded that the crossover operator provided no significant benefit. Rather, random mutation consistently generated more efficient searches [10].

In describing the above studies it must be noted that a number of criticisms have been made about the work carried out so far. For example, Reeves and Wright suggested that the amount of information in a sample can never be sufficient to enable one to decide on the nature of the epistasis in a problem [21]. This implies that the problems Eshelman and Schaffer describe as being most apt for the crossover operator may not be easily recognizable in practice.

Moreover, in reference to the work of Schaffer et al [24], the study itself was limited by a number of issues. These included little attention given to blocking for seed as a source of variation or noise, issues dealing with the calculation of power and sample size were not considered, and there was a lack of a detailed analysis of response curves.

In overview, while there is some suggestion from the literature that the crossover operator may not always be useful, it appears unclear as to when this may occur and the possible reasons for its occurrence.

Next, we re-examine observations from our previous work that provided the impetus for our present research.

3 Observations from our Earlier Work

Our work into the statistical exploratory analysis of genetic algorithms involved Analysis of Variance (ANOVA) examination of four benchmark functions. These were: De Jong's *F1* [7] known as the SPHERE, De Jong's *F3* [7] known as the STEP function, De Jong's *F2* [7] known as ROSENBROCK'S SADDLE and Schaffer's *F6* [6]. These were all implemented as minimization problems and are displayed in Equations 1, 2, 3 and 4, respectively:

$$f_1(\mathbf{x}) = \Sigma_{i=1}^{3} x_i^2, -5.12 \leq x_i \leq 5.12, \tag{1}$$

$$f_3(\mathbf{x}) = \Sigma_{i=1}^{5} \lfloor x_i \rfloor, -5.12 \leq x_i \leq 5.12, \tag{2}$$

$$f_2(\mathbf{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2, -2.048 \leq x_i \leq 2.048, \tag{3}$$

$$f_6(\mathbf{x}) = 0.5 + \frac{(\sin\sqrt{x_1^2 + x_2^2})^2 - 0.5}{(1.0 + 0.001(x_1^2 + x_2^2))^2}, -100.0 \leq x_i \leq 100.0. \tag{4}$$

We found that for De Jong's *F1* and *F3* the traditional GA, where crossover was included, performed optimally when the crossover rate was 100%. In contrast for De Jong's *F2* and Schaffer's *F6*, the crossover operator was statistically demonstrated to be having a *detrimental* effect upon performance. We also found for these latter two functions that the ANOVA interaction term between crossover and mutation was significant and negative, which indicates an inverse relationship between crossover and mutation. Moreover, the difficulty of a problem was associated with the optimal mutation rate, with De Jong's *F2* and Schaffer's *F6* demonstrating optimal mutation rates significantly higher than traditional recommendations. This is consistent with other mutation rules reported in the literature. For example, Petrovski, Wilson and McCall who carried out fractional factorial experiments in the domain of anti-cancer chemotherapy. These were combined with linear regression in order to pinpoint which parameters were significant and estimate their best values. They found bit flipping mutation rates in the order of 19.81% and 10.69% to be the most optimal [20] in their experiments.

When considering the possible difference in these functions that could produce such varied results a clear demarcation between them was that De Jong's *F1* and *F3* are linear-separable problems, echoing the conjecture made by Saloman that linear-separable problems are *Crossover's Niche* [23]. In contrast, De Jong's *F2* and Schaffer's *F6* are not-linear-separable problems.

The question that arose from our work was that, if we compared two test function series *differing only* in that one test function series was linear-separable while the other was not-linear-separable, would we see the same pattern?

The two test function series we decided to compare comprised firstly of the test function series, *FNn*, which we have used in prior work to examine the importance of the ANOVA interaction term between crossover and mutation [4]. This is a linear-separable problem which increases in modality as the value for n increases. We chose an arbitrary number of six functions as our test suite.

The second test function series we used to compare this against consisted of these same functions rotated by 45 degrees in the solution space. This rotation rendered the series of problems, which we call *FNnR45*, not-linear-separable.

By comparing the linear-separable form of the problem to the not-linear-separable form we expected to see a difference in the effect of the crossover operator. Given the suggestions from the literature and our own experience with linear-separable versus not-linear-separable functions, we conjectured that we would see a largely beneficial effect of crossover for the linear-separable problems, *FNn*, but some detrimental effect of crossover for the not-linear-separable problems, *FNnR45*. Furthermore, if the latter turned out to be true, then we would attempt to explain the reasons why crossover acts detrimentally for (specific) not-linear-separable problems.

Finally, given the conjecture by Eshelman and Schaffer that it remains an open question as to how important crossover may be for real-world problems [9] we decided to trial our GA on a practical (but still highly multimodal) landscape minimization problem to see if the results from our test functions would carry over to those obtained on the real world landscape.

In Section 4 we outline our experimental set-up.

4 Methods

Much of our methodology has been published elsewhere [5, 4, 1]. However, we briefly overview aspects pertinent to this paper and also elaborate on any additions to our methods.

4.1 Description of our Test Function Series

As described above, we compared a linear-separable test function series *FNn* against its rotated form (rotated by 45 degrees in the x and y coordinate solution space) rendering it a not-linear-separable test function series *FNnR45*. We then tested the algorithm on a newly devised benchmarking problem from the Huygens Suite [17, 16]. These are described below:

1. Test function *FNn* for n=1 to n=6, which are linear-separable equations, as displayed in Equation 5 below:

$$FNn(x_1, x_2) = \Sigma_{i=1}^{2} 0.5(1 - \cos(\frac{n\pi x_i}{100})e^{-|\frac{x_i}{1000}|}), -100 \leq x_i \leq 100. \quad (5)$$

2. Test function *FNnR45* (*R45* standing for the original test function *FNn* having been *rotated by 45 degrees* in the solution space), being not-linear-separable, for n=1 to n=6 as displayed in Equation 6 below:

$$\begin{aligned} FNnR45(x_1, x_2) = 0.5(1 - \cos(\frac{n\pi \frac{x_1 + x_2}{\sqrt{2}}}{100})e^{-|\frac{\frac{x_1+x_2}{\sqrt{2}}}{1000}|}) + \\ 0.5(1 - \cos(\frac{n\pi \frac{x_1 - x_2}{\sqrt{2}}}{100})e^{-|\frac{\frac{x_1-x_2}{\sqrt{2}}}{1000}|}), -100 \leq x_i \leq 100. \end{aligned} \quad (6)$$

3. MacNish has devised a problem series for benchmarking, that based on fractal landscapes, reflect the attributes of highly multimodal problems seen in real world situations [17, 16].

   We chose to run our GA on the first landscape in MacNish's 20 series for which a plot was provided, shown in Figure 1.

4.2 Implementation of the Genetic Algorithm

We implemented a genetic algorithm as detailed in Table 1. The implementation of the genetic algorithm was deliberately simple so that a clear and concise comparison of linear-separable versus not-linear-separable problems could be made. Further work could include other variants (eg different encodings, crossover strategies, selection mechanisms, elitism, etc) of genetic algorithms proposed in the literature to see if the results are consistent with those variants.

4.3 Experimental Design and Statistical Test

In order to control for the effect of seed we implemented a *randomized complete block* design and used the analysis of variance (ANOVA) in order to compare performances for 2 or more parameters. In ANOVA the null hypothesis is that the means for different levels of a parameter are equal. The alternative hypothesis is that the means for levels of a parameter are not all equal and thus we conclude that the parameter has an effect upon the response variable.

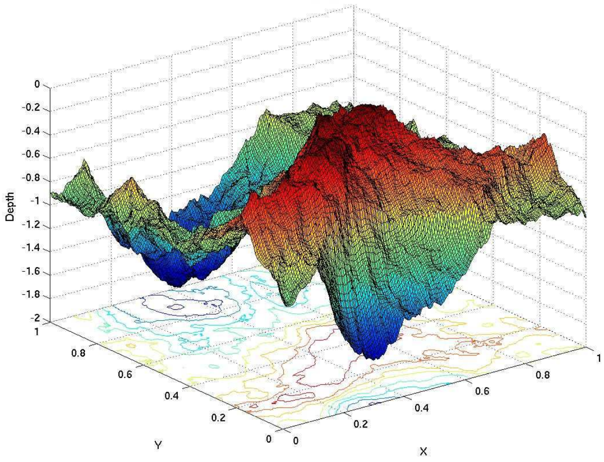

Figure 1: *Landscape 20_101 from the Huygens Suite [17, 16].*

4.4 Level of significance

There are 2 types of errors associated with statistical testing. A type I error is the rejection of the null hypothesis when it is true. A type II error is the non-rejection of the null hypothesis when the alternative hypothesis is true. The probability of making a type I error is denoted by α and the probability of a type II error is denoted by β. Since the null hypothesis represents the most conservative proposal it is considered that a type I error is more serious than a type II error [13].

For published research a level of significance of 1% is often used [15]. P-values less than 1% suggest that the null hypothesis is *strongly rejected* or that the result is *highly statistically significant* [13]. In the present study we have employed 1% as our level of significance.

4.5 Level of Significance for Orthogonal Simultaneous Multiple Comparisons

In a situation of orthogonal simultaneous multiple comparisons within a parameter it is necessary to modify the level of significance. This is because the probability of achieving one or more statistically significant results in n simultaneous independent multiple comparisons will exceed the level of significance chosen (1% in the present study). This is illustrated in Equation 7.

$$P(\textit{at least one significant result in } n) = 1 - (1-\alpha)^n. \qquad (7)$$

This occurs in ANOVA when the sum of squares for each parameter is partitioned into orthogonal contrast terms. In order to ensure that the probability of achieving one or more statistically significant results in n simultaneous multiple comparisons is *exactly* 1%, we use a modified level of significance for testing each of n orthogonal polynomial contrast terms calculated in accordance with Equation 8.

$$\textit{Modified level of significance} = 1 - (1 - \alpha)^{\frac{1}{n}}. \tag{8}$$

Our approach is different from the Bonferroni method [19] which, for the present work, would simply divide the overall level of significance by the number of simultaneous multiple comparisons. The Bonferroni method will ensure that the probability of achieving one or more statistically significant results in n simultaneous multiple comparisons is *no greater than* 1%. Thus, it yields an upper bound such that the actual probability of achieving one or more statistically significant results in n simultaneous multiple comparisons may be much smaller.

Table 1: Details of the genetic algorithm

Variable representation	Bit string
Bits per variable	22
Genes	Binary value 1 or 0
Population size	50 chromosomes
Chromosome coding	Gray coding
Selection	Probabilistic selection [1]
Experimental unit	Blocks containing independent runs of the genetic algorithm for different crossover and mutation rates with the same seeds
Crossover	Single point (randomly selected) per variable
Mutation	Randomly generated bit replacement [2]
Performance measure	Final epoch ie epoch at which fitness of best chromosome $\leq 10^{-\text{threshold}}$ of maximum fitness (see Appendix A for details of the threshold for each test function)

---

[1] Probabilistic selection used here is the random selection of parents with the probability of selection being directly proportional to the fitness of a chromosome.

[2] Mutation is implemented as described by Davis [6]. That is, if the probability test is passed the binary bit is replaced by another binary bit that is randomly generated. Fifty per cent of the time the new bit will be the same as the old bit. The *bit-flipping mutation rate* is therefore half of the implemented mutation rate.

4.6 Power

As outlined previously it is imperative to have some means of calculating whether the size of the sample chosen has sufficient power. In order to do so it is necessary to specify the degree to which the null hypothesis is false. This can be done by using the effect size index, *f*, as described by Cohen [3].

Our initial work had been based on increasing the sample size by a factor of 5 until we achieved at least 80% power for detecting a difference of at least 5 epochs. However, as f is related to the standard deviation, which may differ considerably according to the problem under study, we refined our previous methodology by calculating power based on an accepted *standard* value of *f*.

In our initial research our simplest benchmark problem was De Jong's *F1* [7] which showed the smallest standard deviation. In reference to this problem a difference of at least 5 epochs was approximated by an f value of 0.4 which denotes a *large* effect. To obtain a power of at least 80% using this f value we required a pooled ANOVA analysis using 5 by 500 replicate data-sets. We therefore used 5 by 500 replicate data-sets as a starting point in the present study and then confirmed the level of power achieved.

Given our previous experience in power calculations with GA analysis, we suggest that 0.4 may be used as a standard for the effect size when attempting to analyze the performance of a GA. It should also be noted that in using this approach it is possible to calculate power *a priori* and thus ascertain if a given sample size will confer a required level of power. However, we continued to adhere to *post hoc* power calculations in line with our previous work.

4.7 Pooled Analysis Design

If large data-sets are required these may not be able to be analyzed when a parameter has too many levels resulting in the statistical software having to deal with too many and too large matrices. We again made use of a pooled analysis design for the present study as follows:

1. For each individual experiment we calculated the mean of the performance measure for each combination of crossover and mutation.

2. These data from individual experiments were concatenated into a new *pooled* data file. The response variable was now the mean of the performance measure averaged over the number of replicates in the individual experiment. This results in a smaller error variance as the average of a number of observations is expected to be closer than a single observation to the population mean.

   Each individual experiment denoted one level of the block parameter.

3. Analysis was carried out in the same manner as for individual experiments.

4.8 Estimates of Optimal Values for Crossover and Mutation

The aim of the present research was to explore the detrimentality of crossover. That is, to statistically determine the optimal crossover rate for each test function with detrimental crossover corresponding to an optimal crossover rate of 0%. We therefore used our previous methodology which enlisted polynomial regression to obtain an estimate of the optimal rate for both the crossover and the mutation operators.

5 Results

5.1 Exploratory analysis of test functions *FN1* to *FN6*

The results of exploratory analyzes for test functions *FN1* to *FN6*, are shown in Table A-1, Table A-2, Table B-1 and Table C-1. It should be noted that the results for *FN1* to *FN6* have been published elsewhere [4].

It can be seen that the crossover operator proved beneficial to the performance of the GA in every instance. For example, in Table C-1 the optimal value of crossover was 100% in every instance from *FN1* to *FN6*.

5.2 Exploratory Analysis of test functions *FN1R45* to *FN6R45*

The results of exploratory analyzes for test functions *FN1R45* to *FN6R45* are shown in Table A-3, Table A-4, Table B-2 and Table C-2. For the test function series, *FNnR45*, where the test function *FNn* had been rotated by 45 degrees in the solution space there was a marked difference in the results obtained.

Firstly, Table C-2 illustrates that crossover was detrimental for test functions *FN2R45*, *FN4R45* and *FN5R45*, where for these rotated forms the optimal crossover rate was 0%. This is in contrast to the non-rotated form of these functions, as described above, where in each case crossover proved to be beneficial.

Table C-2 also shows that where crossover was shown to be detrimental the mutation rate was also higher than in instances where crossover was having a beneficial effect. For example, for *FN2R45* the optimal mutation rate was 25.45% (bit flipping mutation rate of 12.72%), for *FN4R45* the optimal mutation rate was 35.30% (bit flipping mutation rate of 17.65%) and for *FN5R45* the optimal mutation rate was 33.38% (bit flipping mutation rate of 16.69%) . In contrast, for *FN1R45* the optimal mutation rate was 8.78% (bit flipping mutation rate of 4.39%), for *FN3R45* the optimal mutation rate was 12.36% (bit flipping mutation rate of 6.18%) and for *FN6R45* the optimal mutation rate was 12.97% (bit flipping mutation rate of 6.48%). Thus, in all cases where crossover was detrimental the optimal mutation rate proved to be notably greater than those instances where crossover was beneficial.

As noted above, as a high mutation rate is a conjectured marker for the difficulty of a problem the above results indicate that the crossover operator proved to be detrimental for the most difficult of the not-linear-separable rotated functions.

6 Factors Affecting the Detrimentality of Crossover

In the preceding work we observed that crossover was detrimental for three of the six not-linear-separable rotated functions analyzed. As indicated by the optimal mutation rates, these proved to be the most difficult of the six functions to solve. Thus, it is conjectured that crossover proves to have a detrimental effect upon GA performance if the not-linear-separable problem is *difficult* or hard for the GA to solve.

As discussed below, we find that what makes a GA hard to solve is a complex issue and involves factors such as the degree of optimization occurring at local minima due to crossover, the bias of the mutation operator and the Hamming Distances involved in the individual problems. In the next sections we discuss each of these factors in turn.

6.1 Optimization Occurring at Local Minima due to Crossover

The first factor which influenced the difficulty of the problem for the GA was the optimization occurring at local minima due to crossover. However, in order to discuss this we must first investigate what roles crossover, and also mutation, are playing in the GA.

Figure 2a, Figure 2b, and Figure 2c display chromosomes situated in one heat map of function *FN2R45*. The heat map represents a view of the function looking down from above with white areas denoting troughs and dark areas denoting peaks. The chromosomes are the 50 chromosomes that our GA contains. These heat maps show the location of the chromosomes during an iteration of the GA.

As can be seen in Figure 2b, which illustrates the location of chromosomes after crossover, the chromosomes have dissipated little, moving by only a small amount at the local minima sites (denoted by the white areas). In this case, crossover is performing its classical function of *exploitation* within, or converging on, the local minima occupied by the chromosomes [12].

In contrast, in Figure 2c after mutation the chromosomes have dissipated more widely over the solution space. In this sense, mutation is performing its classical function of *exploration* of the solution space [12]. It is also important to note that we found that it is largely only with mutation that the chromosomes are able to move out of the local optima that they are in and into newer regions of the solution space. This is seen most clearly by referring to the bottom right hand corner of the heat map for *FN2R45* where several chromosomes have moved from the local optimum situated there into outer lying regions of the solution space.

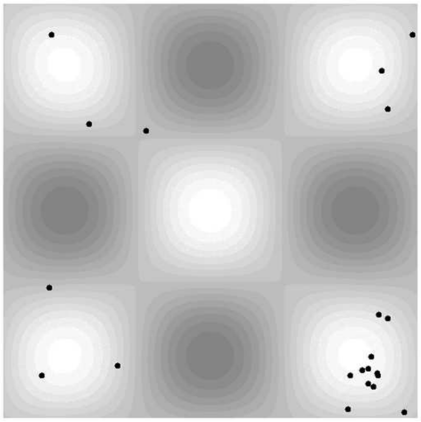

Figure 2a: *FN2R45*:Initial Chromosome Population.

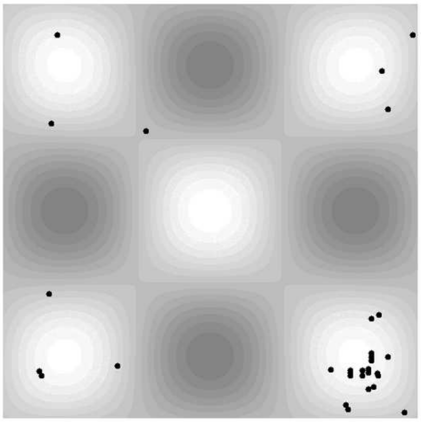

Figure 2b: *FN2R45*:Chromosome Population after Crossover.

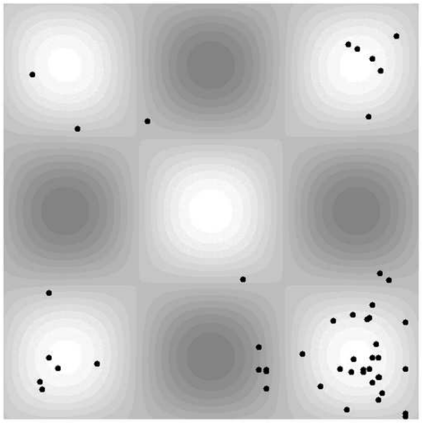

Figure 2c: *FN2R45*:Chromosome Population after Mutation.

In summary, a review of all the heat maps, including the examples shown, illustrated that while mutation was responsible for exploration of the solution space, crossover was enacting exploitation at the sites of local minima. That is, the heat maps showed that crossover was in effect responsible for optimization taking place at the site of local minima thereby keeping chromosomes "stuck" in those local minima. This meant that crossover was having the effect of hindering the movement of chromosomes from local minima into the global minimum.

In order to quantify the degree of optimization at the local minima carried out by crossover we recorded and compared the relative proportion of times crossover and mutation improved the best fitness obtained by the population.

The results were that crossover improved fitness at sites of local minima

82% of the time out of the total number of epochs (with a 99% confidence interval of 80% to 84%) compared to mutation with a value of only 30% (with a confidence interval of 29% to 31%). This lent support to what was visualized on the heat maps, namely, that optimization of chromosomes at local minima due to the crossover operator was hindering chromosomes moving out of these local minima into newer regions of the solution space.

6.2 Bias Associated with the Mutation Operator

The mutation operator corrupts the reproduction of genotypes thereby introducing the variety that fuels natural selection [2]. This being said, there is discussion in the literature as to the possible biases inherent in various implementations of mutation and the degree to which this makes a problem hard for a GA to solve [2, 22].

Thus, to ascertain in the present work if there was any bias associated with the mutation operator which might make the problems harder for the GA to solve we carried out experiments where many copies of a single chromosome comprised of two bit strings, which were initially placed in the center of the local minimum located in the bottom right hand corner of the heat map of *FN2R45*, underwent mutation and then were plotted onto the heat map surface of the rotated function. Figure 3 shows an example of this for *FN2R45* using the optimal mutation rate of 25.45% (bit flipping mutation rate of 12.72%) with 10000 samples.

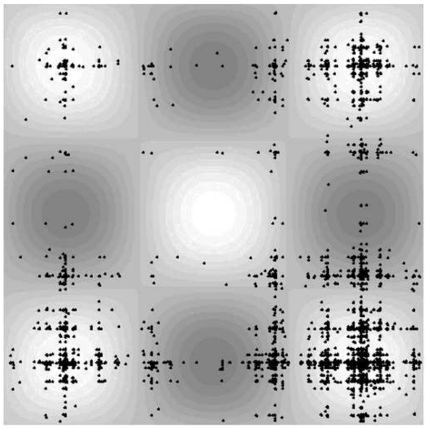

Figure 3: Mutation Plot for Test function *FN2R45*.

As can be seen, after mutation the chromosome landed in a criss-cross pattern along the x and y directions illustrating that it is biased in the axial direction. The reason for this may be explained using a simple example as follows.

Figure 4 illustrates the probabilities associated with moving in the x, y and diagonal directions for a single two bit chromosome. If we assume that a change in a bit has a probability of 10%, then movement in either the x or y direction

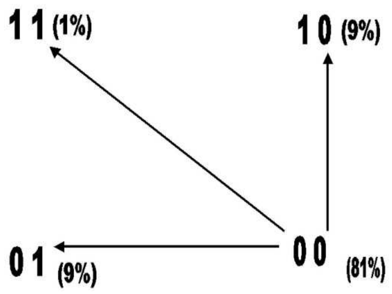

Figure 4: Probabilities associated with the movement of a single two bit chromosome after mutation.

has a probability of 9% (0.9 times 0.1). However, movement in the diagonal direction requires a change in *both* bit strings with a resultant probability of 1% (0.1 times 0.1). Also, the probability of no change occurring to the chromosome, and hence no movement, is 81% (0.9 times 0.9).

Simplistically speaking for the not-linear-separable problems we investigated, the degree to which this bias made the problem hard for the GA was related to the percentage of the local minima which lay on the x and y axes, given that the global minimum was at the origin. In Figure 5a for *FN2R45* none of the local minima lay on the x or y axes compared with Figure 5b for *FN3R45* where four of the twelve local minima lay on the x or y axes. Chromosomes in these local minima were more likely to be shifted towards the global minimum due to the bias of the mutation operator. Overviewing the results for all the rotated functions, we observed that if roughly 20% or more of the local minima lay along the x or y axes, as shown in Table 2, the crossover operator proved to be beneficial for the function, otherwise it was detrimental.

More generally speaking, this axial bias is a special case of the more general relationship between the problem encoding and the solution space, discussed below.

6.3 Relationship between Gray Encoding and the Solution Space

Figure 3 shows a bias not just in axial directions, but towards a grid-like pattern with regions of higher density and others of much lower density. In general it is much harder to make a “jump” to some areas of the space than others. The selection generator compounds the effect of this bias by eliminating candidates that are part way towards a better local minimum but have low fitness.

An illustrative case for the rotated functions is that of *FN2R45* and *FN3R45*. As shown in the response curves depicted in Figure 6a and Figure 6b, *FN2R45* was the more difficult of the two functions for the GA. This is evidenced by the fact that the number of epochs taken to reach the threshold was an order of magnitude greater. This is despite the fact that *FN3R45* is the more modal of

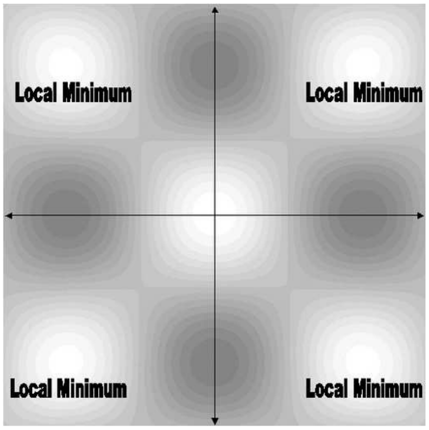

Figure 5a: Heat Map of *FN2R45* illustrating location of local minima along X and Y axes.

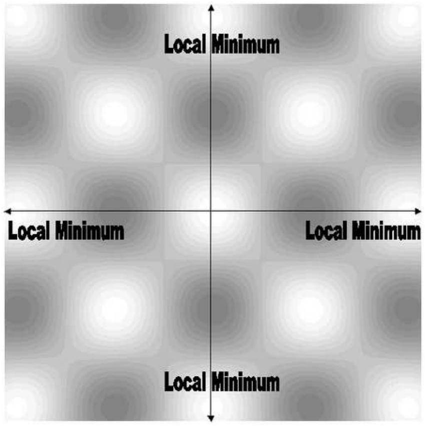

Figure 5b: Heat Map of *FN3R45* illustrating location of local minima along X and Y axes.

Table 2: Relationship between Local Minima and Detrimental Crossover

Test Function	% Local Minima on X and Y Axes	Detrimental Crossover
FN1R45	Nil Local Minima	No
FN2R45	0%	Yes
FN3R45	25%	No
FN4R45	16.67%	Yes
FN5R45	16.67%	Yes
FN6R45	20%	No

the two functions.

To illustrate why this is the case, we can examine the Hamming Distances of the two functions. The Hamming Distance is a measure of the difference or *distance* between two binary sequences of equal length. Hamming Distances between the global minimum and the surrounding local minima for functions *FN2R45* and *FN3R45* are shown in Figure 7a and Figure 7b, respectively.

As can be seen, *FN2R45* has the larger Hamming Distance of 12 from any of the local optima to the global optimum for either the x bit string or the y bit string. The probability of making this (exact) jump with a bit-flipping mutation rate of m for 44-bit chromosomes is:

$$P_1 = m^{24}(1-m)^{20}. \tag{9}$$

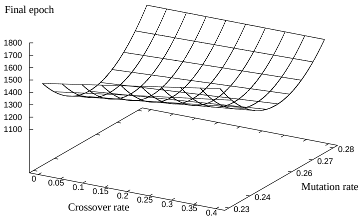

Figure 6a: Response curve for test function *FN2R45*.

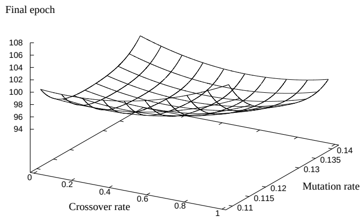

Figure 6b: Response curve for test function *FN3R45*.

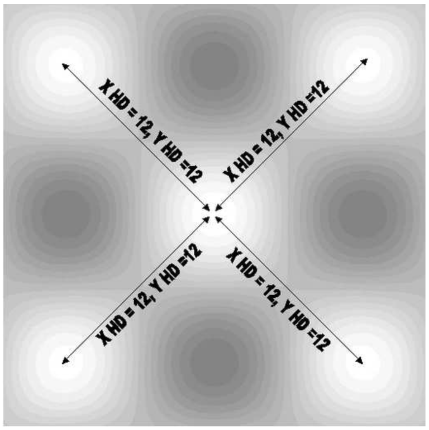

Figure 7a: Hamming Distances for *FN2R45*.

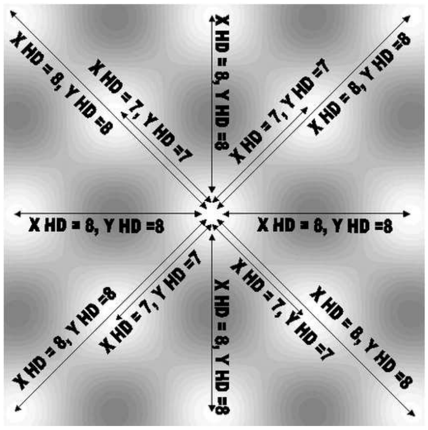

Figure 7b: Hamming Distances for *FN3R45*.

(Clearly a range of nearby jumps are possible, but we use the minima for illustration. The probability will be higher if nearby jumps are taken into account).

In contrast, for *FN3R45*, the Hamming Distance from any of the local minima to the global minimum is only 7 or 8. The probability of making the (exact) jump is therefore of the order:

$$P_2 = m^{15}(1-m)^{29}. \tag{10}$$

As can be seen in Figure 8, the probability of making the required jump is far greater for *FN3R45* for low mutation rates.

The larger Hamming Distances for the functions explained why the optimal mutation rate for *FN2R45* was higher (25.45% corresponding to a bit flipping rate of 12.72%) than for *FN3R45D* (12.36% corresponding to a bit flipping rate of 6.18%). This is because the greater Hamming Distances meant that a greater

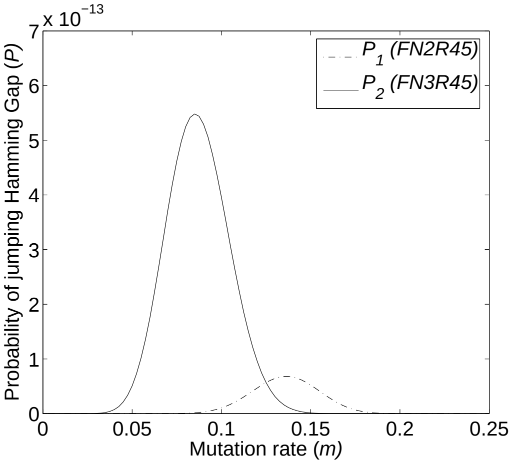

Figure 8: Probability of jumping Hamming Gap versus Mutation rate.

number of bit flips are required in order to move chromosomes from any of the local optima into the global optimum. These Hamming Distances are a direct consequence of the relationship between the encoding and the solution space.

It is interesting to note that finding the optimal mutation rate appears to be a case of finding a fixed point that is high enough up the Hamming Distance probability curves for the space while at the same time minimizing the disruptive effect of mutation on convergence.

7 Extending the Results to Difficult Practical Problems

We have discussed a number of properties that make a problem difficult for a GA to solve, such as high modality and local minima not artificially aligned within the encoding to make the solution easier and their impact on the performance of crossover. However, these have only been tested on artificial sequences of problems that possess features such as symmetry and a regular repetition of local minima. Before leaving this topic, we wanted to see if there was evidence the results would carry over to real-world problems exhibiting the same properties for difficult problems. In order to extend the results to a difficult practical problem, we tested our GA on *Landscape 20_101* shown in Figure 1. The results are shown in Table A-5, Table B-3 and Table C-2.

As can be seen, the same behaviour of the GA emerged as for the difficult rotated functions. That is, crossover, mutation and their interaction had a statistically significant effect upon GA performance. However, for crossover the effect was detrimental with an optimal crossover rate of 0%.

For mutation the optimal rate was a high 18.93% (bit flipping mutation rate of 9.46%), comparable to the high mutation rates seen with the difficult not-linear-separable problems discussed above. Again as noted above, a high mutation rate is a conjectured marker for the difficulty of the problem.

We can conjecture that this problem proved difficult for the GA for similar reasons to the problems analyzed earlier. In the first case the random arrangement of the local minima of this problem makes it unlikely that any of the local minima are aligned in the axial directions. Thus, the bias of mutation means that it is less likely that the global minimum will be found by chromosomes moving in the x and y directions.

In reference to crossover, the fact that the surface of the *Landscape 20_101* has a great number of local minima means that it is very likely that crossover was enacting optimization at the local minima sites. This is supported by the fact that the optimal mutation rate was high at 18.93% (bit flipping mutation rate of 9.46%), suggesting that a high mutation rate was required to get chromosomes to jump out of regions of local minima where they were "stuck" due to local optimization carried out by crossover.

8 Discussion

The issue of whether the crossover operator may be detrimental to the performance of a GA is an issue which has been the subject of limited statistical research. There has not as yet been a direct statistical attempt to prove the detrimentality of crossover nor an attempt to describe the conditions under which such detrimentality may occur. In this regard, we undertook the present research to explore the issue of the detrimentality of crossover using a rigorous statistical methodology.

In the first instance the results from our linear-separable test function series, *FNn*, show that crossover is beneficial for these linear-separable problems. This concurs with the suggestion of Salomon that *Crossover's Niche* is in fact linear-separable problems [23].

On the other hand, results from the rotated not-linear-separable test function series demonstrated several instances where crossover was statistically proven to be detrimental. This occurred for not-linear-separable problems which required the highest mutation rates, which in our experience has been a marker for the difficulty of a problem. Thus, what makes a not-linear-separable problem hard for a GA to solve is linked to whether crossover will be detrimental to the performance of the GA solving the problem.

In the course of our research we found that three factors were involved in making a not-linear-separable problem hard for the GA to solve. These were optimization carried out by crossover at the sites of local minima, the bias of the mutation operator and the Hamming Distances for the individual problems.

In the first case, the difficulty of a problem was due to the degree of optimization at local minima carried out by the crossover operator. That is, crossover was carrying out optimization on chromosomes "stuck" in local minima resulting in their moving deeper into the local minima sites. Our experiments on this showed that at least 80% of the time crossover improved the fitness of chromosomes at sites of local minima.

Secondly, we found that the mutation operator was biased along the x and y axes. If a function had at least some of the local minima and the global minimum aligned in the axial directions this made the problem easier to solve as the chromosomes from these minima would be shifted with a greater likelihood towards the global minimum.

Thirdly, the relationship between the problem and the solution space resulted in situations where a less modal problem was actually more difficult to solve because of the greater Hamming Distance between its local minima and the global minimum. This was illustrated for *FN2R45* and *FN3R45* where the latter was the more modal function, yet proved easier to solve as the Hamming Distances between its local minima and its global minimum were lower.

Finally, we were able to demonstrate the detrimentality of crossover on a real world problem, namely, a problem from the Huygens suite. The results showed that crossover can be detrimental on a real world problem. The reasons for this occurring may be extrapolated from the reasons found for the difficult rotated *FNn* problem series. These include the degree of local optimization attributable to the crossover operator and the bias of the mutation operator.

In conclusion, we have been able to demonstrate that crossover is statistically detrimental for the difficult not-linear-separable problems and also the difficult real world problem in the given configuration. Further research will be required to extend the class of problems and illustrate if crossover can be demonstrated to be detrimental with different encodings and in discrete problem domains. However, our results suggest that crossover can prove to have a truly detrimental effect upon GA performance.

Appendices

A ANOVA tables

Table A-1: ANOVA results of *FN1* to *FN3*

<table>
<tr><th colspan="6">Test function <i>FN1</i></th></tr>
<tr><th>Parameter</th><th>Df</th><th>Sum of Sq</th><th>Mean Sq</th><th>F Value</th><th>p-value</th></tr>
<tr><td>Crossover</td><td>6</td><td>211.3841</td><td>35.23068</td><td>102.8543</td><td>0.0000000</td></tr>
<tr><td>Mutation</td><td>8</td><td>195.0530</td><td>24.38163</td><td>71.1810</td><td>0.0000000</td></tr>
<tr><td>Interaction</td><td>48</td><td>12.5655</td><td>0.26178</td><td>0.7643</td><td>0.8678564</td></tr>
<tr><td>Block</td><td>4</td><td>5.7498</td><td>1.43745</td><td>4.1966</td><td>0.0026330</td></tr>
<tr><td>Residuals</td><td>248</td><td>84.9475</td><td>0.34253</td><td>-</td><td>-</td></tr>
<tr><td colspan="6">Residual standard error=0.5852608, Power=87.03%, Threshold=7.</td></tr>
<tr><th colspan="6">Test function <i>FN2</i></th></tr>
<tr><th>Parameter</th><th>Df</th><th>Sum of Sq</th><th>Mean Sq</th><th>F Value</th><th>p-value</th></tr>
<tr><td>Crossover</td><td>4</td><td>79.23721</td><td>19.80930</td><td>66.65568</td><td>0.0000000</td></tr>
<tr><td>Mutation</td><td>8</td><td>91.09235</td><td>11.38654</td><td>38.31421</td><td>0.0000000</td></tr>
<tr><td>Interaction</td><td>32</td><td>9.95044</td><td>0.31095</td><td>1.04631</td><td>0.4066007</td></tr>
<tr><td>Block</td><td>5</td><td>1.74695</td><td>0.34939</td><td>1.17565</td><td>0.3220536</td></tr>
<tr><td>Residuals</td><td>220</td><td>65.38147</td><td>0.29719</td><td>-</td><td>-</td></tr>
<tr><td colspan="6">Residual standard error=0.54515, Power=88.24%, Threshold=7.</td></tr>
<tr><th colspan="6">Test function <i>FN3</i></th></tr>
<tr><th>Parameter</th><th>Df</th><th>Sum of Sq</th><th>Mean Sq</th><th>F Value</th><th>p-value</th></tr>
<tr><td>Crossover</td><td>12</td><td>14002.2</td><td>1166.85</td><td>3.9242</td><td>0.00001088</td></tr>
<tr><td>Mutation</td><td>6</td><td>313701.8</td><td>52283.64</td><td>175.8325</td><td>0.00000000</td></tr>
<tr><td>Interaction</td><td>72</td><td>31744.0</td><td>440.89</td><td>1.4827</td><td>0.01105187</td></tr>
<tr><td>Block</td><td>4</td><td>5179.7</td><td>1294.94</td><td>4.3549</td><td>0.00188308</td></tr>
<tr><td>Residuals</td><td>360</td><td>107045.7</td><td>297.35</td><td>-</td><td>-</td></tr>
<tr><td colspan="6">Residual standard error=17.24381, Power=95.96%, Threshold=7.</td></tr>
</table>

Table A-2: ANOVA results of *FN4* to *FN6*

Test function *FN4*

Parameter	Df	Sum of Sq	Mean Sq	F Value	p-value
Crossover	6	169.0313	28.17188	93.23987	0.0000000
Mutation	8	131.1151	16.38938	54.24359	0.0000000
Interaction	48	10.1115	0.21066	0.69720	0.9329824
Block	4	4.9472	1.23681	4.09345	0.0031292
Residuals	248	74.9318	0.30214	-	-

Residual standard error=0.5496764, Power=87.03%, Threshold=7.

Test function *FN5*

Parameter	Df	Sum of Sq	Mean Sq	F Value	p-value
Crossover	18	5566.06	309.225	46.8718	0.00000000
Mutation	8	18131.18	2266.398	343.5364	0.00000000
Interaction	144	1558.08	10.820	1.6401	0.00002663
Block	4	54.90	13.724	2.0802	0.08175970
Residuals	680	4486.13	6.597	-	-

Residual standard error=2.568512, Power=99.90%, Threshold=7.

Test function *FN6*

Parameter	Df	Sum of Sq	Mean Sq	F Value	p-value
Crossover	18	207154	11509	4.0106	0.000000057
Mutation	8	16671466	2083933	726.2203	0.000000000
Interaction	144	736294	5113	1.7819	0.000001008
Block	4	41181	10295	3.5878	0.006617639
Residuals	680	1951301	2870	-	-

Residual standard error=53.56828, Power=99.90%, Threshold=7.

Table A-3: ANOVA results of *FN1R45* to *FN3R45*

Test function *FN1R45*

Parameter	Df	Sum of Sq	Mean Sq	F Value	p-value
Crossover	8	596.70	74.59	101.9350	<2x10^{-16}
Mutation	10	1551.30	155.13	212.0105	<2x10^{-16}
Interaction	80	54.96	0.69	0.9389	0.6263
Block	4	2.12	0.53	0.7242	0.5758
Residuals	392	286.83	0.73	-	-

Residual standard error=0.8554008, Power=97.02%, Threshold=7.

Test function *FN2R45*

Parameter	Df	Sum of Sq	Mean Sq	F Value	p-value
Crossover	8	691359	86420	30.6658	<2.2x10^{-16}
Mutation	10	7590923	759092	269.3608	<2.2x10^{-16}
Interaction	80	422004	5275	1.8718	4.963x10^{-05}
Block	4	12955	3239	1.1493	0.3329
Residuals	392	1104705	2818	-	-

Residual standard error=53.08601, Power=97.02%, Threshold=6.

Test function *FN3R45*

Parameter	Df	Sum of Sq	Mean Sq	F Value	p-value
Crossover	20	942.53	47.13	11.0612	<2.2x10^{-16}
Mutation	6	2235.96	372.66	87.4686	<2.2x10^{-16}
Interaction	120	844.28	7.04	1.6514	8.505x10^{-05}
Block	4	69.94	17.48	4.1039	0.002742
Residuals	584	2488.14	4.26	-	-

Residual standard error=2.064100, Power=99.71%, Threshold=7.

Table A-4: ANOVA results of *FN4R45* to *FN6R45*

Test function *FN4R45*					
Parameter	Df	Sum of Sq	Mean Sq	F Value	p-value
Crossover	16	1159371	72461	61.5758	$<2.2\text{x}10^{-16}$
Mutation	8	1968603	246075	209.1107	$<2.2\text{x}10^{-16}$
Interaction	128	402189	3142	2.6701	$1.458\text{x}10^{-15}$
Block	4	6601	1650	1.4022	0.2317
Residuals	608	715477	1177	-	-
Residual standard error=34.30410, Power=99.76%, Threshold=5.					
Test function *FN5R45*					
Parameter	Df	Sum of Sq	Mean Sq	F Value	p-value
Crossover	10	756983	75698	84.9871	$<2.2\text{x}10^{-16}$
Mutation	10	3162538	316254	355.0607	$<2.2\text{x}10^{-16}$
Interaction	100	186328	1863	2.0919	$1.301\text{x}10^{-07}$
Block	4	710	178	0.1994	0.9386
Residuals	480	427538	891	-	-
Residual standard error=29.84466, Power=98.86%, Threshold=5.					
Test function *FN6R45*					
Parameter	Df	Sum of Sq	Mean Sq	F Value	p-value
Crossover	20	12649	632	52.6177	$<2\text{x}10^{-16}$
Mutation	12	269824	22485	1870.6825	$<2\text{x}10^{-16}$
Interaction	240	25698	107	8.9080	$<2\text{x}10^{-16}$
Block	4	111	28	2.3052	0.05652
Residuals	1088	13078	12	-	-
Residual standard error=3.466965, Power=100%, Threshold=7.					

Table A-5: ANOVA results of *Landscape_20_101*

Test function *Landscape_20_101*					
Parameter	Df	Sum of Sq	Mean Sq	F Value	p-value
Crossover	10	897763	89776	21.2996	$<2\text{x}10^{-16}$
Mutation	6	11679219	1946536	461.8201	$<2\text{x}10^{-16}$
Interaction	60	1059207	17653	4.1883	$<2\text{x}10^{-16}$
Block	4	33611	8403	1.9936	0.09541
Residuals	304	1281337	4215	-	-
Residual standard error=64.92244, Power=92.65%, Threshold=5.					

B Fitted response curves

Table B-1: Equations of fitted response curves for *FN1* to *FN6*

FN1	Crossover	Final epoch = $56.97715 - 8.15829\mathrm{Cr}$
	Mutation	Final epoch = $81.23346 - 745.06687\mathrm{Mu}$ $+4338.52814\mathrm{Mu}^2$
FN2	Crossover	Final epoch = $56.9028000 - 7.6368889\mathrm{Cr}$
	Mutation	Final epoch = $7.877\mathrm{x}10^1 - 6.652\mathrm{x}10^2\mathrm{Mu}$ $+3.765\mathrm{x}10^3\mathrm{Mu}^2$
FN3	Crossover	Final epoch = $454.9500 - 26.0478\mathrm{Cr}$
	Mutation	Final epoch = $9.540\mathrm{x}10^3 - 1.047\mathrm{x}10^5\mathrm{Mu}$ $+2.999\mathrm{x}10^5\mathrm{Mu}^2$
FN4	Crossover	Final epoch = $51.395690 - 7.307079\mathrm{Cr}$
	Mutation	Final epoch = $6.958\mathrm{x}10^1 - 5.954\mathrm{x}10^2\mathrm{Mu}$ $+3.539\mathrm{x}10^3\mathrm{Mu}^2$
FN5	Overall	Final epoch = $-218.5247 + 16.10332\mathrm{Cr} + 8.586955\mathrm{Cr}^2$ $+11631.9485\mathrm{Mu} - 113700.7892\mathrm{Mu}^2$ $+344700.9038\mathrm{Mu}^3 - 246.3479(\mathrm{Cr} * \mathrm{Mu})$
FN6	Overall	Final epoch = $-3731.3012 + 892.2784\mathrm{Cr} + 237189.8786\mathrm{Mu}$ $-2052110.9896\mathrm{Mu}^2 + 4964206.9821\mathrm{Mu}^3$ $-4941.4196(\mathrm{Cr} * \mathrm{Mu})$

Crossover parameter level (Cr), Mutation parameter level (Mu).

Table B-2: Equations of fitted response curves for *FN1R45* to *FN6R45*

FN1R45	Crossover	Final epoch = $70.410317 - 8.471164\text{Cr}$
	Mutation	Final epoch = $1.048\text{x}10^{2} - 9.756\text{x}10^{2}\text{Mu} + 5.556\text{x}10^{3}\text{Mu}^{2}$
FN2R45	Overall	Final epoch = $3.666\text{x}10^{4} + 3.283\text{x}10^{3}\text{Cr} - 2.811\text{x}10^{5}\text{Mu} + 5.569\text{x}10^{5}\text{Mu}^{2} - 1.174\text{x}10^{4}(\text{Cr} \ast \text{Mu})$
FN3R45	Overall	Final epoch = $1.228\text{x}10^{4} + 2.619\text{x}10^{1}\text{Cr} + 9.058\text{Cr}^{2} - 3.854\text{x}10^{5}\text{Mu} + 4.577\text{x}10^{6}\text{Mu}^{2} - 2.419\text{x}10^{7}\text{Mu}^{3} + 4.801\text{x}10^{7}\text{Mu}^{4} - 2.605\text{x}10^{2}(\text{Cr} \ast \text{Mu})$
FN4R45	Overall	Final epoch = $-1.260\text{x}10^{5} + 2.234\text{x}10^{3}\text{Cr} + 1.203\text{x}10^{6}\text{Mu} - 3.768\text{x}10^{6}\text{Mu}^{2} + 3.906\text{x}10^{6}\text{Mu}^{3} - 5.934\text{x}10^{3}(\text{Cr} \ast \text{Mu})$
FN5R45	Overall	Final epoch = $-6.428\text{x}10^{4} + 1.858\text{x}10^{3}\text{Cr} + 6.774\text{x}10^{5}\text{Mu} - 2.316\text{x}10^{6}\text{Mu}^{2} + 2.602\text{x}10^{6}\text{Mu}^{3} - 5.032\text{x}10^{3}(\text{Cr} \ast \text{Mu})$
FN6R45	Overall	Final epoch = $1.177\text{x}10^{3} + 7.129\text{x}10^{2}\text{Cr} + 5.974\text{x}10^{1}\text{Cr}^{2} - 3.074\text{x}10^{4}\text{Mu} + 3.463\text{x}10^{5}\text{Mu}^{2} - 1.835\text{x}10^{6}\text{Mu}^{3} + 3.845\text{x}10^{6}\text{Mu}^{4} - 1.633\text{x}10^{4}(\text{Cr} \ast \text{Mu}) - 4.103\text{x}10^{2}(\text{Cr}^{2} \ast \text{Mu}) + 1.232\text{x}10^{5}(\text{Cr} \ast \text{Mu}^{2}) - 3.084\text{x}10^{5}(\text{Cr} \ast \text{Mu}^{3})$

Crossover parameter level (Cr), Mutation parameter level (Mu).

Table B-3: Equation of fitted response curve for *Landscape_20_101*

Landscape_20_101	Overall	Final epoch = $2.214\text{x}10^4 + 5.246\text{x}10^3\text{Cr} - 3.141\text{x}10^5\text{Mu}$ $+1.485\text{x}10^6\text{Mu}^2 - 2.285\text{x}10^6\text{Mu}^3$ $-5.009\text{x}10^4(\text{Cr} * \text{Mu}) + 1.196\text{x}10^5(\text{Cr} * \text{Mu}^2)$

Crossover parameter level (Cr), Mutation parameter level (Mu).

C Optimal parameter values

Table C-1: Optimal Parameter Values for *FN1* to *FN6*

FN1	Crossover	100%
	Mutation	8.59%
FN2	Crossover	100%
	Mutation	8.83%
FN3	Crossover	100%
	Mutation	17.45%
FN4	Crossover	100%
	Mutation	8.41%
FN5	Crossover	100%
	Mutation	14.11%
FN6	Crossover	100%
	Mutation	19.47%

Table C-2: Optimal Parameter Values for *FN1R45* to *FN6R45* and *Landscape_20_101*

FN1R45	Crossover	100%
	Mutation	8.78%
FN2R45	Crossover	0%
	Mutation	25.45%
FN3R45	Crossover	33.23%
	Mutation	12.36%
FN4R45	Crossover	0%
	Mutation	35.30%
FN5R45	Crossover	0%
	Mutation	33.38%
FN6R45	Crossover	39.17%
	Mutation	12.97%
Landscape_20_101	Crossover	0%
	Mutation	18.93%

References

[1] A. Czarn, C. MacNish, K. Vijayan, and B. Turlach. Statistical exploratory analysis of genetic algorithms: the influence of gray codes upon the difficulty of a problem. In G Webb and X Yu, editors, *Proc. 17th Australian Joint Conference on Artificial Intelligence (AI 2004)*, number LNAI 3339, pages 1246–1252. Springer, December 2004.

[2] S Bullock. Smooth Operator? Understanding and Visualising Mutation Bias. *Lecture Notes in Computer Science*, 2159:602–612, 2001.

[3] J Cohen. *Statistical power analysis for the behavioral sciences.* Lawrence Erlbaum Associates, Inc., Publishers: New Jersey, 1988.

[4] A Czarn, C MacNish, K Vijayan, and B Turlach. Statistical exploratory analysis of genetic algorithms: the importance of interaction. In *Proc. 2004 Congress on Evolutionary Computation (CEC 2004)*, pages 2288–2295. IEEE Press, June 2004.

[5] A Czarn, C MacNish, K Vijayan, B Turlach, and R Gupta. Statistical exploratory analysis of genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 8(4):405–421, August 2004.

[6] L Davis, editor. *Handbook of genetic algorithms.* Van Nostrand Reinhold, 115 Fifth Avenue, New York, New York 10003, USA, 1991.

[7] K A De Jong. *Analysis of the Behavior of a Class of Genetic Adaptive Systems.* Phd dissertation, Department of Computer and Communication Sciences, University of Michigan, Ann Arbor, MI, 1975.

[8] L Eshelman. Bit-climbers and naive evolution. *Genetic Algorithms Digest*, 5(39), December 1991.

[9] L J Eshelman and D J Schaffer. Crossover's niche. In S Forrest, editor, *Proc. Fifth International Conference on Genetic Algorithms*, pages 9–14, University of Illinois at Urbana-Champaign, July 1993. Morgan Kaufmann Publishers, San Mateo, California.

[10] D B Fogel and J W Atmar. Comparing genetic operators with gaussian mutations in simulated evolutionary processes using linear systems. *Biological Cybernetics*, 63(2):111–114, 1990.

[11] J H Holland. *Adaptation in natural and artificial systems.* The University of Michigan Press, Ann Arbor: United States of America, 1975.

[12] J H Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* University of Michigan Press, Ann Arbor, MI, 1975.

[13] R A Johnson and G K Bhattacharyya. *Statistics: Principles and Methods.* John Wiley & Sons, Inc, 1996.

[14] T Jones. Crossover, macromutation, and population-based search. In L Eshelman, editor, *Proc. of the Sixth International Conference on Genetic Algorithms*, pages 73–80, San Francisco, CA, 1995. Morgan Kaufmann. citeseer.ist.psu.edu/jones95crossover.html.

[15] R I Levin and D S Rubin. *Applied elementary statistics.* Prentice-Hall, Inc, Englewood Cliffs, New Jersey 07632, USA, 1980.

[16] C MacNish. Benchmarking evolutionary and hybrid algorithms using randomized self-similar landscapes. In Tzai-Der Wang, Xiaodong Li, Shu-Heng Chen, Xufa Wang, Hussein A Abbass, Hitoshi Iba, Guoliang Chen, and Xin Yao, editors, *Simulated Evolution and Learning, 6th International Conference, SEAL 2006, Hefei, China, October 15-18, 2006, Proceedings*, volume 4247 of *Lecture Notes in Computer Science*, pages 361–368. Springer, 2006.

[17] C MacNish. Huygens benchmarking suite. . http://ai.csse.uwa.edu.au/cara/huygens, 2007.

[18] M Mitchell, S Forrest, and J Holland. The royal road for genetic algorithms: Fitness landscapes and GA performance. In F J Varela and P Bourgine, editors, *Towards a Practice of Autonomous Systems: Proc. of the First European Conference on Artificial Life, 1991*, pages 245–254, Paris, 11-13 1992. A Bradford book, The MIT Press. ISBN 0-262-72019-1.

[19] D S Moore and G P McCabe. *Introduction to the Practice of Statistics.* W.H. Freeman and Company, New York, fourth edition, 2003.

[20] A Petrovski, A Wilson, and J McCall. Statistical analysis of genetic algorithms and inference about optimal factors. Technical Report 2, The Robert Gordon University, School of Computer and Mathematical Sciences, Faculty of Science and Technology, The Robert Gordon University, Aberdeen, 1998. SCMS Technical Report 1998/2.

[21] C R Reeves and C C Wright. Genetic algorithms and the design of experiments. In L D Davis, K De Jong, M D Vose, and L D Whitley, editors, *Evolutionary Algorithms: IMA Volumes in Mathematics and its Applications*, volume 111, pages 207–226. Springer-Verlag, New York, 1999.

[22] S. Bullock. Are artificial mutation biases unnatural? In *European Conference on Artificial Life*, pages 64–73, 1999.

[23] Ralf Salomon. Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions: a survey of some theoretical and practical aspects of genetic algorithms. *BioSystems*, 39(3):263–278, 1996.

[24] J D Schaffer, R A Caruana, L J Eshelman, and R Das. A study of control parameters affecting online performance of genetic algorithms for function optimization. Technical report, Philips Laboratories, North American Philips Corporation, 345 Scarborough Road, Briarcliff Manor, New York 10510, 1989.