

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/220885653>

# A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization.

Conference Paper · January 1989

Source: DBLP

CITATIONS

950

4 authors, including:



J. David Schaffer  
Binghamton University

117 PUBLICATIONS 10,597 CITATIONS

[SEE PROFILE](#)

READS

2,475



Rich Caruana  
Microsoft

142 PUBLICATIONS 22,225 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



ontology and machine intelligence to develop Bayesian network process for threat detection / response [View project](#)



A Novel Approach to Event-driven Simulation of Spiking Neural Network [View project](#)

**PROCEEDINGS OF THE  
THIRD INTERNATIONAL  
CONFERENCE ON**

# **GENETIC ALGORITHMS**

George Mason University

June 4-7, 1989

Editor/Program Chair: J. David Schaffer

With support from:

Navy Center for Applied Research in Artificial Intelligence,  
Naval Research Laboratory  
Philips Laboratories, North American Philips Corporation

Conference Committee:

Kenneth A. De Jong, George Mason University (Conference Chair)  
John J. Grefenstette, Naval Research Laboratory (Finance)  
Lashon B. Booker, Naval Research Laboratory (Local Arrangements)  
David E. Goldberg, University of Alabama (Publicity)  
J. David Schaffer, Philips Laboratories (Program)  
Lawrence Davis, Bolt, Beranek and Newman, Inc.  
John H. Holland, University of Michigan  
George G. Robertson, Xerox PARC  
Stephen F. Smith, Carnegie-Mellon University  
Stewart W. Wilson, Rowland Institute for Science

Morgan Kaufmann Publishers, Inc.  
San Mateo, California

1 2 3 4 5 6 7 8 9 10 11 12

improvement in the on-line performance of phenotypic sharing.

### Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant MSM-8451610.

### References

- Booker, L. B. (1982). Intelligent behavior as an adaptation to the task environment. (Doctoral Dissertation, Technical Report No. 243. Ann Arbor: University of Michigan, Logic of Computers Group). *Dissertation Abstracts International*, 43(2), 469B. (University Microfilms No. 8214966)
- De Jong, K. A. (1975). An analysis of the behavior of a class of genetic adaptive systems. (Doctoral dissertation, University of Michigan). *Dissertation Abstracts International*, 36(10), 5140B. (University Microfilms No. 76-9381)
- Deb, K. (1989). *Genetic algorithms in multimodal function optimization*, Master's thesis, TCGA Report No. 88002, University of Alabama, Tuscaloosa.
- Goldberg, D. E. (1983). Computer-aided gas pipeline operation using genetic algorithms and rule learning. (Doctoral dissertation, University of Michigan). *Dissertation Abstracts International*, 44(10), 3174B. (University Microfilms No. 8402282)
- Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Reading, MA: Addison-Wesley.
- Goldberg, D. E., and Richardson, J. (1987). Genetic algorithms with sharing for multimodal function optimization. *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms*, 41-49.
- Goldberg, D. E., and Segrest, P. (1987). Finite Markov chain analysis of genetic algorithms. *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms*, 1-8.
- Goldberg, D. E., and Smith, R. E. (1987). Nonstationary function optimization using genetic algorithms with dominance and diploidy. *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms*, 59-68.
- Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor: The University of Michigan Press.
- Reklaitis, G. V., Ravindran, A., and Ragsdell, K. M. (1983). *Engineering optimization methods and applications*. New York: Wiley.

### A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization

J. David Schaffer  
Richard A. Caruana  
Larry J. Eshelman  
Rajarshi Das

Philips Laboratories, North American Philips Corporation, 345 Scarborough Road, Briarcliff Manor, New York 10510

### Abstract

We present results from a large experiment on the effects of changes in control parameters on the performance of genetic search on function minimization tasks. Online average performance was seen to be sensitive to the combined influences of population size, crossover and mutation rates and to the number of crossover points used in each mating. A greater sensitivity to mutation rate than crossover rate was observed as was a small superiority for 2-point crossover. These effects appear to be independent of the function being searched. Some differences among functions were also observed, leading us to hypothesize that naive evolution (i.e., selection and mutation alone) may be a more significant factor in the behavior of genetic algorithms than has previously been recognized.

### 1. Introduction

It has long been acknowledged that the parameters that control a genetic algorithm (GA) can have a significant impact on its performance and that the theory behind this technology gives little guidance for their proper selection. Given this, it is surprising that so few studies have been done in this area. Empirical studies were done in 1975 [4] and in 1986 [6] and one theoretical study of population size was done in 1985 [5]. Recently an empirical study of population size on parallel machines was done [8]. Believing that there was a need for a more thorough investigation, we conducted a large empirical study, in the hope of gaining insight into the size and direction of the influences of these parameters on genetic search. In this paper, we report our findings with respect to online average performance.

(ds1@philabs.philips.com)  
(raca@philabs.philips.com)  
(lje@philabs.philips.com)

### 2. Background

In 1975, De Jong [4] set out to compare genetic algorithms (GAs) with the best gradient techniques. In this study he devised a suite of five test functions that exhibited a range of properties that presented more or less difficulty to the gradient techniques. Before conducting his comparisons, he performed several experiments to try to gain some insight into the influence of population size, crossover and mutation rates on the efficacy of genetic search. He also investigated generation gap (a means to insure the survival of some proportion of the population to forestall premature convergence) and several variant operators, but the three parameters above were the most vigorously studied and have had the most impact on subsequent research and practice.

To assess the impact of these parameters, he proposed two performance measures, *online* and *offline* averages. Online average is simply the average performance of all structures tested during the search. This measure would be appropriate in situations where the testing of every structure must be "paid for" (and hence the idea of *online*) and penalizes search methods that must test many poor structures before locating good ones. To do well on online performance, a search algorithm must quickly decide where the best values lie and concentrate its search there. The other measure, *offline*, is computed by using only the current best structure value for each evaluation in the average. It is appropriate in situations in which a search might be done "offline" while the current best structure is used (and paid for) until a better one is found. This measure does not penalize search methods for exploring poor regions of the search space on the way to better solutions.

Because of limited computing resources, De Jong could make only limited comparisons among the different combinations of parameter values he wished to study. He was able, however, to offer the following conclusions:

Increasing the population size was shown to reduce the stochastic effects [of random sampling on a finite population] and improve long-term performance at the expense of slower initial response. Increasing mutation rate was seen to improve off-line performance at the expense of on-line performance. Reducing the crossover rate resulted in an overall improvement in performance, suggesting that producing a generation of completely new individuals was too high a sampling rate. [4, p191]

From these experiments there emerged a set of values for these parameters that was found to yield generally good behavior for this suite of problems for both online and offline performance. Subsequently, these values became something of the conventional wisdom on this issue. They were:

population size 50-100  
crossover rate 0.60  
mutation rate 0.001

De Jong also investigated the use of generalized (i.e., multi-point) crossover. After showing that theoretically these operators exhibit very different schema disruption behavior, he concluded that no significant empirical difference could be seen on the suite of test problems.

A more robust approach was suggested by Grefenstette [6] when he used a GA to address the control parameter problem by treating it as a meta-problem. That is, a meta-GA was used to locate parameter sets which themselves were used for GA searches on a set of tasks (the De Jong suite). This approach has the advantage of being able to explore the space of parameter combinations efficiently; it does so by using the GA's powerful implicit parallelism. Unfortunately, it suggests good parameter sets without providing much insight as to how sensitive performance is to variations. Using this method, Grefenstette was able to locate an interesting parameter set that provided significantly better online performance than De Jong's settings, but was unable to improve upon the conventional wisdom for offline performance. His recommended values were:

population size 30  
crossover rate 0.95  
mutation rate 0.01

Note that he recommended a smaller population size and much higher rates of applying the genetic operators than did De Jong.

One theoretical investigation of optimal population size has been conducted by Goldberg [5]. He proposed as a figure of merit for population size, the expected number of new schemata (new in the sense of not already being in the population) per population member. Using this figure of merit and assuming an initial binary coded population randomly generated with equal probabilities for 0 and 1, Goldberg derived an expression for optimal population size. As an approximation to this expression, he reported  $\text{pop} = 1.65 \cdot 2^{0.21\text{length}}$  that suggests population sizes of 130, 557, 2389 and 10244 for strings of lengths 30, 40, 50 and 60 respectively. It is clear that this method suggests very large populations for problems with moderately long chromosomes.

Robertson's investigation of population size on parallel machines [8] found that performance monotonically increased with population size. This is not surprising when there is no "cost" for these large populations. Our interest was in serial machines, or other settings where there is a fixed cost increment for each population member; we do not expect performance to continue to increase with population size.

From this limited amount of research one may draw several conclusions. The results from the empirical studies have not been completely consistent and have differed considerably from the theoretical recommendations. The empirical studies have depended heavily on the De Jong problem suite and have concentrated on parameter sets that provide robust performance on the whole suite. Hence they have reported little on differences among the functions. Our conclusion was that a complete factorial design experiment should be done and analysis of variance (ANOVA) used to identify and quantify the influences of these control parameters. In addition, we felt it necessary to expand the suite of problems.

### 3. The Study Design

The first step in the study design was to expand the problem suite. We included the De Jong functions, but departed from previous use of these

functions by representing all search space parameters using the Gray code. This decision stemmed from our previous finding that Gray coding was often better than traditional binary coding for numeric parameters [3]. In addition, we selected five more problems in an attempt to include a wider range of search space characteristics. F6 and F7 were multimodal with different patterns of their well depths and the heights of the barriers between wells. (The GA is searching for minima in all of these problems.) They were designed to present more or less difficulty to search by simulated annealing. A 2D cross section through the origin of each is shown in Figure 1. Both surfaces were cylindrically symmetric about the z axis and the points in the search space were coded as Cartesian x and y values in the range -100 to +100 with 22-bit Gray code. F8 was a FIR digital filter optimization task with 16 4-bit Gray coded parameters. The high dimensional surface does not lend itself to visualization, but we have estimated that it contains more than 50 million local minima, most of which are very poor. F9 was a well known 30-city traveling salesperson problem [7]. We used the sort order representation suggested by Smith and used by Shaefer[10]. We used 5 gray coded bits for each city. F10 was a highly structured 64-node graph partitioning task studied by Ackley [1]. The representation contained one bit per node, with zero and one specifying to which side of the partition each node belonged. The performance measure for each function was normalized so that a value of 1.0 corresponded to the function average over the domain studied and 0.0 corresponded to the global optimum. A summary of these functions is given in Table 1.

The experimental design was a complete factorial design using 10 replications (using different random seeds and different randomizations of the city numbering and the node numberings for F9 and F10) at each combination of the following GA parameter values: 6 population sizes (10, 20, 30, 50, 100, 200), 10 crossover rates (0.05 to 0.95 in increments of 0.10), 7 mutation rates (0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.10) and 2 crossovers per mating (1 and 2). This design was selected as a compromise among the goals of including a wide enough range of values to bracket the best GA performance, of having enough resolution without increasing the computational load beyond our resources and of being able to compare our findings with previous research. A quick multiplication shows that the study required 8400 genetic searches. Each search was run to 10,000 evaluations with performance statistics recorded at intervals of 1000

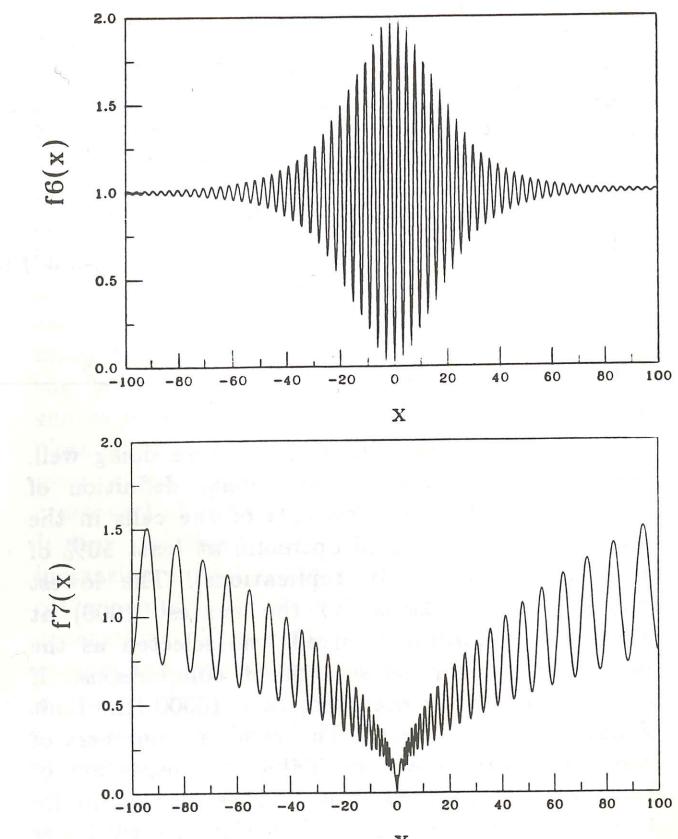


Figure 1. Central cross section of  $f_6$  and  $f_7$

evaluations. The total experiment consumed approximately 1.5 CPU years (on Sun 3 and VAX machines) and the output data occupied approximately 85 megabytes of disk storage.

It is also important to mention the GA controls that were not part of the experiment, i.e., those held fixed throughout the experiment. These were as follows: scaling window [6] (1), coding (Gray for all numeric parameters), spinning threshold (2 generations without an evaluation), convergence threshold (95%), reevaluation of an already evaluated individual (no for all functions except F4), elitist strategy (yes) and selection procedure (Baker's stochastic universal sampling [2]).

Comparing performances among stochastic search procedures can be problematic. Comparison of performances at fixed computational cost requires the choice of the number of evaluations at which to make this comparison. (If all searches find the global optimum, which ours did not, then the computational cost can be compared as Ackley did [1].) Here we departed from the previous practice of selecting a single number of evaluations for all functions. We reasoned that, since we were trying to discriminate among 840 different combinations (cells) of control parameters, we should choose a

Function	Form	Size of Space ( $2^L$ )	Description/Name
F6	$0.5 + \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{[1.0 + 0.001(x_1^2 + x_2^2)]^2}$	L = 44	Sine envelope sin wave
F7	$(x_1^2 + x_2^2)^{0.25} [\sin^2(50(x_1^2 + x_2^2)^{-1}) + 1.0]$	L = 44 L = 64 L = 150 L = 64	Stretched V sine wave FIR Filter 30-city traveling salesperson Graph partitioning
F8			
F9			
F10			

point at which some, but not all, were doing well. We chose the following operational definition of "doing well": at least 10% (84) of the cells in the design located the global optimum at least 50% of the time (5 of the 10 replications). The lowest number of evaluations (to the nearest 1000) at which this condition occurred was selected as the number to use for all subsequent comparisons. If this condition never occurred, then 10000 (the limit of our study) was used. The resulting numbers of evaluations are shown in Table 2. Inspection of Table 2 shows that, as suspected, several of the De Jong functions can be solved in significantly fewer evaluations than others. Hence, a research strategy that uses the same number for all functions, is likely to bias the results [6,9]. It is also apparent that 6 of the 10 functions failed to reach our criterion for "doing well" (F2, F6, F7, F8, F9, F10). Of these, only F9 proved too hard for the GA to solve at least once in 10,000 evaluations.

The analysis commenced with a standard analysis of variance. Among those effects found to be significant ( $\alpha < .05$ ) we concentrated on those of highest order as the ones providing the most meaningful insights. Post hoc contrasts were computed using the Tukey B method to protect the experiment wise  $\alpha$  level. One such contrast often used was an attempt to identify the "best online pool." This was the subset of cells whose mean online performance was statistically indistinguishable from the best single cell in the set. We cannot claim that this approach is the only valid one for such a large dataset, but we believe that it allowed us to test for many of the most meaningful effects.

#### 4. Results

In this paper we focus exclusively on online average performance. As previous researchers have noted, the results for offline or best individual performance are almost surely quite different.

TABLE 2. The number of evaluations at which performance was compared			
function	evaluations to criterion	cells >5/10	cells >1/10
F1	2000	132	283
F2	10000	36	312
F3	2000	158	454
F4	4000	131	334
F5	4000	132	303
F6	10000	29	245
F7	10000	77	275
F8	10000	0	40
F9	10000	0	0
F10	10000	0	60

#### 4.1. Function-independent findings

An interaction among population-crossover-mutation (pcm) was strongly indicated. Interestingly, the function-pcm (fpcm) interaction was confidently rejected.<sup>†</sup> Paralleling this observation, the pcm-number\_of\_crossover\_points effect was significant, while fpcmn was not. Thus, we believe we have statistical support for some important function-independent effects.

We will focus on the pcm effect by examining the 2-point data displayed in Figure 2. Note that because of the normalization of performance, the top of the vertical axis (online = 1.0) is the expectation for random search and the bottom (online = 0.0) corresponds to the global optimum which is an unrealistic value for online average. The x and y axes represent the mutation and crossover rates respectively, each plotted on a linear scale even though the crossover sampling was linear ( $0=0.05, 1=0.15, \dots 9=0.95$ ) and the mutation rate

<sup>†</sup> Although we did not perform a direct  $\beta$  test of the null hypothesis, it is generally recognized that  $\alpha$  and  $\beta$  confidences are inversely related. The  $\alpha$  level on this effect was very high (the algorithm reported 1.0) and so we feel very confident in accepting the null hypothesis.

sampling was not ( $0=0.001, 1=0.002, \dots 6=0.10$ ).

To identify the influence of number\_of\_crossover\_points, the best online pool of cells was identified for each population size-number\_of\_crossover\_points combination.<sup>‡</sup> The means of these pools were compared between 1-point and 2-point crossover, separately for each population size. The reason for comparing only pools (the sets of cells with the best performance) is that, the F test in the ANOVA is sensitive to differences over the whole range of the independent variables, but we are less interested in differences in the regions of poorer performance. We noted that 2-point provided the better online performance at populations of 50 and 100, and that there were no significant differences at the other population sizes. This suggests that 2-point is no worse and sometimes better than 1-point crossover. As suggested by these contrasts, the forms of the curves for 1-point crossover ( $nc=1$ ) were almost identical to those shown in Figure 2. A single best online pool was then computed for this entire set. That is, the best cell among all populations at  $nc=2$  was compared with all others. These points are circled in Figure 2 with the best individual cell labeled "B". For comparison, the De Jong recommendations are labeled "D" and the Grefenstette set "G".

The pattern of the pcm interaction can be seen clearly. At low population size (10) good performance is very sensitive to mutation rate ( $m$ ) and less so to crossover rate ( $c$ ). It can be achieved with  $m=0.02$  and  $c=0.85$ . (It should be mentioned that this point is the worst in the best online pool.) As population size is increased, the sensitivity to mutation rate decreases (the pool widens) and the best mutation rate to use also decreases ( $m=0.002-0.005$  at  $p=50$ ). The generally inverse relation between population and mutation probably reflects the fact that increasing either one increases exploration; they may be traded off while maintaining a more or less constant level of exploration. We were gratified that the range of values we included in our design was sufficient to see the inflection in this tradeoff.

A similar, but less dramatic effect is seen for crossover rate. Note that the best performance at  $p=200$  is achieved at the lowest rates for both crossover and mutation. The generally worsening performance at these higher population sizes (there are no members of the best online pool above

<sup>‡</sup> The choice of population as the first discriminating factor was an arbitrary one.

populations of 50) is probably attributable to generation effect. Since we are comparing performance at a fixed number of evaluations, the higher population sizes get many fewer generations than do the smaller ones. A less obvious aspect of this generation effect results from a strategy employed by our GA. If an individual survives intact into the next generation (i.e., not mutated or mated), it is not reevaluated (except for F4). For cells with low rates of both mutation and crossover, many such events occur each generation, allowing the "unused evaluations" to extend the number of generations before the experimental limit was hit. Note for instance, the steady dropping of the left-most corner (the lowest rates of mutation and crossover) of each plane as population is increased. It drops relative to the lower population sizes due to increasing initial exploration, and relative to the other points in its plane due to increasing available generations. This compound generation effect may explain the inverse population crossover interaction we see in Figure 2. High crossover is best at low populations, a broad range of crossover rates is tolerated at the middle population sizes and only low crossover rates are suggested for the high population sizes.

The 31 members of the best online pool are listed in ascending order of mean online performance in Table 3. Note that one may be inclined to do an "eye-ball" t-test and conclude that the last several members look significantly different from the best. This may or may not be so. The Tukey B test used to identify these members is necessarily a conservative  $\alpha$  test. That is, while protecting the experimentwise confidence, we conclude that those outside of this pool are significantly different from the best member. Thus, we may err on the side of including too many in the best online pools. Nevertheless, it does seem to be strongly indicated that good online performance can be expected with:

population size 20-30  
crossover rate 0.75-0.95  
mutation rate 0.005-0.01

Some latitude to go beyond these ranges seems indicated and analysis of offline performance may indicate which direction looks like the best compromise.

#### 4.2. Function-dependent findings

The fpcn interaction was not significant with a high  $\alpha$  level (0.55) so again we feel reasonably confident in accepting the null hypothesis. The

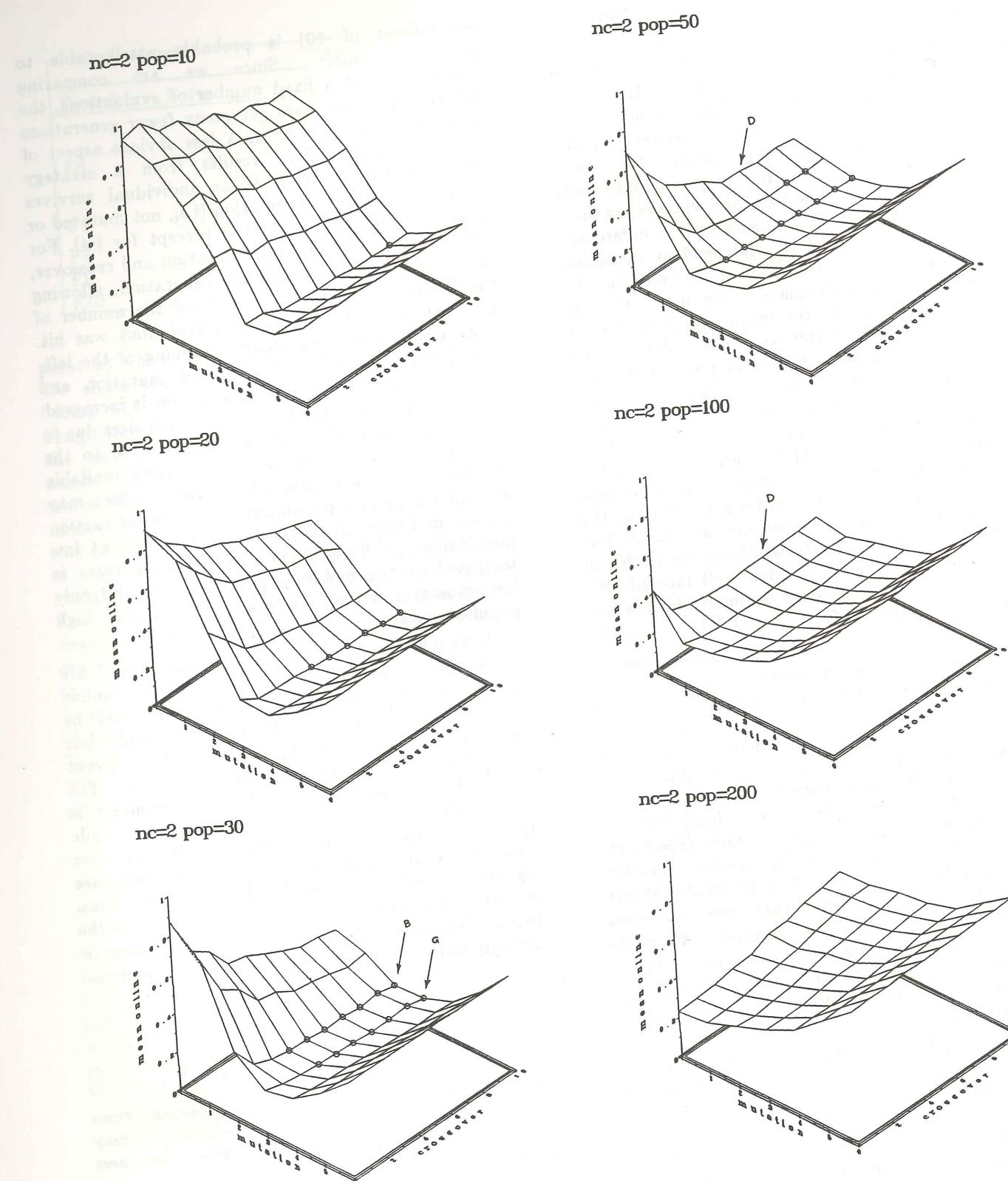


Figure 2. Online average performance of the GA as a function of population size, crossover and mutation rates using 2-point crossover. Each point represents the mean online performance of 100 GA searches, 10 replications for each of 10 functions.

TABLE 3. The Function-independent Best Online Pool at nc=2

population	crossover rate	mutation rate	online mean	s.e.m
30	0.95	0.005	0.141	0.011
20	0.95	0.01	0.143	0.012
20	0.85	0.01	0.145	0.011
30	0.75	0.005	0.147	0.013
30	0.95	0.01	0.149	0.011
50	0.75	0.005	0.153	0.011
30	0.55	0.005	0.153	0.012
50	0.65	0.005	0.153	0.011
50	0.95	0.005	0.154	0.011
50	0.45	0.005	0.154	0.011
30	0.65	0.01	0.156	0.011
50	0.85	0.005	0.158	0.011
30	0.75	0.01	0.159	0.012
30	0.65	0.005	0.159	0.015
20	0.65	0.01	0.159	0.014
20	0.55	0.01	0.160	0.013
50	0.55	0.005	0.161	0.012
30	0.55	0.01	0.163	0.012
20	0.75	0.01	0.163	0.016
50	0.25	0.005	0.163	0.012
30	0.85	0.005	0.164	0.015
50	0.35	0.005	0.165	0.012
30	0.45	0.005	0.167	0.014
20	0.45	0.01	0.168	0.014
30	0.85	0.01	0.169	0.013
30	0.45	0.01	0.169	0.012
30	0.35	0.005	0.171	0.013
30	0.35	0.01	0.171	0.012
50	0.65	0.002	0.173	0.016
50	0.85	0.002	0.174	0.016
10	0.85	0.02	0.177	0.014

fcmn interaction appeared to be significant, but when the ANOVA was repeated on log transformed data (to check for possible biases in the F tests due to correlations among the cell means and variances) the significance vanished.<sup>†</sup>

We now turn our attention to the fpmn interaction which was significant. We followed an approach for this effect similar to the one used above. A best online pool was identified for each function and crossover operator (1-point and 2-point) separately. The mean online performance for the best online pool was compared for 1-point vs. 2-point crossover for each function separately. The result was that 2-point seemed to yield better online performance for F7 and F9 only; the other functions

<sup>†</sup> This was one of only two differences between the analyses of the raw and the transformed data. The other involved the appearance of a significant cmn interaction that was not significant in the analysis of the raw data. We are inclined to believe the results of the transformed analysis as being less influenced by slight violations of the assumptions of ANOVA.

showed no difference. The locations of the best online pools (for 2-point only) are shown in Figure 3. In the figure, the function symbols are clustered around the point to which they apply and lines surround the clusters in an attempt to aid visualization.

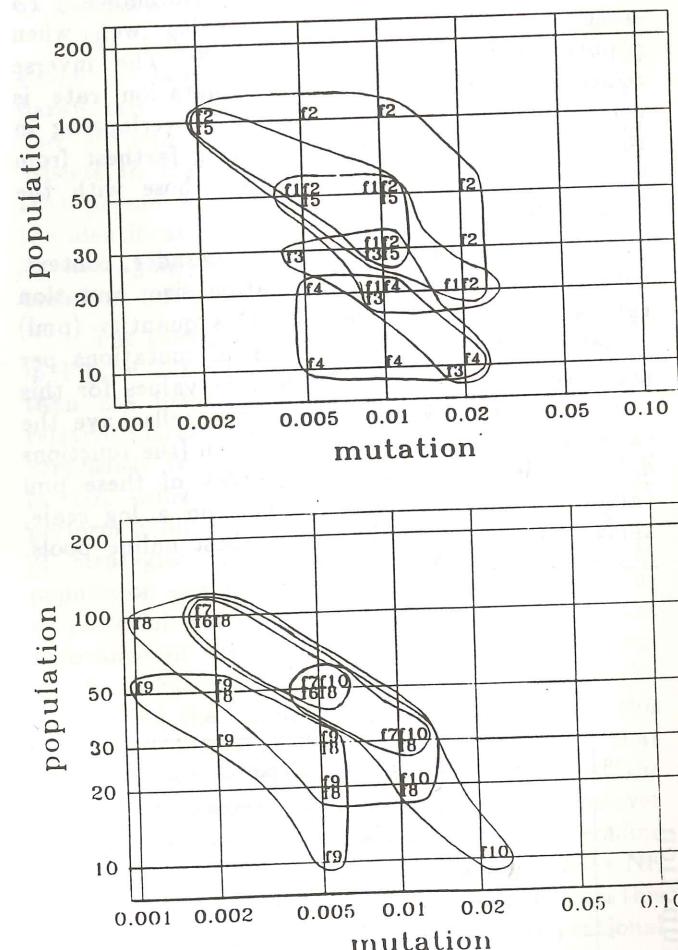


Figure 3. The best online pools by function for 2-point crossover. Each labeled point was a member of the best online pool and its online performance is the average of 100 online averages, 10 replication for each of 10 crossover rates. The functions were separated into two groups for visual clarity.

It can be seen that the best performance is never to be found at a population size of 200 or at mutation rates of 0.05 or higher (when averaged over all crossover rates). Furthermore, the combinations of small population size with low mutation rate ( $p < 20$ ,  $m < 0.002$ ) never produce a best online performance.

Between these extremes the inverse relation between population size and mutation rate mentioned above is generally evident.

As the eye roves over Figure 3, several patterns are noteworthy. F3, F4, F9 and F10, seem to do well with small populations and high mutation rates. The conventional wisdom has been that very small population sizes are too prone to genetic drift to be associated with good performance. F4 especially stands out by not doing well when population size is larger than 20. The inverse relation of population size to mutation rate is generally evident with many pools overlapping in the central region. The pools lying farthest from this central region appear to be those with the longest chromosomes.

To put these pools in a broader context, consider the product of population size, mutation rate and chromosome length. This quantity ( $pml$ ) represents the expected number of mutations per generation. There were 23 distinct values for this quantity in our study design (some cells have the same value for  $pml$ ) for each function (the functions differ on length). The whole array of these  $pml$  values are displayed in Figure 4, on a log scale, along with the locations of the best online pools, shown as darkened circles.

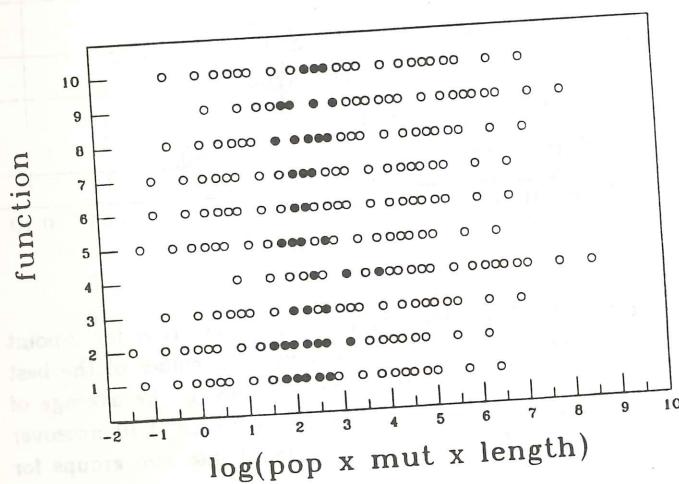


Figure 4. The best online pools in the context of all values for  $\ln(p m l)$  in the study design for each function.

This suggests that among the best online pools, a property related to the total number of mutations in the gene pool per generation is remarkably constant. A linear regression with the logs of

population size, mutation rate and length using the best single cell from each pool yielded this result:

$$\ln(p) + 0.9318 \ln(m) + 0.4535 \ln(\text{length}) = 0.5626$$

If we then project the best online pools onto this regression line, we get the picture shown in Figure 5.

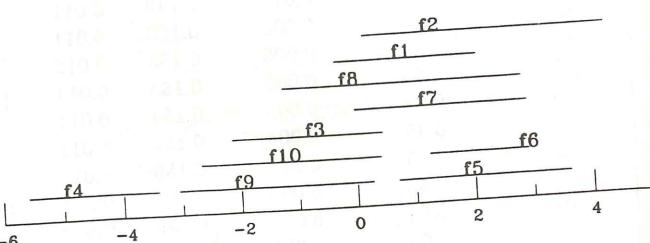


Figure 5. The projection of the best online pools onto the regression line of approximately constant mutations per generation. The labels are placed at the locations of the single best cell within each pool. The vertical placement is purely for visual separation.

We offer the following hypothesis to explain the observed distribution. The relative insensitivity to crossover rate suggests that these winning cells may be relying on a strategy consisting mainly of selection and mutation, what we might call *naive evolution* (NE). We note that the three functions that are highly multimodal with steep sided wells, F5, F6 and F7 are all at the upper end of the regression line. F1 and F2 are known to be smooth and unimodal and seem to fall near each other. F3, towards the middle of the line, is characterized by plateaus, regions where all chromosomes have the same evaluation. F4, at the lower end of the line has the longest chromosome and has added noise. We are unsure of what the contours of F8, F9 and F10 look like. What sort of spectrum of NE might be arrayed along this line?

Consider F4. Without the noise, it is a simple quadratic well with a fairly broad basin about the global optimum. In this region, however the signal to noise ratio is lowest. A NE search with a small population size and a high mutation rate behaves not unlike highly stochastic hillclimbing; stochastic effects are so severe that inferior offspring may survive, but they will, themselves quickly be mutated. The evaluations are reliable only in the very poor regions, so the gene pool quickly locates in the central basin. Here, however, a mutation is no more likely to produce an inferior performer than no change at all, so a high mutation rate may not have an adverse effect on online performance. Note that F4 was the only function for which individuals

surviving intact into the next generation were reevaluated. F3 also has broad basins (plateaus) within which mutation rate has little effect on online average, and, provided the population size is large enough to soften the stochastic effects a little, progress can be made whenever mutation locates a better plateau. On the other hand, a highly stochastic hillclimb would not produce good online average performance when there are many steep sided wells such as in F5, F6 and F7. Here, a large population with a lower mutation rate produces a search more analogous to multiple parallel hillclimbing. By mutating only a small percentage of the population, inferior points can be tried without severe impact on the online performance, while the improved offspring are retained because stochastic effects are low. The smooth contours of F1 and F2 appear to be well searchable with an intermediate strategy.

Armed with the hypothesis that position on this line represents a strategy spectrum, we now consider what influences might limit the range of strategies admitted to the best online pool. The slow, broad search strategies (towards the right side of Figure 5) can be expected to find good points on many contours, but the question is, how quickly? The competition to be included in the best online pool is determined by how well the best cell does which in turn is influenced by the number of evaluations at which the competition is held. For example consider F2 (refer also to Table 2). To get the global optimum reliably is difficult. It was only achieved by 4% of the cells (36/840). It does appear, however, that getting close is much less difficult; the best online pool for F2 had the lowest average online performance of any of the ten functions. By allowing a large number of evaluations to F2 we may have allowed the best online pool to expand to the right hand end of Figure 5. The limit at the left hand end may depend on the severity of the risk to online average imposed by the function when the strategy becomes more stochastic. We offer these speculations in the hope that other researchers may devise ways to test them.

In addition to the effects just discussed, the ANOVA indicated statistical significance for the fpc and fcm interactions. Unfortunately, our analytical approaches to these effects failed to yield insights that we felt were meaningful for the understanding of GA behavior. Each of these effects contain patterns that result when data are collapsed across variables that the analyses above indicate are very influential. The fpc effect involves averaging over mutation rate and fcm involves averaging over

population size. Our feeling is the same with respect to the lower order interactions and the main effects; they yield little useful insight since they collapse across control parameters over which the GA practitioner has control and, in fact, for which he must choose a single value. We do not present these findings.

## 5. Summary

We believe that this is the most thorough study to date of the influence of these control parameters on genetic search. We were gratified to find statistical support for function-independent aspects to the behavior of the GA. An approach that presumes function independence and prevents the identification of function differences, for instance by averaging over functions, may point to robust behavior, but precludes testing this assumption.

The general shape of the performance surfaces in Figure 2 suggests a stronger role for mutation than has previously been admitted.<sup>†</sup> In fact, the relative insensitivity to crossover rate was a little surprising. We speculate that *naive evolution* (NE) (a GA using only selection and mutation) does perform a hillclimb-like search and given the range of strategies that can be achieved by varying population size and mutation rates, it is likely to be a powerful search algorithm, even without the assistance of crossover. The use of Gray coding, which makes hillclimbing even more effective by eliminating the Hamming cliffs, contributes to this effect [3]. Note that we do not wish to imply that crossover had no effect in our experiment. Just the opposite. There is evidence that the lowest crossover rates are not associated with best online performance. We do note, however, that NE appears to be a much stronger component in this experiment than suggested by the conventional wisdom.

A large population size can achieve a large sampling of the space (exploration) at least in the initial generation (if random). However, a large population imposes a large cost per generation, and the exploration for schemata not present in the initial population can be achieved by the operators. This suggests that the criterion used by Goldberg [5] was too conservative, leading him to recommend very large populations.

<sup>†</sup> Grefenstette suggested this in 1986 [6], as did Caruana and Schaffer in 1988 [3], but one still hears mutation referred to as strictly a background operator.

In general, 2-point crossover edged ahead of 1-point, but the effect was not dramatic. This influence can, of course, only be seen to the extent that crossover is contributing to the performance being monitored. If our hypothesis is correct that NE is a major factor in the performance differences we observed, then we expect the differences between 1-point and 2-point to be hard to see. We expect less of the NE effect as populations increase, but at 200 the paucity of generations can again mask the influence of crossover. This may explain why the differences we did see were apparent only at populations of 50 and 100.

Figure 2 provides an overview of the online performance landscape not previously available. In addition, we provide corroboration of Grefenstette's recommendations for parameter settings [6]; they were among the function-independent best online pool and among the best online pools for 7 of our 10 functions. This occurred in spite of our using a different number of evaluations, additional functions and Gray coding.

Function differences appeared most strongly in the interaction of population size and mutation rate. These patterns prompted us to speculate about the range of strategies available to NE and the kinds of functions on which they may be most effective.

We note several cautions that should be exercised when interpreting our results. We have discussed only online average performance; we expect a rather different picture to emerge when we examine offline or best single solution performance. It is not clear to what extent these results may be generalized beyond the 10 functions of our suite. It is encouraging that others have observed generally robust behavior from GAs in widely varying domains, but there have been observations of premature convergence also. The success of relatively high mutation rates and the insensitivity to crossover rates hints that our suite may still be too easy to adequately study the effect of crossover.

#### References

1. D. H. Ackley, *A Connectionist Machine for Genetic Hillclimbing*, Kluwer Academic Publishers, Boston, MA, 1987.
2. J. E. Baker, Reducing Bias and Efficiency in the Selection Algorithm, *Genetic Algorithms and Their Applications: Proceedings of the Second , MIT, July 28-31 1987*, 14-21.
3. R. A. Caruana and J. D. Schaffer, Representation and Hidden Bias: Gray vs. Binary Coding for Genetic Algorithms, *Proceedings of the 5th International Conference on Machine Learning*, Morgan Kaufmann, Los Altos, CA, June 12-14 1988, 153-161.
4. K. A. De Jong, Analysis of the Behavior of a Class of Genetic Adaptive Systems, Ph.D. Dissertation, Department of Computer and Communication Sciences, University of Michigan, Ann Arbor, MI, 1975.
5. D. E. Goldberg, Optimal Initial Population Size for Binary-coded Genetic Algorithms, TCGA Report Number 850001, The Clearinghouse for Genetic Algorithms, Department of Engineering Mechanics, University of Alabama University, Alabama 35486, Nov 1985.
6. J. J. Grefenstette, Optimization of Control Parameters for Genetic Algorithms, *IEEE Transactions on Systems, Man & Cybernetics SMC-16,1* (January-February 1986), 122-128.
7. I. M. Oliver, D. J. Smith and J. R. C. Holland, A Study of Permutation Crossover Operators on the Traveling Salesman Problem, *Genetic Algorithms and Their Applications: Proceedings of the Second , MIT, July 28-31 1987*, 224-230.
8. G. G. Robertson, Population Size in Classifier Systems, *Proceedings of the 5th International Conference on Machine Learning*, Morgan Kaufmann, Los Altos, CA, June 12-14 1988, 142-152.
9. J. D. Schaffer and A. Morishima, An Adaptive Crossover Distribution Mechanism for Genetic Algorithms, *Genetic Algorithms and Their Applications: Proceedings of the Second , MIT, July 28-31 1987*, 36-40. Philips Laboratories, Technical Report-86-052, Dec 30, 1986.
10. C. G. Shaefer, The ARGOT Strategy: Adaptive Representation Genetic Optimizer Technique, *Genetic Algorithms and Their Applications: Proceedings of the Second , MIT, July 28-31 1987*, 50-58.

## Adapting Operator Probabilities In Genetic Algorithms

Lawrence Davis\*  
BBN Laboratories  
BBN Systems and Technologies Corporation  
10 Moulton Street  
Cambridge, MA 02238

### 1 Introduction

Running a genetic algorithm entails setting a number of parameter values. Finding settings that work well on one's problem is not a trivial task; if poor settings are used, a genetic algorithm's performance can be severely impacted. This paper deals with a new technique for setting the probabilities of applying genetic operators during the course of a run. The technique involves adapting the operator probabilities based on their observed performance as the run takes place. In the sections that follow, I shall summarize prior work in determining reasonable operator probabilities, describe a mechanism for adapting operator probabilities, describe experiments measuring the performance of the new technique, and describe a use of the technique in its first real-world application.

### 2 Finding Good Parameter Settings Can Be Hard

The problem dealt with in this paper is that of determining the probability that a given genetic operator will be applied during reproduction. Setting operator probabilities correctly can be very important when one has limited resources with which to solve a problem or when one is comparing the performance of a genetic algorithm with the performance of other types of algorithms on the same problem. Two useful strategies have been employed by researchers in the genetic algorithm field to find good operator probabilities. One was contained in Ken DeJong's thesis work [DeJong 1975]; the other was described by John Grefenstette in [Grefenstette 1986].

The problem DeJong and Grefenstette were attacking was that of determining robust parameter settings for population size, operator probabilities, and evaluation normaliza-

\*Part of this work was supported by a contract administered by Sam Wilson, of the SITS Laboratory at Point Mugu. The paper has benefited from discussions with Rich Sutton, Gilbert Syswerda, Craig Shaefer, Wayne Mesard, and other members of BARGAIN, an informal group of researchers in the Boston area. The paper has also benefited from discussions with Lashon Booker and John Grefenstette at NRL.

tion techniques. The problem of adapting genetic algorithm parameters in general was simplified, in that the only chromosomal representation technique used was the bit string, and only two operators were considered — mutation and crossover. This is a simpler problem than that of finding robust parameters for genetic algorithms across all representations, across the range of all possible operators, and so forth. Yet, even stripped down to the simple form just given, the problem is not easy to solve.

A first difficulty is that even in this stripped-down form, the problem is not precisely specified. We are asking for good values for the parameters, but measuring how good such values are is non-trivial, since the range of potential applications of genetic algorithms using binary representations is of infinite size. DeJong solved this difficulty by simplifying the problem more: he used a test suite of function optimization problems compiled by researchers in the field of function optimization theory to stand in for the full range of possible domains. DeJong's choice of functions in the test suite was a good one — fourteen years later, that suite is still the one researchers use.

A second difficulty in solving this optimization problem given a test suite is that one's evaluation function is noisy. Genetic algorithms are stochastic, and the same parameter settings used on the same problems by the same genetic algorithm generally yield different results. A consequence of this fact is that it can take a tremendous amount of computer time to find good parameter settings across a number of problems. When DeJong published his parameter settings, they were used widely by other researchers for just this reason. It could take longer to derive parameter values tailored to one's problem than the time available for solving the problem itself. Thus, DeJong's research was of immense benefit to people using genetic algorithms because it provided them with parameter values that were robust, in the sense that they had been validated on a number of different problems of different types.

The problem of noise in the evaluation function meant that DeJong was forced to derive his parameter values by hand. Several years later, however, as computing resources had become cheaper, Grefenstette derived new values for