

Three Interconnected Parameters for Genetic Algorithms

Pedro A. Diaz-Gomez^{*}
Department of Computing & Technology
School of Science and Technology
Cameron University
Lawton, OK USA

p Diaz-go@cameron.edu

Dean F. Hougen
Robotics, Evolution, Adaptation, and Learning
Laboratory (REAL Lab)
School of Computer Science
University of Oklahoma
Norman, OK USA

hougen@ou.edu

ABSTRACT

When an optimization problem is encoded using genetic algorithms, one must address issues of population size, crossover and mutation operators and probabilities, stopping criteria, selection operator and pressure, and fitness function to be used in order to solve the problem. This paper tests a relationship between (1) crossover probability, (2) mutation probability, and (3) selection pressure using two problems. This relationship is based on the schema theorem proposed by Holland and reflects the fact that the choice of parameters and operators for genetic algorithms needs to be problem specific.

Categories and Subject Descriptors

I.2.6 [Artificial Intelligence]: Learning—*parameter learning*

General Terms

Algorithms, Experimentation, Measurement, Performance, Theory, Verification

Keywords

Performance Analysis, Empirical Study, Evolution Dynamics, Genetic Algorithms, Theory, Working Principles of Evolutionary Computing, Parameter Tuning, Schema Theorem

1. INTRODUCTION

This paper looks at three genetic algorithm (GA) parameters: (1) crossover probability, (2) mutation probability, and (3) selection pressure. Through an empirical study based on Holland's schema theorem [11] we demonstrate that these

^{*}Conducting research at the Robotics, Evolution, Adaptation, and Learning Laboratory (REAL Lab), School of Computer Science, University of Oklahoma

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'09, July 8–12, 2009, Montréal, Québec, Canada.
Copyright 2009 ACM 978-1-60558-325-9/09/07 ...\$5.00.

parameters are interdependent and should be selected in conjunction with one another rather than selected independently.

The crossover operator combines portions of existing chromosomes with one another to create new chromosomes that are potentially more fit. However, crossover can often be destructive, that is, result in new chromosomes that are less fit than the originals. Similarly, the mutation operator gives the possibility of local changes that can improve fitness and also increase diversity at the level of the allele. However, mutations can also be destructive. Because crossover and mutation can be either constructive or destructive, a GA with just crossover and mutation would be a random walk.

Selection is the operator that drives the search for possible solutions. However, high selection pressure may cause the algorithm to converge prematurely to a suboptimal solution while low selection pressure may cause the algorithm to waste resources by repeatedly cycling over the same poor-quality solutions [1]. Furthermore, what constitutes high and low selection pressure cannot be stated independent of the crossover and mutation probabilities used nor can it be stated without considering characteristics of the particular optimization problem to be solved using the GA.

Given this, the question becomes how one should set selection pressure (s_p), probability of crossover (p_c) and probability of mutation (p_m) so that a GA can function effectively for a particular problem. While a method for choosing exact values of these three parameters remains elusive, we propose in this paper the use of a formula based on the schema theorem to find ranges of possible values for these parameters that are likely to help a GA to obtain better solution quality within a given number of evaluations.

2. THE SCHEMA THEOREM AND THE TRIPLE PARAMETER HYPOTHESIS

A schema ξ represents a set of $n(\xi, t)$ chromosomes at step t in the evolutionary process. A schema is characterized by l , the length of the chromosome; $h(\xi)$, the number of *defining bits* of the schema which indicates the number of bits that must have particular values for the schema to be instantiated by a given chromosome; and $d(\xi)$, the *distance* between the defined bits that are farthest apart. For example, if one has a binary GA (which has bits for alleles, that is, allele values 0 or 1) and $\xi = 1**0*$ (where '*' indicates an undefined bit, that is, where the allele value could be either 0 or 1 and still fit the schema), then $l = 5$, $h(\xi) = 2$, and $d(\xi) = 3$.

The schema theorem is simplified by Goldberg and Sastry [9] to be equivalent to

$$E(\xi, t+1) \geq \left(1 - p_c \frac{d(\xi)}{l-1}\right) s_p n(\xi, t) \quad (1)$$

where $E(\xi, t+1)$ is the expected number of schema ξ at step $t+1$, $d(\xi)/(l-1)$ corresponds to the probability of disruption of schema ξ due to crossover, s_p is the selection pressure which plays the role of $\hat{\mu}_\xi(t)/\bar{\mu}(t)$ with $\hat{\mu}_\xi(t)$ the mean fitness value of schema ξ , $\bar{\mu}(t)$ the mean fitness value of all the chromosomes in the population, and $n(\xi, t)$ is the number of elements in schema ξ at step t . Note that $n(\xi, t)$ in Equation 1 should be greater than or equal to 1 for *target schema* ξ_T , which correspond to maximally fit solutions. In order to obtain a non-decreasing number of such schema, Equation 1 should satisfy

$$\left(1 - p_c \frac{d(\xi)}{l-1}\right) s_p \geq 1 \quad (2)$$

as pointed out by Goldberg and Sastry [9], which gives an upper bound on the probability of crossover with respect to selection pressure of

$$p_c \leq \frac{l-1}{d(\xi)} \left(1 - \frac{1}{s_p}\right) \quad (3)$$

with $s_p > 1$.

Building on this, the equation central to this paper (derived previously [7]), which takes into account mutation probability, is

$$p_c \leq \frac{l-1}{d(\xi)} \left(1 - \frac{1}{(1 - p_m h(\xi)) s_p}\right), \quad (4)$$

with $p_m h(\xi) \neq 1$, $p_m > 0$, $p_m \approx 0$, and $h(\xi) > 1$. Clearly, $d(\xi)$ and $h(\xi)$ depend on ξ which in turn depends on the problem being encoded for application of the GA, as does l . This is as it should be, as no one would expect a single set of parameters to be preferable for all optimization problems [19].

The limits on the parameters for Equation 4 should be noted, particularly those for p_m and s_p . In this formulation, $s_p = 1$ would indicate that the schema in question is no more likely to be selected than any other schema (that is, there is no selection in favor of the preferred schema to drive the evolutionary process), and $s_p < 1$ would indicate selection *against* the schema in question, so it is necessary to require $s_p > 1$.

Similarly, $p_m = 0$ would mean that we run the risk of never generating chromosomes that match the schema in question. For example, if we have a specific gene that has an allele value of 1 for all chromosomes in a population, then neither cloning a single chromosome nor crossing over chromosomes with one another can change that value. If a value of 0 is required for that gene for ξ_T , our GA will never find it using only cloning and crossover. Mutation, then, is the only operator that can change the value of that gene¹ (in this case to 0) so we cannot allow $p_m = 0$. However, there is no similar limit on p_c . It is true that if $p_c = 0$, then only mutation can change chromosomes, which may limit a GA's ability to extract itself from local maxima. However, such extraction is not impossible, just unlikely, and then only if

¹This assumes one is not using inversion or any other non-standard GA operator.

p_m is small. Since $p_m < 0$ and $p_c < 0$ are nonsensical, it is necessary to require $p_m > 0$ and $p_c \geq 0$.

Based on Equation 4, we put forth the following hypothesis:

TRIPLE PARAMETER HYPOTHESIS Let U be a set of triples $\langle s_p, p_c, p_m \rangle$ that satisfy Equation 4 and V be a set of triples $\langle s'_p, p'_c, p'_m \rangle$ that do not satisfy Equation 4. Then $X(G, U) \geq X(G, V)$ where $X(G, S)$ is the expected performance measured as the mean expected solution quality after a given number of generations of a genetic algorithm G using the parameters in set S .

3. EXPERIMENTAL APPROACH

To give the triple parameter hypothesis a rigorous test, we chose two very different optimization problems to which to apply it, as well as parameter sets similar to those seen in practical GA applications and existing GA research [3, 13].

3.1 Problems Tested

The two optimization problems we selected are: (1) the well-known *one-max problem* [13] and (2) the mathematically interesting *snake-in-the-box problem* [3, 4, 5, 6, 7, 17]. The former problem is, of course, an extremely simple toy problem while the latter is a highly challenging problem with real-world applications [3, 4, 5, 6, 7, 17].

These two very different problems were selected in order to test the hypothesis under very different conditions. Specifically, the one-max problem is an example of a small-schema problem as each bit contributes to the solution independently², whereas the the snake-in-the-box problem is an example of a large-schema problem since the bits define an an open, connected, chord-free path in a hypercube, which means that a change in one bit impacts the overall possible valid paths in complex ways [3, 4, 5, 6, 7, 17].

For the snake-in-the-box problem, three different cases are considered:

1. when the initial population is generated randomly,
2. when the initial population is seeded with one half of its members containing ξ_T and the other half generated randomly, and
3. when the entire initial population is seeded with ξ_T .

These three cases were considered in order to determine if the number of target schema present in the population affects whether the triple parameter hypothesis holds. In all three cases we embed a maximal-length snake from the next lower dimension in the lower half of the hypercube, as is common when applying GAs to the snake-in-the box problem, so it should be noted that it is only in the upper half of the hypercube that the cases apply.

3.2 Triple Sets Used

The parameter ranges used in our experiments were $2 \leq s_p \leq 6$ with step size 1 which gives 5 possible values for selection pressure, $0.004 \leq p_m \leq 0.018$ with step size 0.002

²While some might object to the overuse of toy problems such as one-max in GA research, it should be noted that there is no way to formulate a small-schema problem that is not, in essence, a toy problem. Small-schema problems are inherently easy to solve.

which gives 8 possible values for probability of mutation, and $0.0 \leq p_c \leq 1.0$ with step size 0.1 which gives 11 possible values for probability of crossover. All combinations of these values gives 440 triples to divide between sets U and V .

These ranges were selected to ensure broad and uniform coverage of parameter triples that satisfy Equation 4 (set U) while restricting parameter values to “reasonable” values for the triples that do not satisfy Equation 4 (set V) and providing roughly the same number of samples from each set for both problems to ensure the appropriateness of our statistical hypothesis testing methods.

Consider probability of mutation p_m . As discussed previously (Section 2), $p_m = 0$ would not satisfy Equation 4 because of the limit placed on such values to ensure coverage of the search space. However, if we were to select triples that include $p_m = 0$ in our test set V , we would bias our experiments towards concluding that the triple parameter hypothesis is true (but perhaps only for uninteresting cases of V). Likewise, if we were to include probability of mutation values in the range $0 \leq p_m < 0.004$ this would add many more triples to the set U than V for the one-max problem (since all these triples would belong to set U for that problem) whereas a sound statistical analysis of our results requires us to have a similar number of values in each set for both problems (one-max and snake-in-the-box).

As for the upper range of p_m , we could have chosen large p_m values, such as $p_m = 0.5$ or even $p_m = 1.0$ for V , since these would not satisfy Equation 4. However, such extremely high mutation rates would not be expected to lead to good GA performance, since even without knowledge of Equation 4 most practitioners and researchers would realize that extremely high mutation rates could overwhelm selection. So, to provide the hardest possible challenge to the triple parameter hypothesis, we chose p_m values for V just beyond those that belong to U . These are values that practitioners and researchers lacking knowledge of Equation 4 might well select. For example, for one-max we estimated $h(\xi) = 50$ because initially the population is generated randomly, i.e., it is expected that there are approximately 50 ones and 50 zeros in each chromosome. If $p_c = 0$ and $s_p = 2$ then $p_m = 0.01$ is the maximum value for points that satisfy Equation 4 and, at the same time, it is the lower bound for points that do not satisfy Equation 4. If $p_c = 0$ and $s_p = 6$, then $p_m = 0.017$ is upper bound for points that satisfy Equation 4 and the lower bound for points that do not. So, we chose the range for p_m in our experiments to be $0.004 \leq p_m \leq 0.018$ with step size 0.002 which means for $p_c = 0$ and $s_p = 2$, U takes the lowest four p_m values and V takes the next four with no gap between them. Similarly, for $p_c = 0$ and $s_p = 6$, U takes the lowest seven p_m values and V takes the remaining one, again with go gap between them.

Similarly, as discussed previously (Section 2), selection pressure s_p must be greater than 1 in order for a GA to perform better than a random walk. So, while we could have chosen to include triples with $s_p = 1$ in V , this would not have provided a sufficient challenge to our hypothesis to ensure that our results are useful.

Plugging the 440 triples provided by the selected ranges for p_c , p_m , and s_p into Equation 4 gives us 297 triples in U and 143 triples in V for the one-max problem and 167 triples in U and 273 triples in V for the snake-in-the-box problem. Note that the number of triples in each set differs between the two problems, since one is a short-schema problem and

the other is a long-schema problem. As discussed previously (Section 2), this is as one would expect, since different parameter sets are appropriate for different problems [19] and Equation 4 is, correspondingly, problem-specific.

One may also note that the numbers of triples in each set is not the same within either problem. Further, for one-max $|U| > |V|$ whereas for the snake-in-the-box problem $|U| < |V|$. This is also because one is a short-schema problem and the other is a long-schema problem which means that we cannot have $|U| = |V|$ for both problems and also use the same parameter sets for both. Note, for example, that if we extend the range of one parameter to increase $|V|$ for one-max, we will also necessarily increase $|V|$ for the snake-in-the-box problem. However, what is important to note here is that this imbalance in set size is not large enough to invalidate our statistical analysis.

3.3 Operators and Other Parameters

A chromosome A needs to be selected in order to undergo reproduction. The selection operator, then, gives the pressure that pushes the population toward including increasingly more of the fittest chromosomes [1] using as a measure the payoff $\mu(A)$ [11]. Various selection operators have been proposed, including proportional probability selection and tournament selection [14].

The selection operator that most closely follows Holland’s ideas is *proportional probability selection* in which a chromosome’s probability of selection is proportional to its fitness $\mu(A)$ relative to the fitness of the other chromosomes in the population. That is, if the population at time t is $\mathcal{A}(t)$, then structure $A \in \mathcal{A}(t)$ can be chosen with a probability equal to $\mu(A) / \sum \mu(A_i)$ for all $A_i(t) \in \mathcal{A}(t)$ [1]. An alternative is *q-tournament selection* in which q chromosomes $A_i(t) \in \mathcal{A}(t)$ are chosen randomly, each with equal probability, and the one that has the greatest fitness value $\mu(A_i)$ among those q is the one that is selected for reproduction.

In this paper we use *q-tournament selection* because it provides the ability to control selection pressure. With proportional probability selection, as the algorithm iterates it is expected that selection pressure will decrease because the algorithm looks for the fittest chromosomes which begin to dominate. That is, $\hat{\mu}_{\xi_T}(t) / \bar{\mu}(t)$ will tend toward 1 as ξ_T comes to dominate the population, converging to 1 if all chromosomes in the population instantiate ξ_T . With *q-tournament selection*, the selection pressure is even over all iterations (with the possibility that if an algorithm needs a higher or lower selection pressure, then q can be increased or decreased accordingly).

Following selection comes reproduction. Here we have chosen a standard generational approach wherein all individuals in generation g are the result of reproduction of individuals found in generation $g - 1$ (except for the initial generation, which is generated randomly and/or seeded with particular chromosomes). A selected chromosome may reproduce through crossover (with probability p_c) or cloning (copied without crossover with probability $1 - p_c$). Just as there are various selection operators, there are various crossover operators available. For operating on fixed-length chromosomes, crossover operators include one-point, two-point, n -point, and uniform crossover [14]. Here we use one-point crossover, in order to follow the analysis of Holland [11], although extensions to Holland’s analysis could be followed for other forms of crossover (e.g., [18]).

Following reproduction, mutation may take place. With binary GAs, mutation only has the possibility of converting one or more bits to the opposing value (0 to 1 and/or 1 to 0) but how those bits are chosen provides for different mutation operators [14]. Here we select bits for mutation individually at a rate reported as p_m and if a bit is selected for mutation, its value is changed to the opposing value, which follows Holland’s analysis as well [11].

3.4 Problem-Specific Parameters

Besides the triple parameters under investigation in this research, other parameters need to be specified in order to run the GA. We chose to follow parameters suggested in the literature by other authors so as not to bias our results in favor or against the three parameter hypothesis.

For one-max we set population size $N = 100$, chromosome length $l = 100$, and stop criteria to be when the GA reaches the global maximum or a maximum of 20 generations. A similar set of parameters was suggested by Lobo and Goldberg [13]. They found that the algorithm usually converges to the global maximum in 25 generations. We reduced the number of generations to 20, in order to see the effect of s_p , p_c , and p_m (avoiding ceiling effects) while keeping all other parameters the same.

For the snake-in-the-box problem we chose to look at an 8-dimensional hypercube which means that the chromosome length is 2^8 bits. Within that hypercube we embedded a maximal-length (50 link) snake from a 7-dimensional hypercube. We set population size $N = 1,000$ and stop criterion to be when the GA finds a maximal-length snake in 8 dimensions or a maximum of 100 generations. A similar set of parameters was suggested by Bitterman [3]. We only reduced the number of generations to 100 in order to see the effect of s_p , p_c , and p_m (avoiding ceiling effects).

3.5 Data Collection and Analysis Methods

To collect sufficient data for a meaningful statistical analysis, we run the GA 10 times for each triple in U and V , recording the best solution found in the last generation for each run. We then calculate the mean, standard deviation, maximum, minimum, and quartiles for the sets U and V , and perform both Kolmogorov-Smirnov (K-S) and analysis of variance (ANOVA) tests on the data. Sets U and V are compared using the K-S test in order to corroborate that the two sets U and V belong to statistically different distributions and the ANOVA test checks the impact of each parameter s_p , p_c , and p_m independently, in pairs, and the three together, on the quality of the solution.

The ANOVA test shows the effect of the independent variables over the dependent variable, as well as if there are some interactions of the independent variables that affect the dependent variable. In order to perform the analysis of variance, it is useful to define a focal independent variable, a first order moderator independent variable, and a second order moderator independent variable [12]. It is a common belief that the crossover operator is the principal factor of influence in GAs and that the mutation operator is a background operator [11]. So, we propose the crossover operator as the focal independent variable, the mutation operator as the first order moderator, and the selection pressure as the second order moderator. This helps us to analyze the three way interaction that could exist between crossover, mutation, and selection pressure.

4. RESULTS

All results for both optimization problems, including all three cases for the snake-in-the-box problem, are consistent with the triple parameter hypothesis.

4.1 Results for One-Max

Table 1 shows the mean, standard deviation, maximum, minimum, and quartiles for sets U and V . All three quartiles were higher for U than for V . The p value of the K-S test was 9.07^{-40} , the H value was 1, and the D value was 0.22 showing that the two data sets differs significantly. Figure 1 shows that U outperforms V .

Table 2 shows the ANOVA results for this hypothesis. Each operator, p_c , p_m , and s_p independently; each pair of operators; and the three operators together, have a statistically significant impact on the quality of the solution at much greater than the 95% confidence level. Clearly s_p moderates the impact of p_m on the relationship between p_c and the quality of the solution.

Data Set	Mean	Stdv.	Min	Q ₁	Q ₂	Q ₃	Max
U (satisfies Eq 4)	87.39	5.27	70	84	88	91	98
V (does not satisfy Eq 4)	85.20	5.16	73	82	85	89	96

Table 1: Statistics for one-max.

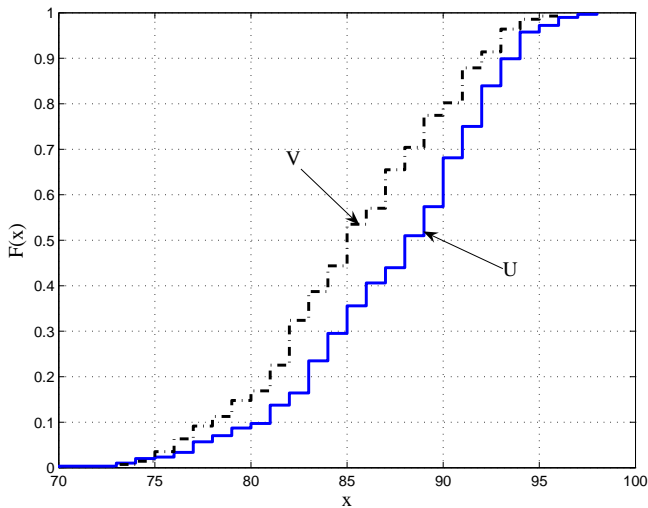


Figure 1: Empirical cumulative distributions of U and V for one-max where x corresponds to fitness.

Source	Sum. Sq.	d.o.f.	Mean Sq.	F	Prob > F
p_m	1,115.4	7	159.3	9,225.38	0
p_c	57,486.3	10	5,748.6	332,815.17	0
s_p	54,477.6	4	13,619.4	788,491.73	0
$p_m * p_c$	2,605.1	70	37.2	2,154.63	0
$p_m * s_p$	1,674.1	28	59.8	3,461.53	0
$p_c * s_p$	1,186.5	40	29.7	1,717.35	0
$p_m * p_c * s_p$	6,327.8	280	22.6	1,308.38	0
Error	68.4	3,960	0	-	-
Total	124,941.3	4,399	-	-	-

Table 2: ANOVA for one-max.

4.2 Results for Snake-in-the-Box

Tables 3, 4, and 5 show the mean, standard deviation, maximum, minimum, and quartiles for both data sets for each case. The corresponding K-S test results are: case 1 $p < 5.50^{-282}$, $D = 0.557$; case 2 $p < 1.78^{-292}$, $D = 0.568$; case 3 $p < 4.47^{-294}$, $D = 0.569$. This shows that in all cases the data sets U and V are statistically significantly different. Figures 2, 3, and 4 show that U outperforms V for all cases.

Data Set	Mean	Stdv.	Min	Q ₁	Q ₂	Q ₃	Max
U (satisfies Eq 4)	55.92	11.67	1	53	56	61	76
V (does not satisfy Eq 4)	38.80	19.41	1	29.5	49	52	75

Table 3: Statistics for snake-in-the-box case 1.

Data Set	Mean	Stdv.	Min	Q ₁	Q ₂	Q ₃	Max
U (satisfies Eq 4)	57.07	7.40	1	53	56	61	79
V (does not satisfy Eq 4)	45.09	11.62	1	39	49	52	71

Table 4: Statistics for snake-in-the-box case 2.

Data Set	Mean	Stdv.	Min	Q ₁	Q ₂	Q ₃	Max
U (satisfies Eq 4)	57.31	7.61	1	53	56	61	79
V (does not satisfy Eq 4)	44.95	11.88	1	39	49	53	70

Table 5: Statistics for snake-in-the-box case 3.

Table 6 shows ANOVA results for case 1. There is a statistically significant impact of each operator alone, and pairwise in the quality of the solution, but not of the three operators together.

Source	Sum. Sq.	d.o.f.	Mean Sq.	F	Prob > F
p_m	454,114.9	7	64,873.6	433.92	0
p_c	3,294	10	329.4	2.2	0.0151
s_p	327,984.6	4	81,996.1	548.45	0
$p_m * p_c$	15,165.7	70	216.7	1.45	0.009
$p_m * s_p$	112,379	28	4,013.5	26.85	0
$p_c * s_p$	10,007.4	40	250.2	1.67	0.0051
$p_m * p_c * s_p$	43,609	280	155.7	1.04	0.3101
Error	592,037	3,960	149.5	-	-
Total	1,558,591.6	4,399	-	-	-

Table 6: ANOVA for snake-in-the-box case 1.

Table 7 shows ANOVA results for case 2. There is a statistically significant impact of each operator alone, pair-wise, and all the three together on the quality of the solution.

Source	Sum. Sq.	d.o.f.	Mean Sq.	F	Prob > F
p_m	217,740.3	7	31,105.8	915.44	0
p_c	2167	10	216.7	6.38	0
s_p	188,113.7	4	47,028.4	1,384.04	0
$p_m * p_c$	7,274.7	70	103.9	3.06	0
$p_m * s_p$	21,234.9	28	758.4	22.32	0
$p_c * s_p$	4,552.3	40	113.8	3.35	0
$p_m * p_c * s_p$	13,325.5	280	47.6	1.4	0
Error	131,566.9	3,872	34	-	-
Total	583,200.1	4,311	-	-	-

Table 7: ANOVA for snake-in-the-box case 2.

Table 8 shows ANOVA results for case 3. There is a statistically significant impact of each operator alone, and pairwise in the quality of the solution, but not of the three operators together.

Source	Sum. Sq.	d.o.f.	Mean Sq.	F	Prob > F
p_m	238,216.4	7	34,030.9	869.8	0
p_c	3,981.6	10	398.2	10.18	0
s_p	198,001.4	4	49,500.4	1,265.19	0
$p_m * p_c$	7,870.6	70	112.4	2.87	0
$p_m * s_p$	21,737	28	776.3	19.84	0
$p_c * s_p$	2,807	40	70.2	1.79	0.0016
$p_m * p_c * s_p$	12,186.2	280	43.5	1.11	0.1035
Error	154,934.6	3,960	39.1	-	-
Total	639,735.2	4,399	-	-	-

Table 8: ANOVA for snake-in-the-box case 3.

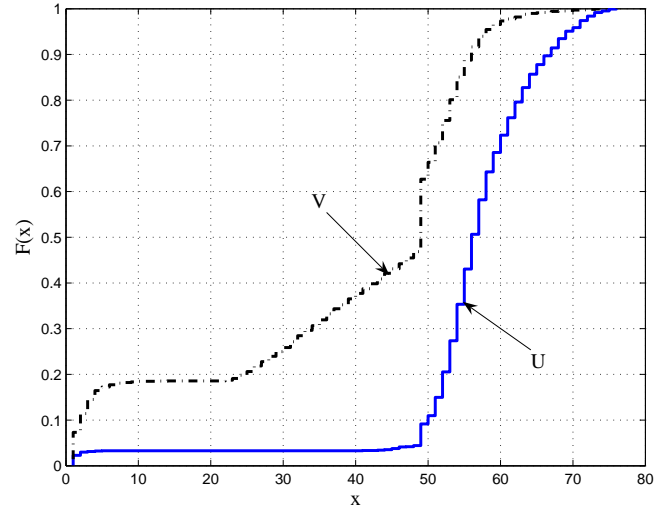


Figure 2: Empirical cumulative distributions of U and V for snake-in-the-box case 1.

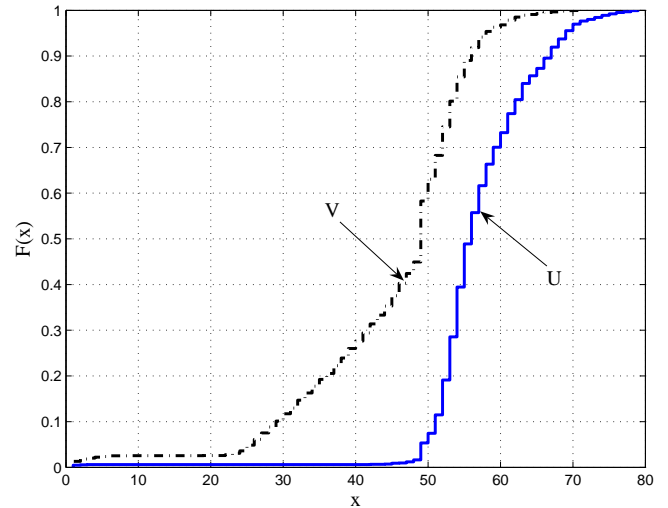


Figure 3: Empirical cumulative distributions of U and V for snake-in-the-box case 2.

Figures 5 to 9 show crossover and mutation probability versus mean solution quality for different selection pressures for snake-in-the-box case 1.

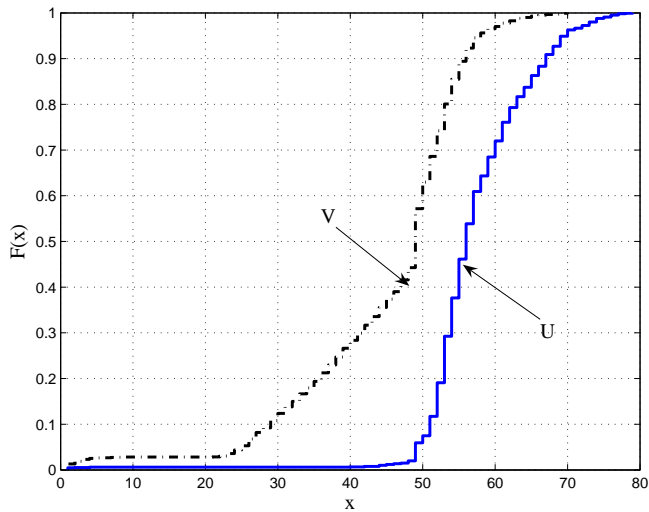


Figure 4: Empirical cumulative distributions of U and V for snake-in-the-box case 3.

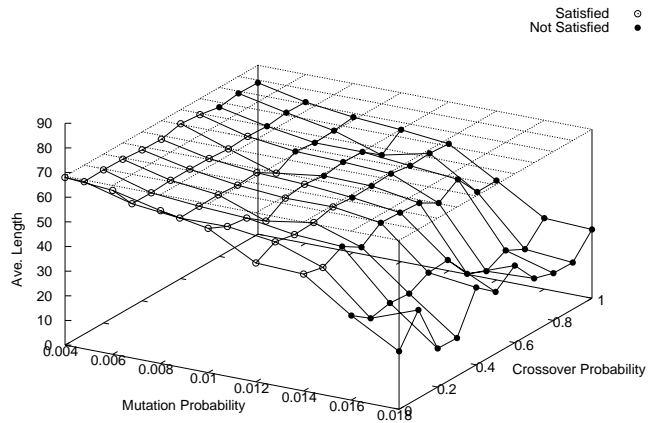


Figure 7: p_c and p_m vs. solution quality with $s_p = 4$ for snake-in-the-box case 1.

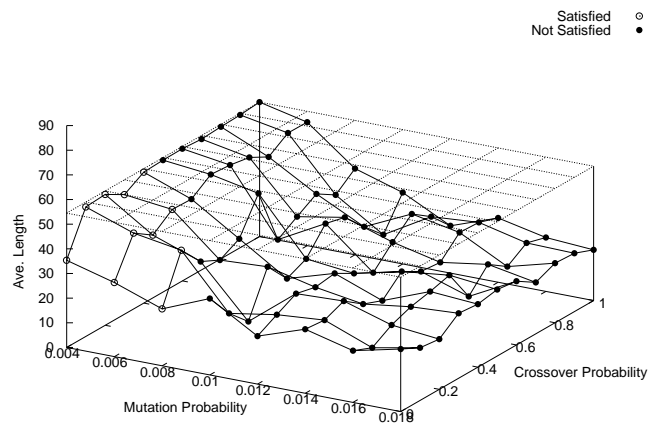


Figure 5: p_c and p_m vs. solution quality with $s_p = 2$ for snake-in-the-box case 1.

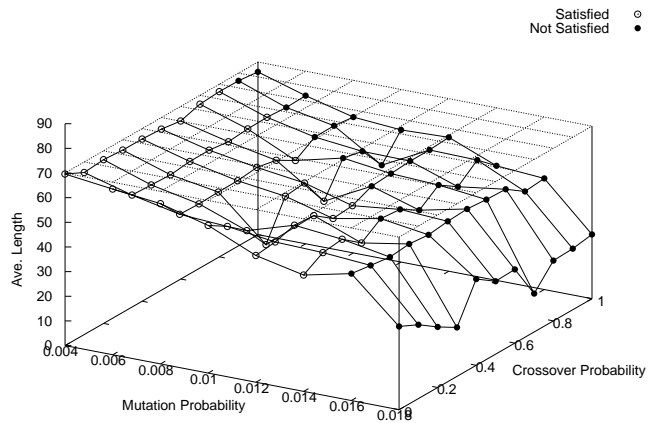


Figure 8: p_c and p_m vs. solution quality with $s_p = 5$ for snake-in-the-box case 1.

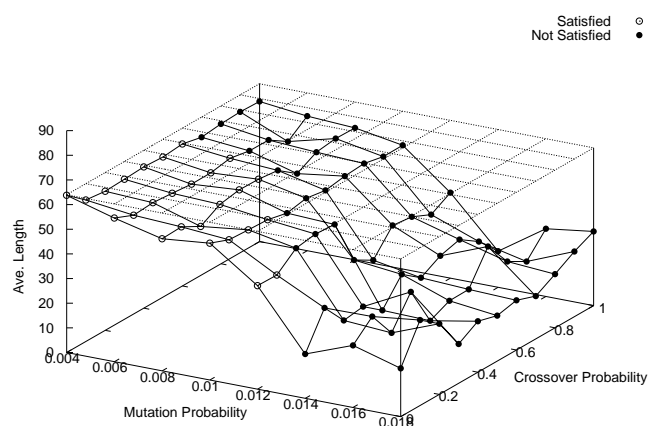


Figure 6: p_c and p_m vs. solution quality with $s_p = 3$ for snake-in-the-box case 1.

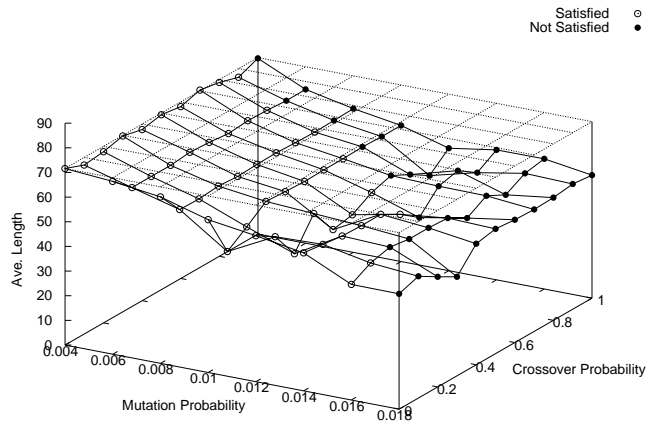


Figure 9: p_c and p_m vs. solution quality with $s_p = 6$ for snake-in-the-box case 1.

5. DISCUSSION

It has been common practice to use a default set of parameters for GAs, or, in some cases a trial and error procedure is used [8]. More recently, approaches that systematically test parameter sets have been proposed [2, 15]. Nonetheless, parameter tuning remains a substantial problem for researchers in evolutionary computation [16]. The hypothesis tested in this paper addresses this problem and uses a relationship between p_c , p_m , and s_p as in Equation 4 [7]. Equation 4 shows that the selection of p_c , p_m , and s_p depends on the problem itself (l , $d(\xi)$, and $h(\xi)$) and that there is an interrelation between p_c , p_m , and s_p that could affect the quality of the solution.

Equation 4 was derived from the schema theorem as described in previous work [7], where it is expected that the number of target schema ξ_T at step t is not going to decrease at step $t + 1$. It is very unlikely that ξ_T is present in the initial population for all but trivial problems, so the GA has to build it progressively at each time step. However, in the building process, destructive factors occur owing to crossover and mutation, so the probabilities of crossover and mutation should be such that they minimize that destructive effect to a level that allows quality chromosomes to be accumulated through selection. This is what the triple parameter hypothesis is proposing with Equation 4. However, an optimal setting of parameters for GAs is very difficult to find [8] so the triple parameter hypothesis proposes something more modest: a set U of possible values from which to draw values for the triple parameters of p_c , p_m , and s_p , based on the optimization problem to be solved.

Two problems were chosen to test this hypothesis: the one-max problem and the snake-in-the-box problem. The one-max problem is such that the schema under consideration is small ($d(\xi_T) = 1$) compared to the chromosome length (we chose $l = 100$) and the snake-in-the-box problem in a 8-dimensional hypercube is such that the schema could be considered long ($d(\xi_T) = 110$) relative to the chromosome length ($l = 127$) for an 8-dimensional hypercube because a 50-link snake that begins at node 0 and finishes at node 110 was embedded in the lower hypercube. The algorithm needs, then, to preserve the sub-snake given and look at the 128 nodes of the upper hypercube to complete it.

The set of triples (p_c, p_m, s_p) chosen to perform the tests of specific hypotheses were such that there is almost no boundary between the points that satisfy Equation 4 (set U) and the points that do not satisfy Equation 4 (set V), which ensures a very challenging test of the hypothesis. The probability of crossover could be set in the full range $0.0 \leq p_c \leq 1.0$. The probability of mutation began at 0.004 because for the case of the one-max problem all the points with $p_m \leq 0.004$ belong to set U (330 in total independent of p_c and s_p), and we wish to have sets of approximately equal size. For example, for $s_p = 2$, the points that belong to V begin at $p_m = 0.010$, so the range for p_m was between 0.004 and 0.018.

For the one-max problem, as the schema is small, it is expected that the crossover operator is not destructive. This fact is so strong that the algorithm performs quite well with p_c high in both sets U and V . However, on average set U that obeys Equation 4 outperforms set V that does not obey Equation 4 as is shown in Table 1 in Section 4.1.

Table 2 in Section 4.1 shows the ANOVA results for the one-max problem. Each operator p_c , p_m , and s_p alone; each

pair-wise set of operators; and the three operators together impact the quality of the solution and s_p moderates the impact of p_m on the relationship between p_c and the quality of the solution.

For snake-in-the-box case 1, where there is no presence of ξ_T , s_p plays its roll where it is more beneficial for the GA to have higher selection pressures for both sets U and V (see Figure 5 for selection pressure of 2 and Figure 9 for selection pressure of 6). Theoretically, p_c should be quite destructive but initially ξ_T is not present, so p_c plays its roll of building ξ_T . However, it should be taken into account that for $3 \leq s_p \leq 6$ on average the maximum solution quality was obtained when $p_c = 0.0$ and $p_m = 0.004$ (a point belonging to set U). Table 3 shows the statistics for set U and V . On average, set U outperformed set V .

Table 6 shows the analysis of variance for snake-in-the-box case 1. There is a statistically significant impact of each operator alone and in pair-wise combinations on the quality of the solution, but not the three operators together. The ANOVA test shows, then, that the interaction effect between s_p , p_c , and p_m together on the dependent variable is not statistically significant. It is possible that the cause is that ξ_T is not initially present so it is expected that the majority of individuals are going to have approximately the same fitness value. The effect is that the moderation of s_p 's on the impact of p_m on the relationship between p_c and the quality of the solution is not statistically significant.

For snake-in-the-box case 2, half of the initial population definitely has ξ_T and half are generated randomly. In this case, the crossover operator could be destructive if an instance of ξ_T is selected to mate with a chromosome that was randomly generated. However, this possibility is reduced by the fact that ξ_T probabilistically tends to be selected (ξ_T is more fit) so ξ_T will often be crossed with ξ_T . This time again, on average the maximum mean solution quality was obtained when $p_c = 0$ and $p_m = 0.004$, and set U outperformed set V as is shown in Table 4.

Table 7 shows the analysis of variance for snake-in-the-box case 2. There is a statistically significant impact of each operator alone, in pair-wise combinations, and all the three together on the quality of the solution. The ANOVA test shows, then, that there is a statistically significant correlation between the operators p_c , p_m , and s_p that impact the quality of the solution, and it is possible that the cause is that ξ_T is initially present. In this case, ξ_T competes with different schema and the three operators p_c , p_m , and s_p interact together to impact the quality of the solution.

For snake-in-the-box case 3, where all individuals in the initial population have ξ_T , the disruptive factor of p_c on ξ_T is not present. This does not mean that s_p has no impact on the solution quality because the algorithm nonetheless has to look for a snake in the 8-dimensional hypercube that contains the sub-snake given. This time again, on average the maximum mean solution quality was obtained when $p_c = 0$ and $p_m = 0.004$, and set U outperformed set V as is shown in Table 5. For cases where ξ_T is already present, the mutation operator could be quite destructive, such as a change of a bit in the $h(\xi) = 50$ positions already defined because of the embedded snake causes ξ_T to be destroyed.

Table 8 shows the analysis of variance for snake-in-the-box case 3, where the initial population has ξ_T in all chromosomes. There is a statistically significant impact of each operator alone and in a pair-wise combinations on the qual-

ity of the solution. However, there is no statistically significant impact looking at the three independent variables together. It is expected that if all the individuals already have ξ_T , then they are going to have the same fitness values, making it harder for s_p to moderate the impact of p_m in the relationship between p_c and the solution quality.

6. CONCLUSIONS AND FUTURE WORK

Numerous factors need to be set when solving a problem with GAs including the fitness function³, population size, selection operator and pressure, crossover probability, mutation probability, and stop criteria. Setting of these factors constitutes an optimization problem [6, 8]. Any set of these factors can lead the algorithm to a different solution quality [10]. Therefore, it is expected that the time spent in the selection of these factors could be compensated for in the performance of the GA, where performance is measured in solution quality and possibly in the number of generations to obtain a solution [8]. This is only an expectation, of course, because GAs are, at bottom, a stochastic search method which means that even using the same parameter choices repeatedly will almost surely produce some variability in the performance obtained. There is no guarantee, then, even that the best parameter choices will outperform the worst. Note, for example, that an optimal solution may be present in the initial population regardless of parameter setting. Better parameter choices can only be defined, then, in terms of the probability of obtaining better performance. What this paper provides is a guideline to help determine whether one parameter triple is likely to provide better performance than another.

The relationship between the elements of $\langle s_p, p_c, p_m \rangle$ as in Equation 4 in effect tested the schema theorem, which usually is not considered in the design of GAs.

Future research includes empirically testing the triple parameter hypothesis using additional optimization problems, defining schema diversity, investigating the use of new operators and the possible importance of the mutation operator in the building of better solutions where the crossover operator is quite destructive, the possible adaptation of the triple $\langle s_p, p_c, p_m \rangle$ according to Equation 4 as the GA runs, and including additional parameters such as population size in the investigation.

7. REFERENCES

- [1] T. Bäck. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, 1996.
- [2] T. Bartz-Beielstein, C. W. G. Lasarczyk, and M. Preuss. Sequential parameter optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 773–780, 2005.
- [3] D. S. Bitterman. New lower bounds for the snake-in-the-box problem: A prolog genetic algorithm and heuristic search approach. Master’s thesis, University of Georgia, Athens, GA, USA, Dec. 2004.

³The fitness function defines the optimization problem but in a practical sense selecting a fitness function is a necessary step in the implementation of GAs. For example, in the snake-in-the-box problem, multiple fitness functions try to find the longest snake, but each one influences the finding of a better solution [5].

- [4] D. A. Casella and W. D. Potter. New lower bounds for the snake-in-the-box problem: Using evolutionary techniques to hunt for snakes. In *Proceedings of the Florida Artificial Intelligence Research Society Conference*, pages 264–268, 2004.
- [5] P. A. Diaz-Gomez and D. F. Hougen. Hunting for snakes in hypercubes using genetic algorithms. *ACIS International Journal of Computer & Information Science*, 7(1-4):1–12, 2006.
- [6] P. A. Diaz-Gomez and D. F. Hougen. Initial population for genetics algorithms: A metric approach. In *Proceedings of the International Conference on Genetic and Evolutionary Methods*, pages 43–49, 2007.
- [7] P. A. Diaz-Gomez and D. F. Hougen. A trade-off of the schema theorem. In *Proceedings of the Conference on Artificial Intelligence and Pattern recognition*, pages 326–333, 2007.
- [8] A. E. Eiben, R. Hinterding, and Z. Michalewicz. Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 3(2):124–141, 1999.
- [9] D. E. Goldberg and K. Sastry. A practical schema theorem for genetic algorithm design and tuning. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 328–335, 2001.
- [10] J. J. Grefenstette. Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-16(1):122–128, 1986.
- [11] J. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.
- [12] J. Jaccard. *Interaction effects in factorial analysis of variance*. Number 07-118 in Quantitative Applications in the Social Sciences. SAGE Publications, Inc, Thousand Oaks, CA, USA, 1998.
- [13] F. G. Lobo and D. E. Goldberg. The parameter-less genetic algorithm in practice. *Information Sciences: Informatics and Computer Science*, 167(1-4):217–232, 2004.
- [14] M. Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, 1998.
- [15] V. Nannen and A. E. Eiben. A method for parameter calibration and relevance estimation in evolutionary processes. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 183–190, 2006.
- [16] V. Nannen, S. K. Smit, and A. E. Eiben. Costs and benefits of tuning parameters of evolutionary algorithms. In *Proceedings of the Conference on Parallel Problem Solving from Nature*, pages 528–538, 2008.
- [17] W. D. Potter, R. W. Robinson, J. A. Miller, K. Kochut, and D. Z. Redys. Using the genetic algorithm to find snake-in-the-box codes. In *Proceedings of the International Conference on Industrial & Engineering Applications of Artificial Intelligence and Expert Systems*, pages 421–426, 1994.
- [18] W. M. Spears and K. A. D. Jong. An analysis of multi-point crossover. In *Foundations of Genetic Algorithms*, pages 301–315, 1990.
- [19] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, Apr. 1997.