

Understanding EA Dynamics via Population Fitness Distributions

Elena Popovici
epopovic@gmu.edu
& **Kenneth De Jong**
kdejong@gmu.edu

Department of Computer Science
George Mason University
Fairfax, VA 22030

Abstract. It is clear from the study of complex non-linear systems in general, and evolutionary algorithms (EAs) in particular, that there is no single analysis tool or technique capable of providing a reasonably comprehensive understanding of the behavior of a complex non-linear system. Rather, we continue to develop sets of tools that collectively provide a more comprehensive view than any one tool. In this paper we provide an initial assessment of a new technique that is based on an old idea: the importance of understanding how the fitness distribution of an EA population changes over time.

1 Introduction and Background

Although a sense of the importance of population fitness distributions is deeply ingrained in the mathematical biology community, it has generated much less interest in the evolutionary computation (EC) community, perhaps because of the strong focus on solving optimization problems for which techniques involving "best-so-far" curves or "expected first encounter" times seem quite natural and helpful. However, several groups including [1, 8, 9] have shown how deeper insights into EA behavior can be obtained by focusing on the changes to the entire population fitness distribution rather than just the tails of these distributions (which is typically the focus of optimization-oriented analyses). A key element in these approaches is the ability to characterize how repeated applications of selection and reproduction modify the population fitness distributions over time. So far, this has proved to be extremely difficult to achieve analytically and has been done successfully for only a few very specialized EAs and/or very simple fitness landscapes.

Closely related to this work is the research on operator fitness correlation [4–6] and the research on operator fitness distributions [2]. Here the focus is on understanding how the fitness of a child produced by typical reproductive operators differs from the fitness of its parent(s). With sufficiently precise characterizations of this type along with knowledge about selection and the fitness

distribution of the current population, predictions can be made about the expected fitness distribution of the next generation. However, such characterizations are difficult to obtain in general for several reasons, including 1) the fact that most fitness landscape are not isotropic and hence the fitness correlation of operators is a dynamic function of where an EA is in the search space, and 2) the fitness correlation of reproductive operators such as recombination depends on the properties of the parents in the current population and hence can change dramatically as evolution proceeds.

With these known difficulties in adopting a purely analytical approach, we were struck by an empirically derived series of snapshots of the population fitness distribution of a fairly vanilla GA in [9], presented in the beginning of that paper to motivate the analytical work. Our reaction was that, independent of whether a formal analysis was possible, these empirically derived fitness distribution snapshots by themselves had the potential for providing interesting and useful insights into the behavior of EAs. And so, we began to explore a number of issues related to this possibility including characterizing:

- the properties of evolving population distributions.
- the effects of particular EA design choices on evolving population fitness distributions.
- the effects of fitness landscapes on evolving population fitness distributions.

In the remainder of this paper we describe in more detail our methodology and present some of our initial findings.

2 A Population Fitness Distribution Analysis Methodology

The methodology we have developed consists of two phases: 1) collecting population fitness distribution data from multiple runs of real EAs, and 2) analyzing the collected data in several different ways to provide useful insights into EA behavior. In that sense our approach is quite general and can be applied to any EA and to any fitness landscape.

2.1 Collecting and Visualizing Population Fitness Distribution Data

We begin by instrumenting an EA to provide snapshots of the population fitness data at designated points during the evolutionary process, and typically display them in a histogram-like manner as illustrated in Figure 1. Since the instrumented EAs are stochastic with finite populations, even fitness distribution snapshots taken at generation 0 can vary from run to run. By using line-oriented histograms rather than the bar-oriented versions, one can plot the histograms of multiple snapshots on a single graph for comparison purposes. Finally, one can expose general trends by averaging a set of line histograms taken at the same point in time but collected across multiple independent EA runs.

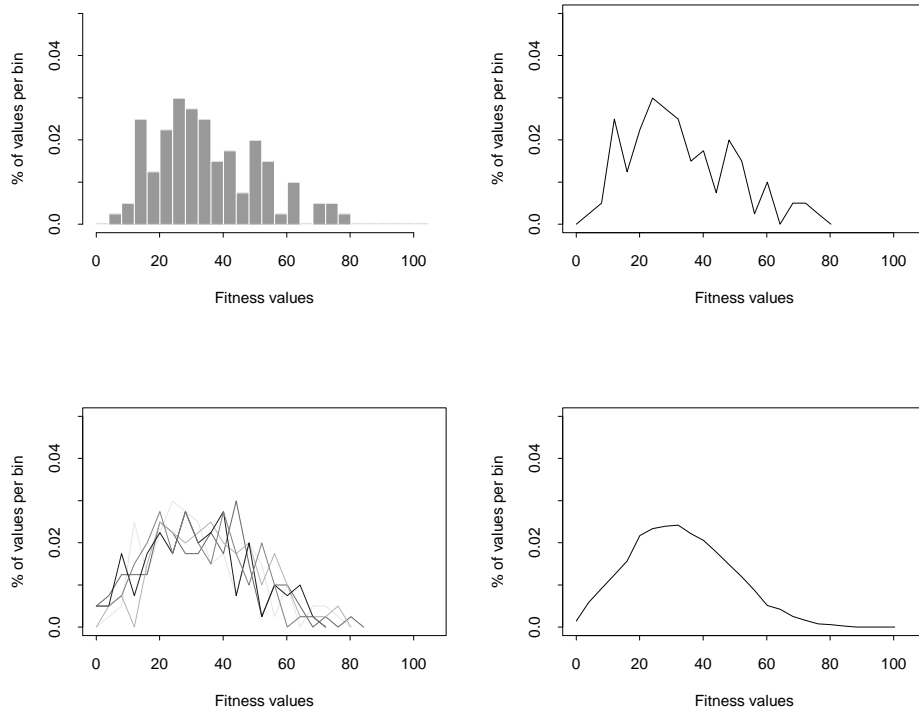


Fig. 1. Histogram types used: top-left: bar-oriented, 1 run; top-right: line-histogram, the same 1 run; bottom-left: super-imposed line histograms for 5 runs; bottom-right: histogram obtained by averaging each bin over 100 runs. Plotted data represents randomly generated initial populations of size 100 on De Jong’s F1 function.

These histogram averages can then be collected and compared at various points in an evolutionary process to provide insight into EA behavior over time. For example, in [9] this technique was used to visualize successive snapshots at generations 0, 10, 20, 30 and 40 of the population fitness distribution of a generational GA being applied to an energy minimization problem. The result was an intuitively appealing picture of a Gaussian-looking fitness distribution whose mean and variance were being reduced over time. Figure 2 illustrates these ideas on a simpler landscape, namely function F1 of the De Jong test suite.

What is plotted are the histogram averages of population fitness distribution snapshots taken every 7th generation starting with generation 1. The rightmost histogram corresponds to generation 1, and the remaining cluster of histograms correspond to generations 8, 15, ... Comparing this figure with the one presented in the Shapiro paper [9] already hints at the answers to some of our questions.

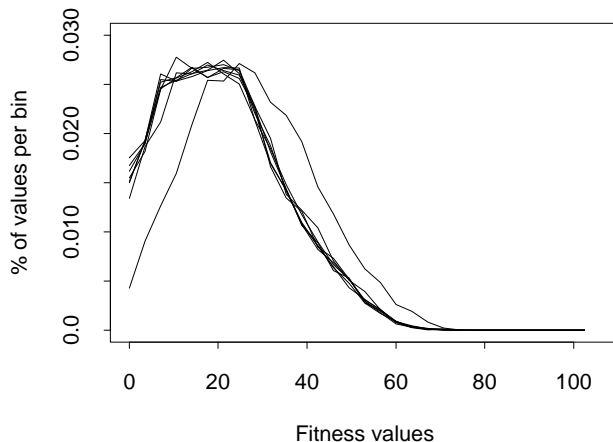


Fig. 2. Histogram averages of population fitness produced by a GA on F1.

The more complex energy minimization landscape produced more Gaussian-looking histograms than what we see on F1. Also, the GA in Shapiro study used a more aggressive Boltzmann selection mechanism than the binary tournament selection mechanism used in Figure 2. As a result, unlike the Shapiro study, the population fitness distributions in Figure 2 reach a steady state rather quickly.

2.2 Quantitative Analysis of Population Fitness Distributions

Although visualizing population fitness distributions can be quite insightful, the collected data allows us to go a step further and perform a more quantitative analysis of the observed fitness distributions. In particular, it is possible to estimate how likely the empirically generated distributions reflect an underlying standard distribution (e.g, normal, exponential, etc.). There is a standard statistical technique for doing this involving the generation of “Q-Q” plots which can be used to compute an R^2 value that represents the likelihood that an empirically generated distribution is due to a particular theoretical distribution [3]. By generating Q-Q plots for a variety of standard theoretical distributions, the corresponding R^2 values provide a quantitative estimate of which, if any, of the standard distributions are a good fit to the observed fitness distribution data.

So, for example, applying these techniques to the distributions in Figure 2 allows one to make quantitative statements regarding how ‘Gaussian-like’ they really are. Such quantitative analyses have potentially important implications for a deeper understanding of EA dynamics. In particular, in order to keep

the mathematics tractable, much of the theoretical work in this area makes simplifying assumptions concerning the form of population distributions (e.g., Gaussian). The quantitative analysis described here provides the ability to assess the validity of such assumptions.

In our preliminary studies we have been comparing our empirical data to 9 plausible theoretical distributions. Three of them are discrete: binomial, negative binomial and poisson distributions. The other six are continuous: uniform, normal, log normal, exponential, gamma and Pareto.

The R^2 value corresponding to a set of fitness values and a theoretical distribution is obtained through a (rather complex) procedure described in the remainder of this section for completeness, but can be safely skipped over by a first-time reader.

First, there is a test to see if there is still some variance in the population fitness values. The coefficient of variance is computed. If it is smaller than a lower threshold (0.05 was used), or if the the number of distinct fitness values in the set is < 4 , then the data set can be considered almost constant. In this case, it does not make sense to construct Q-Q plots for any of the theoretical distributions.

The theoretical distributions are in general parameterized and their shape may change according to these parameters. This can affect the shape of the Q-Q plot and also the R^2 value obtained from the Q-Q plot. Therefore, an attempt is made to tune the parameters of each theoretical distribution to each individual data set in order to obtain as good an R^2 as possible.

Since there is a relationship between the mean and variance of a theoretical distribution and its parameters, the mean and the variance of the empirical data can be used to provide an initial estimate of the parameter values to be tuned to optimize R^2 . It can happen, however, that the equations that give the dependency between the parameters and the mean and the variance cannot be solved (or do not yield acceptable values). This is the same as saying that there were no valid values for the parameters of the theoretical distribution in question that would generate the sought mean and variance. In this case no R^2 value is generated.

Finally, since population fitness distributions vary from run to run, multiple independent EA runs are used to collect a set of snapshots of the same EA time period. Each of those snapshots is quantitatively analyzed against the 9 theoretical distributions and the resulting R^2 values are presented in “box plot” format as illustrated in Figure 3.

The box plot format permits a concise, comparative visualization of the mean of the valid R^2 s, the confidence interval for the mean, the quartiles and the outliers. Since larger R^2 values correspond to better fits, the data presented in Figure 3 indicates that 4 of the 9 theoretical distributions provide reasonable fits to this particular set of population fitness distribution snapshots. The lines below the box plot summarize the data sets used to produce them.

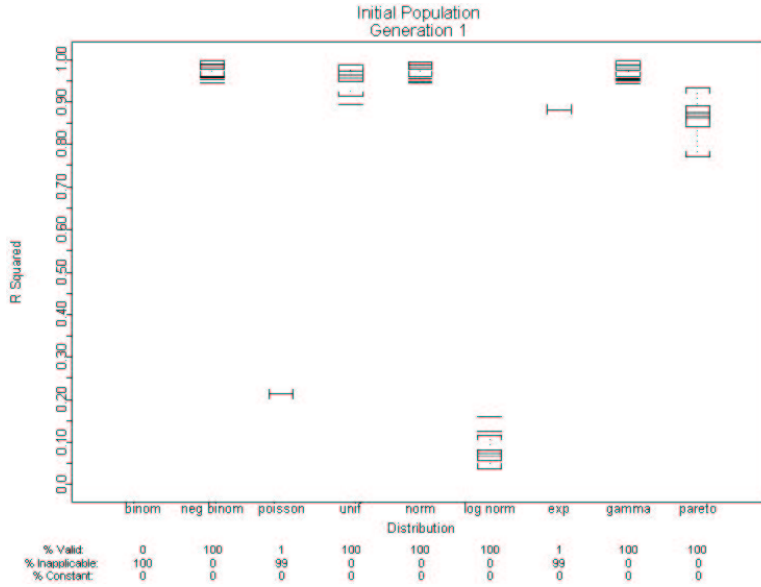


Fig. 3. A box-plot visualization of the R^2 values for 9 theoretical distributions produced from 100 independently generated EA initial population fitness distributions on F1.

3 Initial Experiments

Our initial experiments were designed to provide some insight into the following questions:

- how dependent are these observations on the particular type of EA being used?
- how dependent are the observed population fitness distributions on a particular fitness landscape?
- can useful insights be gained by taking snapshots within a generation to study the effects of selection and reproduction on population fitness distributions?

3.1 EA details and settings

Since earlier work had focused on the population fitness distributions of standard generational GAs, for contrast we focused on an EA that used a binary representation, standard crossover and mutation, but used $\mu + \lambda$ population dynamics. The particular EA parameter settings were:

- parent population size: 100 - randomly initialized
- offspring population size: 200
- bit-flip mutation applied at a rate of $4/L$ (string-length)
- parameterized uniform crossover (0.2) applied at a 0.8 rate

3.2 Data collection

An important part of the experimental design is to decide at what point in the evolutionary process snapshots of population fitness distributions are to be taken. In our case we are not just interested in snapshots taken on generational boundaries, but also within generations in order to see the effects of selection and reproduction on population fitness distributions. So, in our case, this meant that each independent EA run involving n generations produced $1 + 3 * n$ snapshots of the population fitness, corresponding to one snapshot for the initial population and three snapshots (after crossover, after mutation and after survival) for each generation.

For the results presented here, each experiment involved $p = 100$ independent EA runs of $n = 50$ generations each. The large collection of generated data is then systematically analyzed using the Q-Q and R^2 techniques described in the previous section to see how well the set of $k = 9$ selected theoretical distributions model the empirical data for each of $1 + 3 * n$ sets of snapshots.

4 Initial Experimental Results

To keep things simple initially, our first set of experiments used the standard formulation of F1 from the De Jong test suite: minimize $F_1(x) = \sum_{i=1}^d x_i^2, x = (x_1, x_2, \dots, x_d)$. The value used for d was 4 and the x_i were restricted to $[-5.12, 5.12]$. The minimum fitness value is 0 and, if it is found, the algorithm is stopped before the maximum number of generations is reached.

4.1 Observed Differences Between $\mu + \lambda$ and Generational GAs

We have already noted earlier that increasing the selection pressure in a generational GA results in significant qualitative and quantitative differences in the evolving population fitness distributions. The effects are even more striking if we switch to a $\mu + \lambda$ EA as illustrated in Figure 4. Visually compare these results with those presented earlier in Figure 2 produced by a generational GA. Notice how the EA's strong truncation selection pressure dramatically distorts the shape of the fitness distributions and produces a series of fitness distributions whose means and variances are steadily decreasing.

4.2 Observed Effects of Selection, Crossover and Mutation

So far we have been looking at snapshots of population fitness distributions to get a sense of how they change over a number of evolutionary generations. Of equal interest is to look at snapshots taken within a single generation to see how the operations of selection, crossover, and mutation affect the population fitness distributions. For example, a nice intuitive picture (or hypothesis) of generational GA suggested by the earlier discussed figure in [9] might be as follows. At the beginning of a generation, the population fitness distribution is

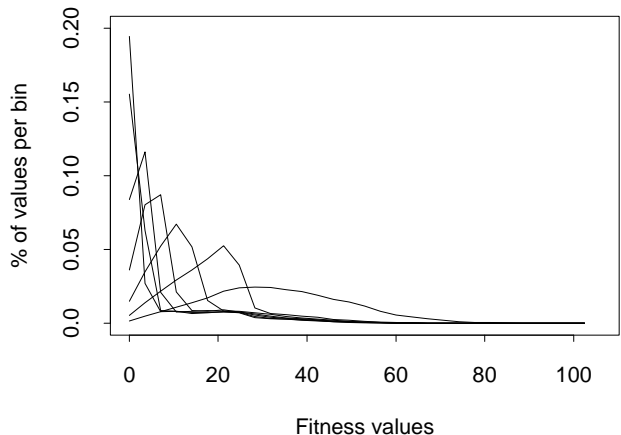


Fig. 4. Histogram averages of population fitness produced by a $\mu + \lambda$ EA on F1.

normally distributed. Parental selection skews this distribution (to the left for minimization problems). The subsequent effect of crossover and mutation is to renormalize the population fitness distribution. So, the next generation begins with a normal distribution whose mean and variance is smaller than at the beginning of the current generation.

While that makes an appealing story, we already know from Figure 2 and Figure 4 that a change in either the fitness landscape or the selection pressure can invalidate this hypothesis. Our methodology allows us to push such an analysis a step further. Not only can we visualize the effects of such operations, but we can also use the Q-Q and R^2 techniques to see which, if any, of the standard distributions fits the actual data. Figures 5 - 7 illustrate this nicely.

Figure 5 illustrates the effects that crossover, mutation, and truncation selection have on the population fitness distribution. There is little difference in the distortions produced by crossover and mutation in the early generations. However, as the population converges, the difference in distortions becomes more apparent. This is most clearly seen in Figures 6 - 7 which summarize how well these distorted distributions match any of the 9 theoretical distributions. In the early generations, the box plots are quite similar. However, as the population converges, the distortions produced by crossover fail to match any of the theoretical distributions while the distortions after mutation continue to match several theoretical distributions reasonably well.

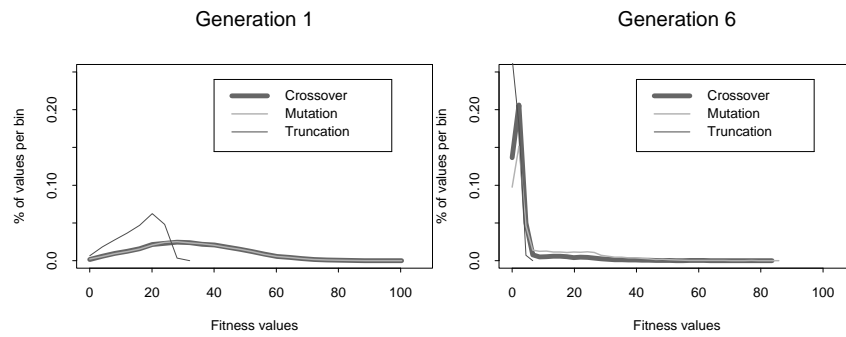


Fig. 5. Histograms showing the effects of crossover, mutation and truncation on F1.

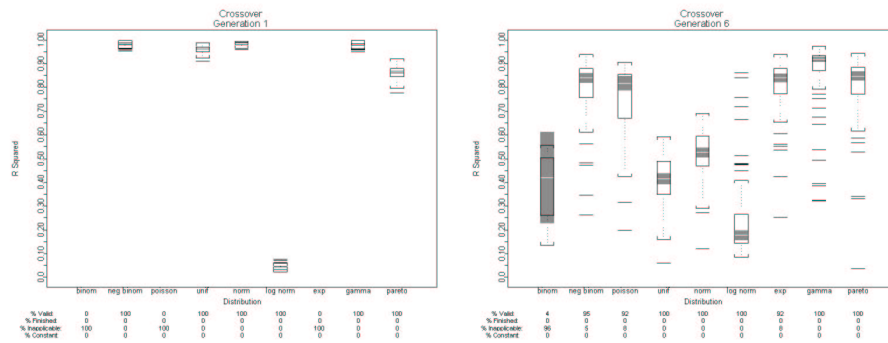


Fig. 6. Box plots characterizing the effects of crossover on F1.

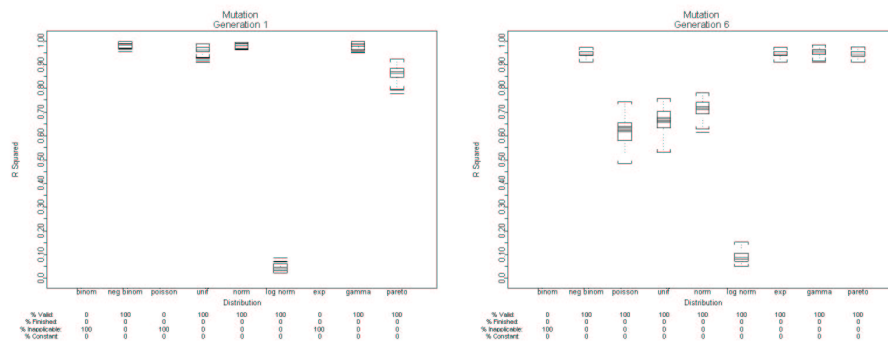


Fig. 7. Box plots characterizing the effects of mutation on F1.

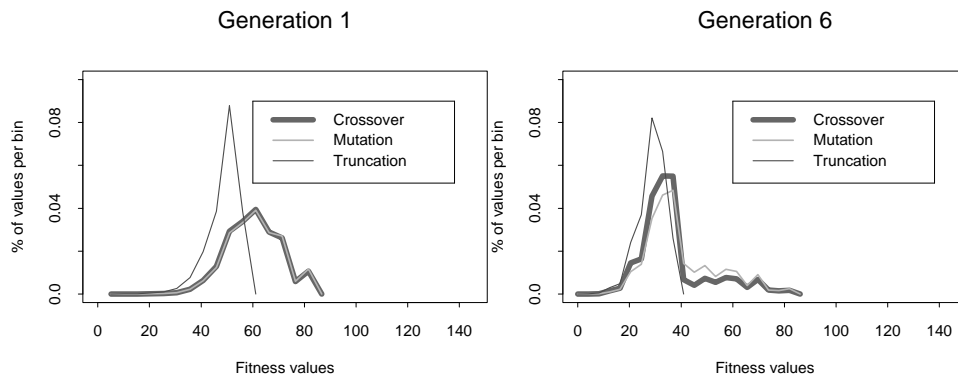


Fig. 8. Histograms characterizing the effects of operators on CA.

4.3 Experiments Using a Complex Cellular Automata Fitness Landscape

As a contrast to the simple fitness landscape F1, we also applied this methodology to a much more complex problem, namely, that of evolving rules for two-dimensional cellular automata (CAs) that generate some predefined pattern [7].

The CAs used operate on a 12-by-12 grid of 0s and 1s, surrounded by a border. The CAs are synchronous, at each step all cells update their value in parallel, by looking at their own previous value and the previous values of their neighbors and applying some set of rules. The size of the neighborhood used was $5 = 1$ (the cell itself) + 4 neighbors (North, South, East and West). Each CA update rule was encoded using 11 bits. Rule sets of size 30 were used.

For evaluation, each CA was run for 128 iterations or until it converged, whichever happened first. The Hamming distance d_{it} between the current grid and the goal pattern was measured at each iteration. The goal was to evolve a CA rule set capable of generating a pattern that is distance 0 from the target pattern using as few CA iterations as possible. For the experiments reported here, the target pattern used was a spiral of 1s starting in the upper-left corner.

Figures 8 - 10 summarize the results. Figure 8 illustrates that, although the population fitness distributions for the CA problem are quite different in shape, the distortion effects due to crossover and mutation are much the same as we saw on F1. In the early generations they have pretty much the same effect, but increasingly differ in their effects as evolution proceeds. As with F1 the box plots in Figures 9 - 10 show this divergence of effects very clearly and are strikingly similar to those for F1 (except that Pareto seems to be a better fit for CA while gamma is more suitable for F1).

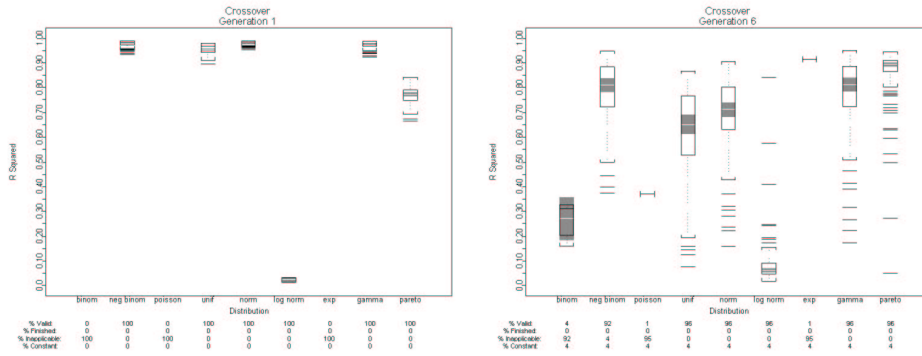


Fig. 9. Box plots characterizing the effects of crossover on CA.

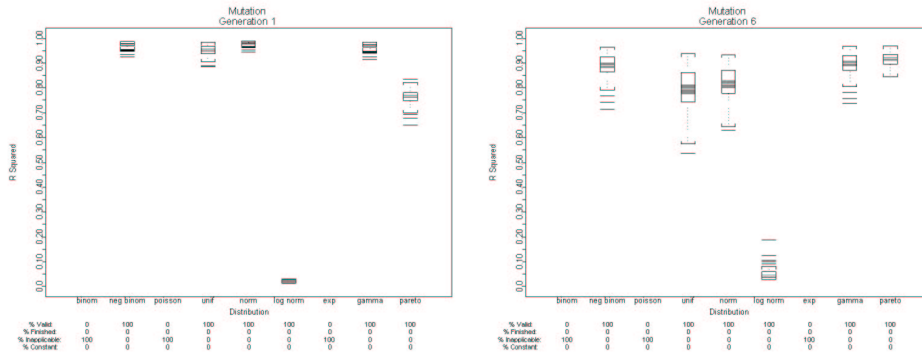


Fig. 10. Box plots characterizing the effects of mutation on CA.

5 Discussion And Conclusions

In summary we are optimistic that the methodology presented here will prove to be a useful addition to the current set of tools for analyzing the behavior of EAs. Even the simple experiments presented here have yielded useful insights, including the fact that population fitness distributions are *seldom* observed to be normally distributed, that the shapes of these distributions are heavily dependent on both the fitness landscape and the EA selection pressure, and that the differences between the population fitness distribution distortions due to crossover and mutation are only significant in the later stages of the evolutionary process.

6 Future Work

Clearly, the experiments presented here are quite preliminary in nature, and are suggestive rather than conclusive regarding several important characteristics of population fitness distributions. Additional experiments involving additional fitness landscapes and other EA variations are required to solidify these results.

An important next step will be to see how other existing tools can be used in a complementary manner with this methodology in order to provide additional insights beyond those provided by any single technique. Our ultimate goal is to use these techniques to understand the behavior of EAs in a deeper sense and enable us to design more effective EAs.

References

1. Lee Altenberg. The Schema Theorem and Price's Theorem. In L. Darrell Whitley and Michael D. Vose, editors, *Foundations of Genetic Algorithms 3*, pages 23–49, Estes Park, Colorado, USA, 1995. Morgan Kaufmann.
2. John J. Grefenstette. Predictive models using fitness distributions of genetic operators. In L. Darrell Whitley and Michael D. Vose, editors, *Foundations of Genetic Algorithms 3*, pages 139–161, Estes Park, Colorado, USA, 1995. Morgan Kaufmann.
3. R. Jain. *The Art of Computer Systems Performance Analysis*. John Wiley and Sons, Inc., New York, 1991.
4. T. Jones and S. Forrest. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In L. eschelmann, editor, *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 184–192. Morgan Kaufmann, 1995.
5. Stuart A. Kauffman. Adaptation on rugged fitness landscapes. In Daniel L. Stein, editor, *Lectures in the Sciences of Complexity*, volume 1, pages 527–618. Addison Wesley, 1989.
6. B. Manderick, M. de Weger, and P. Spiessens. The genetic algorithm and the structure of the fitness landscape. In R. K. Belew and L. B. Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 143–150. Morgan Kaufmann, 1991.
7. Melanie Mitchell, James P. Crutchfield, and Rajarshi Das. Evolving cellular automata with genetic algorithms: A review of recent work. In *Proceedings of the First International Conference on Evolutionary Computation and Its Applications (EvCA '96)*. Russian Academy of Sciences, 1996.
8. H. Mühlenbein and D. Schlierkamp-Voosen. Predictive models for the breeder genetic algorithm. *Evolutionary Computation*, 1(1):25–49, 1993.
9. J. Shapiro, A. Prügel-Bennett, and M. Rattray. A statistical mechanical formulation of the dynamics of genetic algorithms. In Terence C. Fogarty, editor, *Evolutionary Computing, AISB Workshop*, volume 993 of *Lecture Notes in Computer Science*. Springer, 1994.