

Applying Price's Equation to Survival Selection

Jeffrey K. Bassett
George Mason University
Krasnow Institute, MS2A1
Rockfish Creek Lane
Fairfax, VA 22030
jbassett@cs.gmu.edu

Mitchell A. Potter
Naval Research Laboratory
4555 Overlook Ave., S.W.
Washington, DC 20375
mpotter@aic.nrl.navy.mil

Kenneth A. De Jong
George Mason University
Computer Science, MS4A5
4400 University Dr.
Fairfax, VA 22030
kdejong@gmu.edu

Track: Genetic Algorithms

ABSTRACT

Several researchers have used Price's equation (from biology theory literature) to analyze the various components of an Evolutionary Algorithm (EA) while it is running, giving insights into the components contributions and interactions. While their results are interesting, they are also limited by the fact that Price's equation was designed to work with the averages of population fitness. The EA practitioner, on the other hand, is typically interested in the best individuals in the population, not the average.

In this paper we introduce an approach to using Price's equation which instead calculates the upper tails of population distributions. By applying Price's equation to EAs that use survival selection instead of parent selection, this information is calculated automatically.

1. INTRODUCTION

Users of Evolutionary Algorithms (EAs) have a limited set of tools and methodologies available with which to understand the effects of their design decisions. Most researchers today still rely on best-so-far curves and t-tests to compare the results of using different components in the algorithm. New tools are needed in order to provide a more principled approach to making better design decisions.

One approach that some have explored is to use theory from the evolutionary biology literature. In particular, some have used Price's equation [10] as an inspiration for building tools to analyze running EAs [1, 7, 9, 4].

Evolutionary biologists tend to be interested in the average values of various attributes in the population, such as gene frequency. The EA community, on the other hand, tends to be less interested in averages, and more interested in the best individuals. Hence, the extensive use of best-so-far curves. This makes biology theory limited in its value when applied to EAs.

In order for these theories to be more applicable to EAs,

they need to provide ways to analyze the upper tails of the the various probability distributions we deal with, such as population fitness, instead of just the means. Some researchers have tried to address this issue by modifying Price's equation to calculate higher cumulants, such as the standard deviation [4]. While these results are interesting, they seem limited to providing only a very subjective view of what is happening in the upper tails of the distributions. In order to make this approach useful, we need to achieve more objective results. In this paper, we present a method that takes us one step closer to this goal.

In most previous work using Price's equation, researchers have applied the equation to algorithms assuming that selection always precedes the genetic operators. This is referred to as parent selection, and is a commonly used approach in EAs. We propose instead to apply Price's equation to EAs in which selection follows the genetic operator instead of preceding them. This is typically called survival selection. As it turns out, any EA can be considered to be using either parent selection or survival selection just by changing one's "frame of reference". This will be described in more detail in the paper.

When we apply Price's equation in this way, selection acts as a filter which clears away many of the damaging effects of the genetic operators from the calculations. As a result, the terms in Price's equation end up representing the upper tails of the distributions instead of the means.

One reassuring aspect of this filtering effect is that it more closely resembles the notion of fitness in biology. Specifically, an individual's fitness is determined by the number of descendants it gives rise to in future generations.

In the following sections we discuss similar work in the field, followed by a description of our methodology. In section 4 we presents results of using our approach on some example problems. And finally we conclude with a discussion of future work.

2. BACKGROUND

2.1 Price's Equation

In his efforts to understand selection, Price developed the well know equation that bears his name [10]. Ultimately his goal was to create a general theory of selection that could be applied across a variety of disciplines, including biology and economics. To demonstrate the use of his equation, he first applied it to evolutionary biology where it was used to describe and predict the change in gene frequency from one generation to another.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO 2005 Washington, DC USA

Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...\$5.00.

In particular, Price noticed that a covariance relationship exists between the number of successful offspring that an individual produces (the biological notion of fitness) and the frequency of any given gene in that individual. If this covariance value is high, then the existence of that gene is a good predictor of selection.

The other important contribution that his equation offers is that it separates the average effects of selection from the average effects of the genetic operators. Price generally assumed that the effects of the operators would be close to zero, and would consequently drop this term from his equation. As we will see though, this is often not a good assumption, at least when dealing with an evolutionary algorithm.

The most common formulation of Price’s equation is

$$\Delta Q = Q_{t+1} - Q_t = \frac{Cov(z, q)}{\bar{z}} + \frac{\sum z_i \Delta q_i}{N\bar{z}}, \quad (1)$$

where ΔQ is the change from one generation to the next in some measurable attribute q of a population, such as the number of occurrences of a particular gene or combination thereof, Q_t and Q_{t+1} are the attribute means taken over all members of the populations of parents and children respectively, z_i is the number of children produced by the i th parent, \bar{z} is the mean number of children produced, Δq_i is the difference between the average q value of the children of i and i ’s own q value, and N is the size of the parent population. The covariance term in equation 1 represents the portion of ΔQ attributable to selection, and the summation term represents the portion attributable to the genetic operators.

2.2 Applications in EC

The most direct application of Price’s work in evolutionary computation involves situations in which the attribute being measured is gene frequency. For example, Langdon and Poli show how measuring gene frequencies is equivalent to determining the frequency of use of the available primitives in the solution trees being evolved with genetic programming [7]. They were able to use this information to diagnose the probable causes of poorer performing runs.

Altenberg applied Price’s equation to a Vose-style model of an evolutionary algorithm [2, 14]. With this he was able to re-derive Holland’s schema theorem [6], thus creating a stronger theoretical connection between Vose’ and Holland’s work. He also demonstrated that gene frequency is not the only attribute of the individuals which can be measured. He identified several different measurement functions which could be useful, including mean fitness from both the biological and evolutionary computation perspectives, frequency of schemata, and evolvability.

Potter et al. used Price’s equation to visualize EA dynamics as they relate to the evolvability of a population [9]. Of particular interest was the balance between the exploration and exploitation of the various genetic operators. The second term in equation 1 measures the combined effects of all the genetic operators. In order to make Price’s equation more applicable to the visualization of an EA, the second term was extended to separate the effects of each genetic operator so they could be viewed independently. The result

is the following equation:

$$\Delta Q = \frac{Cov(z, q)}{\bar{z}} + \sum_{j=1}^k \frac{\sum z_i \Delta q_{ij}}{N\bar{z}}, \quad (2)$$

where k is the number of genetic operators, q'_{ij} is the average value of the measurement function applied to the children of i after the application of operator j , and $\Delta q_{ij} = q'_{ij} - q'_{i(j-1)}$.

Bassett et al. further extended Price’s equation to calculate the second central moment (variance) of the measured population attribute to get a better sense of how often an operator creates the above average individuals that enable the population to continue to improve over time [4]. From equation 1 the random variable of interest is $X = q_{ik} - q_i$, where q_{ik} is the measured q of the k th child of parent i . This gives the following variance equation:

$$Var[X] = \frac{\sum_{i=1}^N \sum_{k=1}^{z_i} \left(q_{ik} - q_i - \frac{\sum_{i=1}^N z_i \Delta q_i}{N\bar{z}} \right)^2}{N\bar{z}}. \quad (3)$$

A similar variance equation can be derived from equation 2 using the random variable $X = q_{ijk} - q_{i(j-1)k}$, where q_{ijk} is the measured q of the k th child of parent i after the application of operator j .

2.3 Statistical Models of EAs

Our work with Price’s equation has a strong statistical component, and so a comparison with other statistical models of EAs is warranted. One such model was developed by Prügel-Bennett and Shapiro, and is based on statistical mechanics theory [11]. The state of the EA is represented in each generation by a set of cumulants of a population’s fitness distribution. Using maximum entropy distributions, equations of motion can then be derived, providing a method to infer the change in the cumulants from one generation to the next. This approach has proved successful in making reasonably accurate predictions of EA behavior on some moderately difficult problems. It can also predict effects where other models typically fail, such as predicting the effects of finite population sizes.

Mühenbein developed another statistical model of EAs based upon the methods and theories used in livestock breeding [8]. In his Breeder Genetic Algorithm (BGA), the “response to selection” equation [5] is used to build a predictive model of the GA’s behavior. Using this model, Mühenbein is able to analyse a variety of operators to determine which will be most effective. It is interesting to note that Mühenbein also uses Price’s equation as a way to estimate the response to selection.

In general, theory has two purposes: to predict, and to explain. In the two cases just described, the authors built an iterative model which could predict the behavior of a running EA. Our work differs from theirs in that we are focusing using theory to explain the behavior of an algorithm, and not to do prediction.

3. METHODOLOGY

3.1 Shifting the Frame of Reference

One way to view an evolutionary algorithm is as a set of operators applied in turn to a series of intermediate populations. As is shown in figure 1, selection is applied to the

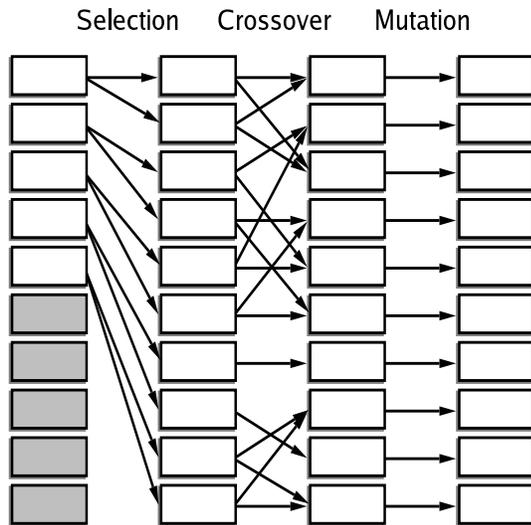


Figure 1: Intermediate populations in an EA using parent selection.

parent population to create a new intermediate population. Crossover is then applied to this intermediate population to create another intermediate population. And finally, mutation is applied to create the next generation. The gray boxes represent individuals that do not survive into the next generation.

In a Genetic Algorithm (GA) the operators are typically placed in the order just described: selection, crossover, mutation. Since selection is applied to the parent population, it is sometimes referred to as parent selection. An Evolution Strategies (ES), on the other hand, tends to use the order: crossover (maybe), mutation, then selection. When selection is performed after the genetic operators, it is often referred to as survival selection.

Bäck has shown that these two selection approaches are equivalent [3]. The only real difference comes at the very beginning and end of the run, at which point there may be one extra, or one missing selection operation, depending on how you view things. One can view these two selection schemes as simply having a slightly different frame of reference as shown in figure 2. Or more accurately, all EAs can be viewed from either frame of reference.

Most applications of Price's equation to EAs have been implemented using the parent-selection frame of reference. This is well matched with the intuitive notion of what the terms in Price's equation mean. For example, the covariance term in the equation, which Price claimed to be the selection component, calculates the difference between the entire parent population and some subset of that population (perhaps with some members duplicated). This gives the impression that selection is expected to operate on the parent population. In fact, if one were to measure the differences between the average q values of the intermediate populations in figure 1, they would match exactly with the terms in Price's equation.

One might expect Price's equation to have the same behavior when applied in a survival-selection frame of reference, but this is not the case. The property described above

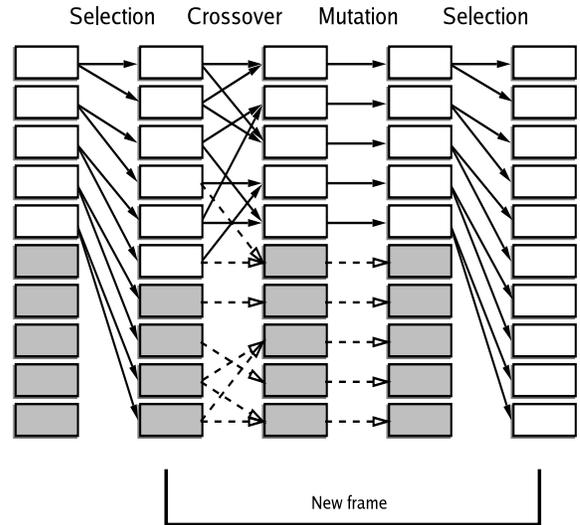


Figure 2: Intermediate populations in an EA using survival selection.

no longer holds. The terms of Price's equation no longer match the differences between the average q values of the intermediate populations. This does not mean that Price's equation is invalid, because the resulting ΔQ is still correct. So the question here is, what do the terms of Price's equation mean, and are they useful?

To answer this question, we need to take a closer look at what is actually being calculated. One important aspect of the notation for Price's equation that is not immediately obvious is that an association between parents and their children is built in. Specifically, when we apply Price's equation to an EA using parent selection, Δq_i in equation 1 represents the difference between a parent's q value and the average q value of all of its children, including the ones that are not viable.

In a parent-selection frame of reference, what ends up happening is that all children that are ever produced end up being associated with their parents. But in a survival-selection frame of reference, some children which get produced in the intermediate populations end up not surviving into the next generation. When performing the association between parents and children, these failed intermediate children are lost and not considered, as illustrated by the shaded boxes and dashed lines in figure 2. The equation still balances out though, with these lost children being accounted for in the covariance term. Furthermore, this is more consistent with the biological notion of successful offspring, which are most likely the only offspring Price intended to measure.

The net effect of this is that survival selection acts as a filter, weeding out the poorer fitness individuals from the calculations for the operators. So instead of representing the overall effects of the operators, the terms in Price's equation now represent only the beneficial effects of each operator. Or perhaps it is more accurate to say that they represent only the effects of each operator that did not end up killing the child. As a result, the terms which calculate the change in fitness due to the genetic operators now gives us the mean of what is essentially the upper tail of the distributions. These

calculations should give us a better indication of what each operator is actually contributing to the search process.

3.2 Example Problem

To demonstrate the utility of using the survival-selection frame of reference with Price’s equation, we have chosen a standard problem from the function optimization literature introduced by Schwefel [13]. The objective function

$$f(\vec{x}) = \sum_{i=1}^n x_i \sin\left(\sqrt{|x_i|}\right)$$

defines a landscape covered with a lattice of large peaks and basins. The predominant characteristic of the Schwefel function is the presence of a second-best maximum far away from the global maximum, intended to trap optimization algorithms on a suboptimal peak. The best maximums are near the corners of the space. In this formulation of the problem, the global maximum is $418.9829n$. In our experiments the problem has thirty independent variables constrained to the range $(-500.0, 500.0)$, which means the global maximum has a value of 12569.487 .

3.3 Algorithm Specifications

The Evolutionary Algorithm we have chosen to analyze uses a real-valued representation. We use a generational model, meaning that there is no overlap between parent and child populations. The operators used are truncation selection in which the top 20 percent of the offspring survive, uniform crossover at 100 percent, and Gaussian mutation with either fixed or adaptive sigma values, depending on the experiment.

We chose a 100 percent crossover rate because a lower crossover rate would have some side-effects that we wanted to avoid. Specifically, when crossover is not performed, a clone is created and mutated. This means that we would be analyzing two pairs of operators; one that produces children by crossover followed by mutation, and another that produces children by cloning followed by mutation. In the future we plan to explore the ramifications of this in more detail.

Gaussian mutation was chosen for a similar reason, since in most implementations (including ours) it is applied to all genes in the genome. Thus it is essentially used 100 percent of the time, avoiding the issue mentioned above. In situations where the sigma values of the mutation operator are adapted, we store a separate sigma value for each gene, and adapt them using the approach described in [3].

All experiments use a population size of 100. While this is generally acknowledged to be too small to find the global optimum much of the time [12], it is sufficient for our purposes here.

4. RESULTS

To evaluate this new approach to using Price’s equation, we will apply it in a series of four experiments. Our goal is to see if the results give us any more information than already established approaches. In all experiments we use the EA described above to solve the Schwefel function, and average results over 100 runs of the EA.

4.1 Experiment 1

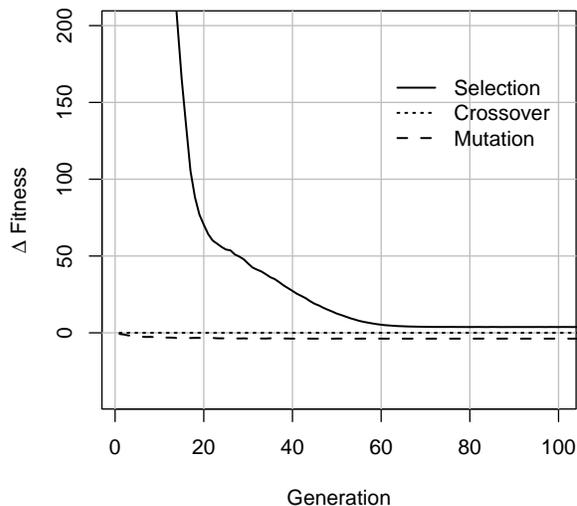


Figure 3: Price plot from experiment 1 using parent-selection frame of reference.

In our first experiment, the gaussian mutation operator is configured with a fixed standard deviation of 1.0. Figure 3 shows a Price plot generated using the parent-selection frame of reference. The q value that we are measuring here is fitness, and we plot the average change in fitness as a result of using each one of the operators: selection, crossover and mutation.

In general, unless a genetic operator is highly tuned to a particular search space, we would expect it to reduce the fitness of an individual at least as often as it improves it. As a result, we can expect lines for crossover and selection to be at or below the zero line almost all of the time. Given what is being used to calculate these values, we conclude that this plot gives us an indication of how disruptive (on average) each operator is at any given time during the run. While this information is useful, it does not answer the questions we are really interested in. Specifically, how much are the operators contributing to the evolvability of the EA? We might infer that disruptive operators are not helpful, but this could be wrong. They might on occasion create individuals which are very good, and this might be critical to the search process. A notion we refer to as “constructive diversity”.

An approach that has been used in an attempt to answer the question of operator contribution has been to extended Price’s equation to calculate the standard deviations of the ΔQ components in addition to the means [4]. Figures 4 and 5 plot these standard deviations for both the crossover and mutation operators. The figures show that crossover is creating much greater improvements to individuals than the mutation operator, and hence is contributing more to the evolvability of the population. But again, these results are not completely satisfying. Knowing that the potential is there is helpful, but we would also like to have a clear idea of what is actually happening.

Figure 6 shows a Price plot for the same algorithm and problem, but using the survival-selection frame of reference.

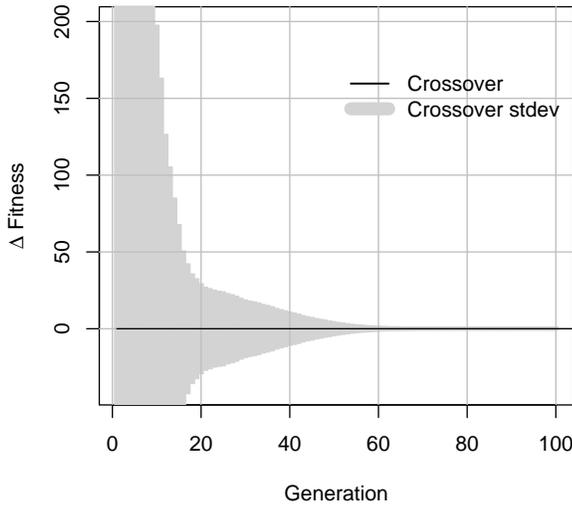


Figure 4: Standard deviation of crossover component of Δ fitness from experiment 1 using parent-selection frame of reference.

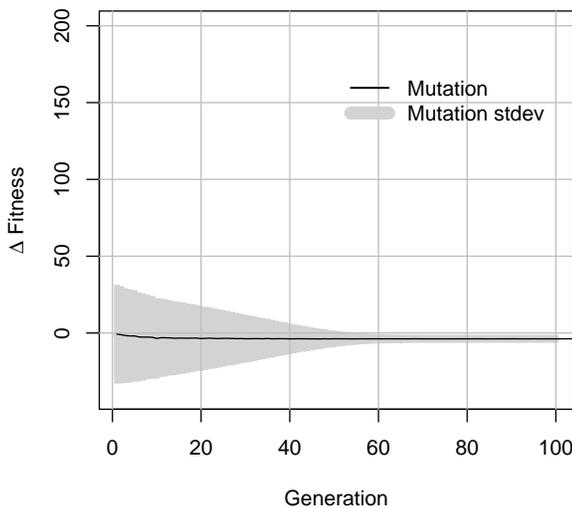


Figure 5: Standard deviation of mutation component of Δ fitness from experiment 1 using parent-selection frame of reference.

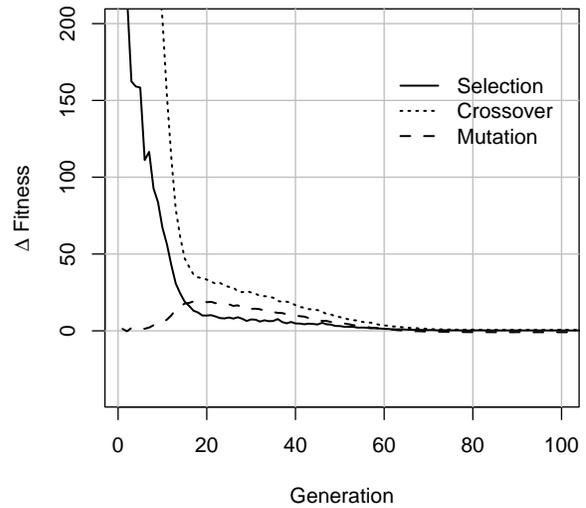


Figure 6: Price plot from experiment 1 using survival-selection frame of reference.

Now we see that the crossover and mutation lines have moved up above the zero mark, indicating that they are showing the contribution of just the viable individuals, and weeding out the contribution of the poorer individuals that did not survive.

It is interesting to observe the effects of mutation in this plot. At first mutation seems to contribute nothing. This is probably because crossover is contributing so much to the evolvability that the effects of mutation are negligible, and therefore washed out. But as the population converges, and the effects of crossover diminish, the effects of mutation become more prominent, and affecting the likelihood of and individual being selected. This causes the line for mutation to rise up and almost meet the crossover line.

4.2 Experiment 2

Next we examine what happens when we switch the mutation operator to an adaptive gaussian mutation. In figure 7 we plot the terms of Price's equation from a parent-selection frame of reference. From this we can see that the mutation function is very disruptive. But does this mean that it is ineffective?

In figures 8 and 9 we again plot the standard deviations of the crossover and mutation components of the ΔQ fitness distribution. As in experiment 1, we see that the upper tail of the crossover distribution extends higher than the upper tail of the mutation distribution. Therefore, it seems likely that crossover is contributing more to evolvability than mutation.

This is corroborated by the survival-selection plot shown in figure 10. Note that the scale on the y-axis has been changed in order to make this plot more readable.

4.3 Experiment 3

Next, we examine the effects of modifying the problem domain by performing a random coordinate rotation along

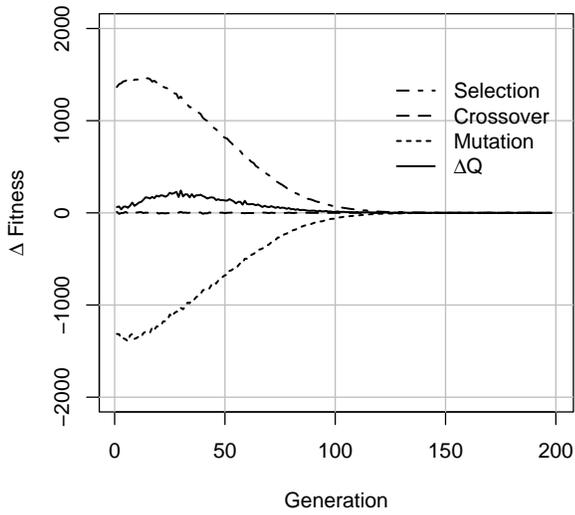


Figure 7: Price plot from experiment 2 using parent-selection frame of reference.

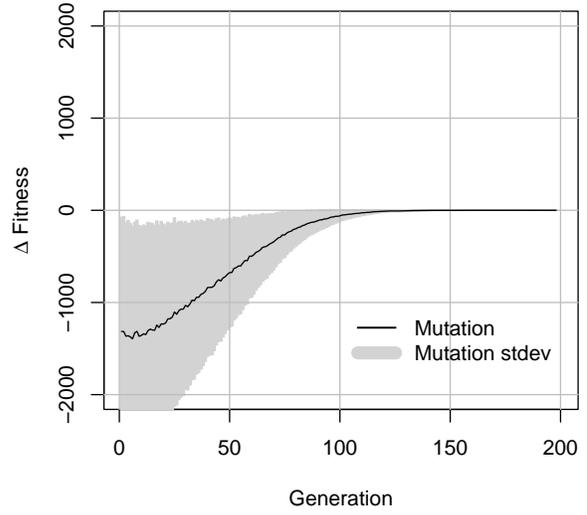


Figure 9: Standard deviation of adaptive mutation component of Δ fitness from experiment 2 using parent-selection frame of reference.

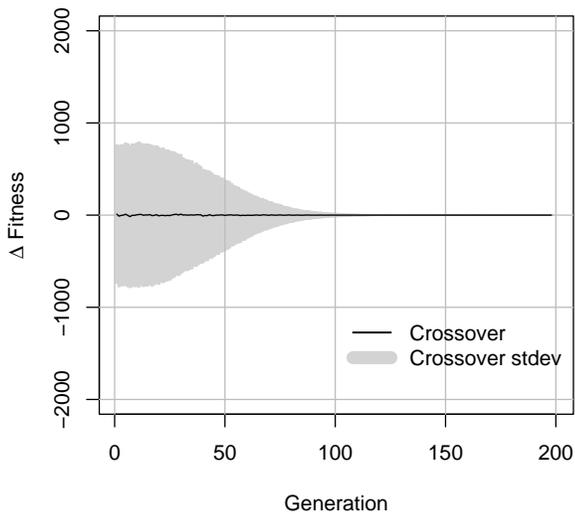


Figure 8: Standard deviation of crossover component of Δ fitness from experiment 2 using parent-selection frame of reference.

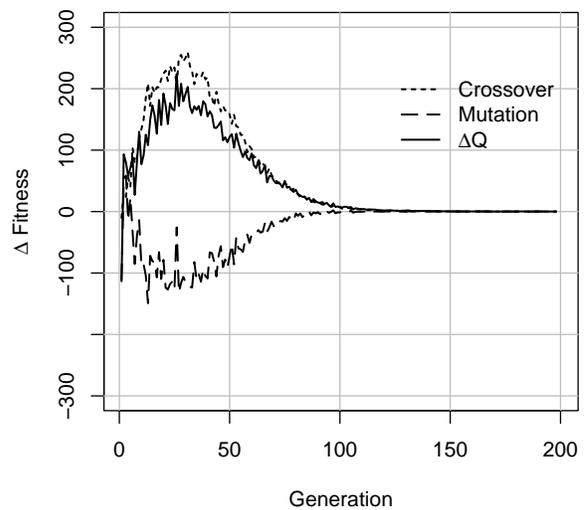


Figure 10: Zoomed Price plot from experiment 2 using survival-selection frame of reference.

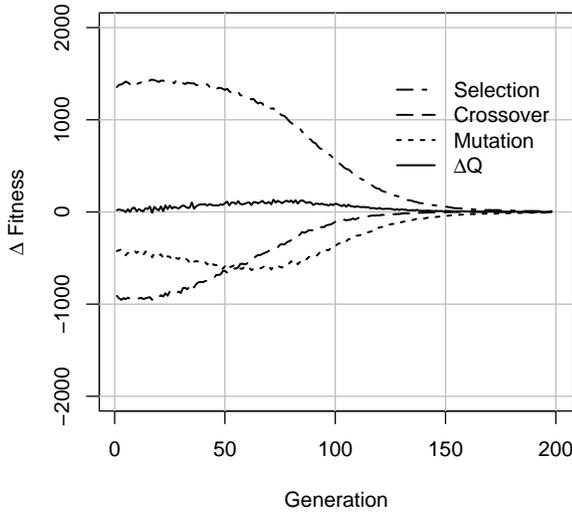


Figure 11: Price plot from experiment 3 using parent-selection frame of reference.

each axis of the landscape. Salomon [12] demonstrated that several of the standard GA genetic operators are biased so that they perform particularly well when the fitness landscapes are decomposable. In other words, the fitness function is an aggregation of smaller functions, each applied to just one gene. This means that there are no interdependencies between the genes, and thus no epistasis. Salomon demonstrated that by rotating such landscapes, they lost the property of decomposability.

The unrotated Schwefel function is decomposable and our crossover operator has a strong bias towards this type of landscape, so we would expect rotation to have a negative effect on crossover. The gaussian mutation operator, on the other hand, does not have the same type of bias, so we would expect it to be much less effected by rotation. When we look at the parent-selection plot in figure 11, we see that the crossover operator has indeed become much more disruptive. But how much are crossover and mutation now contributing to the search?

The standard deviation plots for the two operators (see figures 12 and 13) show that the distributions are very similar, with the upper tail of the mutation operator going a little higher and the lower tail of the crossover operator going a little lower. Although, any interpretation of these results is likely to be subjective, we will say that mutation seems to be contributing more to evolvability than crossover.

The survival-selection plot in figure 14 seems to verify the interpretation given above, showing that mutation is contributing more to evolvability than crossover.

4.4 Experiment 4

The EA used in all of the previous experiments applies crossover followed by mutation. In this final experiment we reran the first three experiments with the order of the crossover and mutation operators swapped (i.e., children are produced by applying mutation followed by crossover) to see

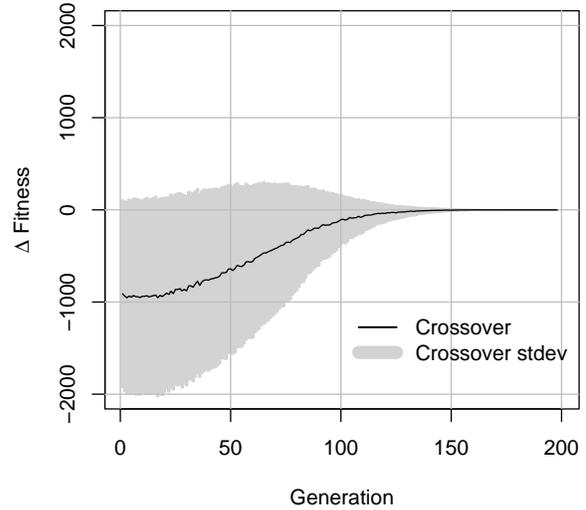


Figure 12: Standard deviation of crossover component of Δ fitness from experiment 3 using parent-selection frame of reference.

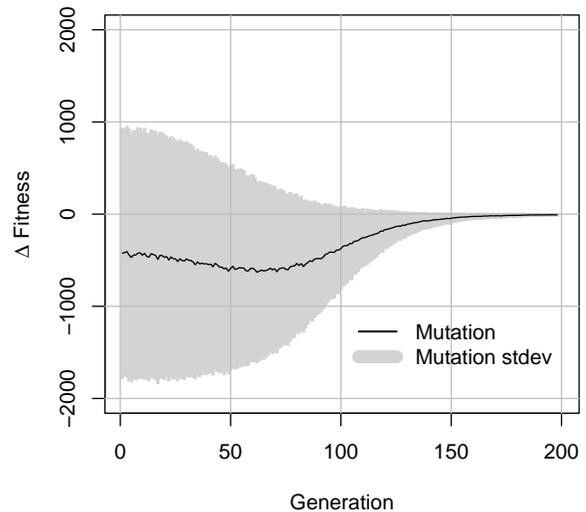


Figure 13: Standard deviation of mutation component of Δ fitness from experiment 3 using parent-selection frame of reference.

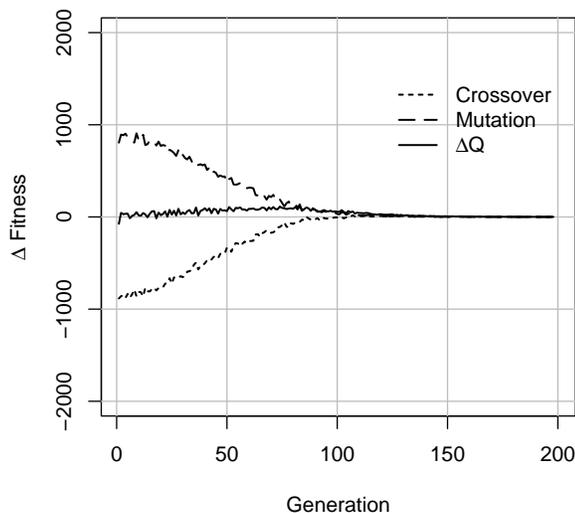


Figure 14: Price plot from experiment 3 using survival-selection frame of reference.

if they would achieve the same results, at least qualitatively.

In the first two experiments, the results were consistent and the survival-selection plots looked roughly the same. In experiment 3 though, we saw something different. The plot (not shown here) looked to be almost the exact opposite of figure 14. Crossover seemed to be the operator contributing most to evolvability, and the line for mutation fell well below zero.

We surmise that in this case, since both operators are fairly disruptive on average, the first one damages most of the individuals, creating children that are lower down on the peaks. This creates an opportunity for the second operator to repair this damage, and move the children back up the peaks to higher fitnesses. In the end though, the overall behaviors of the algorithms are the same.

5. CONCLUSIONS AND FUTURE WORK

If tools like Price's equation are going to be applicable to the work we do in evolutionary computation, they will have to provide some information about the upper tails of distributions, and not just the means. This work has successfully shown one approach to doing this.

While these results are not completely free of ambiguity, they do provide a less subjective method for evaluating what contribution an operator makes to the evolvability of an EA than some of the previous research. An important next step in this research is to make sure we can apply statistical tests to our results so that we can make assertions with some degree of confidence.

It is important to note that this work has focused exclusively on the genetic operators, and ignored the meaning of the covariance (or selection) term in Price's equation. We suspect that in a survival selection frame of reference, this term instead becomes a measure of contribution due to inheritance. In the future work we would like to explore this

further.

Our ultimate goal is to extend these tools to other representations and operators which are not as well understood. The real power of these tools can be realized when designing new representations and operators.

6. REFERENCES

- [1] L. Altenberg. The evolution of evolvability in genetic programming. In K. E. Kinneer Jr., editor, *Advances in Genetic Programming*, chapter 3, pages 47–74. MIT Press, 1994.
- [2] L. Altenberg. The schema theorem and Price's theorem. In L. D. Whitley and M. D. Vose, editors, *Foundations of Genetic Algorithms III*, pages 23–49. Morgan Kaufmann, 1995.
- [3] T. Bäck and H.-P. Schwefel. An overview of evolutionary algorithms for parameter optimization. *Evolutionary Computation*, 1(1):1–23, 1993.
- [4] J. K. Bassett, M. A. Potter, and K. A. De Jong. Looking under the EA hood with Price's equation. In *Proceedings of the 2004 Genetic and Evolutionary Computation Conference (GECCO 2004)*. Springer, 2004.
- [5] R. A. Falconer. *Introduction to Quantitative Genetics*. Longman, London, 1981.
- [6] J. H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1975.
- [7] W. B. Langdon and R. Poli. *Foundations of Genetic Programming*. Springer-Verlag, Berlin Heidelberg, 2002.
- [8] H. Mühlenbein. The equation for response to selection and its use for prediction. *Evolutionary Computation*, 5(3):303–346, 1998.
- [9] M. A. Potter, J. K. Bassett, and K. A. De Jong. Visualizing evolvability with price's equation. In *Proceedings of the 2003 Congress on Evolutionary Computation*, pages 2785–2790. IEEE, 2003.
- [10] G. Price. Selection and covariance. *Nature*, 227:520–521, 1970.
- [11] A. Prügel-Bennett and J. L. Shapiro. An analysis of genetic algorithms using statistical mechanics. *Physical Review Letters*, 72:1305–1309, 1994.
- [12] R. Salomon. Reevaluating genetic algorithm performance under coordinate rotation of benchmark functions. *BioSystems*, 39(3):267–278, 1996.
- [13] H.-P. Schwefel. *Numerical optimization of Computer models*. John Wiley & Sons, Ltd., Chichester, 1981.
- [14] M. D. Vose. *The Simple Genetic Algorithm*. MIT Press, 1999.