Linear Selection

Mario Graff
mgraff@essex.ac.uk

Riccardo Poli
rpoli@essex.ac.uk

Alberto Moraglio
amoragn@essex.ac.uk

ABSTRACT

In this paper we investigate a form of selection where parents are not selected independently. We show that a particular form of dependent selection, linear selection, leads a genetic algorithm with homologous crossover to become very similar to a genetic algorithm with standard (independent) selection and headless chicken crossover, i.e., it turns crossover into a type of mutation. In the paper we analyze this form of selection theoretically, and we compare it to ordinary selection with crossover and headless chicken crossover in real runs.

1. INTRODUCTION

Different selection methods have been analyzed mathematically in depth in the last decade or so. The main emphasis of previous research has been the takeover time [4], i.e., the time required by selection to fill up the population with copies of the best individual in the initial generation, and the evaluation of the changes produced by selection on the fitness distribution of the population [2, 3, 7]. In this second line of research, the behavior of selection algorithms is characterized using the loss of diversity, i.e., the proportion of individuals in a population that are not selected.

Starting from some simple observations on the sampling behavior of tournament selection, in [9, 8] it was shown that this is a possible source of inefficiency in EAs. This previously unknown phenomenon has very deep implications, its analysis effectively led to a completely new class of EAs – the backward-chaining EA – which is more powerful and closer in spirit to classical AI techniques than traditional EAs. In addition, this analysis was used in [13] to define new forms of tournament selection that would not suffer from the this phenomenon.

These theoretical studies are very comprehensive and appeared to have completely characterized selection, fundamentally making it a largely understood process. However, all theoretical studies have considered forms of selection where the parent individuals are selected independently. Naturally, some forms of selection where this is not the case have been considered by practitioners. For example, [12] introduced the notion of tournament selection without replacement, which effectively induces a small dependency in the selection of individuals. Here, however, we want to study much more extreme forms of dependent selection. In particular, we are interested in understanding the effects of the interactions between such forms of selection and crossover operators.

When crossover is used, two parents need to be selected. These are typically drawn independently, so the probability of a pair of parents (x, y), is given by the product of their selection probabilities, i.e., $p(x, y) = p(x)p(y)$. In some selection schemes, such as the one originally proposed in [5], one parent is selected based on fitness, while the second is randomly picked from the population. In this case, the probability of selecting it is simply given by the frequency, $\phi(x)$, of such individual in the population, so $p(x, y) = p(x)\phi(y)$. However, in principle, any assignment of $p(x, y)$ such that $p(x, y) \geq 0$ and $\sum_x \sum_y p(x, y) = 1$ would be an acceptable form of joint parent selection. Any such form of selection would also be implementable, albeit not very efficiently. Given a population P one would just need to create a new population, P^2, of pairs of individuals (effectively the Cartesian product of P with P), associate to each pair a virtual fitness $p(x, y)$, and then select pairs via roulette wheel selection.

Not all such forms of joint selection would make sense, however. So, it is natural to start by asking whether meaningful ways of performing the joint selection of two parents based on $p(x)$, $p(y)$, $\phi(x)$ and/or $\phi(y)$ other than via a product formula. In this paper we study the following forms:

- The simplest of such combinations is where a pair of parents is selected based on the straight average of the selection probabilities of the parents. That is we consider the case

$$p(x, y) = \frac{p(x) + p(y)}{2} \cdot \alpha^{-1} \tag{1}$$

  where α is a normalization factor such that $\sum_{x,y} p(x, y) = 1$. We will term this form of selection *pure linear selection.*

- As we will explain in the following section we also consider a second form of linear selection, *semi-linear selection*, which has the following form:

$$p(x, y) = \frac{p(x) + p(y)}{2\alpha} \delta_x \delta_y \tag{2}$$

where δ_x is 1 if x is in the population and 0 otherwise (likewise for δ_y).

- We also consider

$$p(x, y) = \frac{F(x) + F(y)}{2}\phi(x)\phi(y) \cdot \alpha^{-1} \qquad (3)$$

where, again α is a normalization factor, and $F(x)$ is a function of the fitness of individual x (but does not necessarily coincide with it). Likewise for $F(y)$. We call this form of selection *Holland's selection* for reasons that will become clear later.

Surprisingly, as we will see in the following, we find that linear and semi-linear selection lead a genetic algorithm with ordinary (homologous) crossover to become very similar to a GA with standard (independent) selection and headless chicken crossover [6, 1]. Headless chicken crossover is a form of crossover where an individual selected from the population is crossed-over with a randomly created individual. So, with most forms of crossover used in standard GAs operating on fixed-lenght strings (e.g., uniform crossover, one- and multi-point crossover, etc.), each application of headless chicken crossover introduces 50% random material in the offspring. That is, unexpectedly linear selection effectively transforms crossover into a type of adaptive macro mutation.

Holland's selection, instead, is surprising for a different reason. It is provably identically to the selection method used by Holland [5] who, in a selecto-recombinative GA, selected the first parent based on fitness and chose the second parent randomly and uniformly from the population. This is in fact the reason why we gave this name to the selection scheme in Equation (3).

This article is organized as follows. In Section 2 we provide a more precise definition of linear and semi-linear selection and Holland's selection and derive exact evolution equations that describe the dynamics of a system with such selections and crossover in the infinite population limit. We then compare these with corresponding equations for normal selection and for headless chicken crossover in Section 3. This allows us to identify efficient algorithms to implement linear selection. In Section 4 we study the behavior of different forms of selection both by performing real runs and by integrating the infinite population evolution equation. Finally, Section 5 presents some conclusions and suggests possible avenues for future work.

2. LINEAR AND HOLLAND'S SELECTION

It is well known (e.g., see [14]), that in the infinite population limit, a genetic system with with selection and 100% crossover (i.e., $p_{xo} = 100\%$), but in the absence of mutation (i.e., $p_m = 0\%$), is governed by the following equation

$$\phi(z, t+1) = \sum_{x,y \in \Omega} p(x, y, t) p(x, y \to z) \qquad (4)$$

where $\phi(z, t+1)$ represents the frequency of individuals of type z in the next generation $(t+1)$, Ω is the search space, $p(x, y, t)$ is the probability of selecting parents x and y at generation t, and $p(x, y \to z)$ is the probability of obtaining an offspring of type z when crossing-over parents of types x and y.[1]

We could trivially specialize this equation to the form of linear selection mentioned in Section 1 by setting

$$p(x, y, t) = \frac{p(x, t) + p(y, t)}{2\alpha(t)} \qquad (5)$$

where $p(x, t)$ and $p(y, t)$ represent the selection probabilities for the parents at generation t if selected independently by normal selection. This would lead to the equation

$$\phi(z, t+1) = \sum_{x,y \in \Omega} \frac{p(x, t) + p(y, t)}{2\alpha(t)} p(x, y \to z). \qquad (6)$$

It is, however, immediately apparent that this form of selection presents an unusual feature: $p(x, y, t)$ may be non-zero even if one of the parents, say y, is absent from the population. This is because if, for example, $p(y) = 0$, Equation (5) transforms into $p(x, y, t) = \frac{p(x,t)}{2\alpha}$, which will be non-zero whenever $p(x, t)$ is non-zero.

We can correct this behavior by modifying our definition of linear selection. One way to achieve this is to ensure that $p(x, y, t)$ is zeroed whenever either x or y are not in the population. This is what led to the definition of semi-linear selection in Equation (2). There, $\delta_x(t)$ indicates whether or not $\phi(x, t)$ is zero. Note that in many forms of selection (such as fitness proportionate selection, tournament selection and rank selection) $p(x, t)$ is zero if and only if $\phi(x, t)$ is zero, and, so, effectively $\delta_x(t)$ is also an indicator of whether or not $p(x, t)$ is zero. That is, in non-greedy forms of selection have the property $p(x, t) = 0 \iff \phi(x, t) = 0$. In *all* forms of selection, however, $\phi(x, t) = 0 \implies p(x, t) = 0$. In other words, $\delta_x(t) = 0 \implies p(x, t) = 0$. We will use this property later in this section to simplify the evolution equation for a GA under semi-linear selection and crossover.

For semi-linear selection we obtain

$$\phi(z, t+1) = \sum_{x,y \in \Omega} \frac{p(x, t) + p(y, t)}{2\alpha(t)} \delta_x(t)\delta_y(t) p(x, y \to z) \qquad (7)$$

Expanding we obtain

$$\begin{aligned} \phi(z, t+1) = \frac{1}{2\alpha(t)}[&\sum_{x,y \in \Omega} p(x, t) p(x, y \to z) \delta_x(t)\delta_y(t) \\ &+ \sum_{x,y \in \Omega} p(y, t)\delta_x(t)\delta_y(t) p(x, y \to z)] \end{aligned} \qquad (8)$$

With a suitable renaming of summation variables and gathering of terms we then obtain

$$\begin{aligned} \phi(z, t+1) = \frac{1}{2\alpha(t)} &\sum_{x,y \in \Omega} p(x, t) \\ &[p(x, y \to z) + p(y, x \to z)]\delta_x(t)\delta_y(t) \end{aligned} \qquad (9)$$

Note that, for non-greedy selection, $\delta_x(t) = 0$ whenever $p(x, t) = 0$ and $\delta_x(t) = 1$ whenever $p(x, t) > 0$. Therefore $\delta_x(t)$ can be omitted from Equation (9). Also, note that if crossover is symmetric,[2] then $p(x, y \to z) = p(y, x \to z)$.

[1]Naturally, different crossover operators lead to different $p(x, y \to z)$ distributions. Since the theory presented in this paper applies to all, here we will not provide a more detailed characterization of this distribution.

[2]We obtain a symmetric crossover, if, for example, we select the parents and then randomly choose which parent to consider as the first and which as the second, or if we generate two offspring and then randomly select which one to return.

So, in these fairly general conditions Equation (9) simplifies to

$$\phi(z,t+1) = \frac{1}{\alpha(t)} \sum_{x\in\Omega} p(x,t) \sum_{y\in\Omega} p(x,y\to z)\delta_y(t) \tag{10}$$

We are now in a position to compute the value of the normalization constant $\alpha(t)$. We start by summing both sides of Equation 10 over all values of z in Ω obtaining

$$\sum_{z\in\Omega} \phi(z,t+1) = \sum_{z\in\Omega} \frac{1}{\alpha(t)} \sum_{x\in\Omega} p(x,t) \sum_{y\in\Omega} p(x,y\to z)\delta_y(t) \tag{11}$$

which can be transformed into

$$1 = \frac{1}{\alpha(t)} \sum_{x\in\Omega} p(x,t) \sum_{y\in\Omega} \delta_y(t) \sum_{z\in\Omega} p(x,y\to z) \tag{12}$$

since $\sum_{z\in\Omega} \phi(z,t) = 1$ for any t by definition. Noting that $\sum_{z\in\Omega} p(x,y\to z) = 1$ since crossover must always produce some element of Ω irrespective of the choice of parents x and y. So,

$$\alpha(t) = \sum_{x\in\Omega} p(x,t) \sum_{y\in\Omega} \delta_y(t) \tag{13}$$

The two summations in this equation commute. Noting that $\sum_{x\in\Omega} p(x,t) = 1$ we then obtain

$$\alpha(t) = \sum_{y\in\Omega} \delta_y(t) \tag{14}$$

That is $\alpha(t)$ is the number of types, i.e., distinct individuals, in the population at generation t, which not be confused with the number of individuals in the population.

So, Equations (2) and (14) completely define linear selection, while the following equation describes the dynamics of a system with linear selection and crossover:

$$\phi(z,t+1) = \sum_{x\in\Omega} p(x,t) \sum_{y\in\Omega} p_\delta(y,t) p(x,y\to z) \tag{15}$$

where

$$p_\delta(y,t) = \frac{\delta_y(t)}{\sum_{w\in\Omega} \delta_w(t)}. \tag{16}$$

Following similar calculations, for Equation 6, one can prove that, for symmetric crossover,

$$\alpha(t) = |\Omega| \tag{17}$$

and

$$\phi(z,t+1) = \sum_{x\in\Omega} p(x,t) \sum_{y\in\Omega} \frac{1}{|\Omega|} p(x,y\to z) \tag{18}$$

Similarly one can transform Equation (3), obtaining, for symmetric crossover,

$$\phi(z,t+1) = \frac{1}{\alpha(t)} \sum_{x,y\in\Omega} F(x) p(x,y\to z)\phi(x,t)\phi(y,t) \tag{19}$$

where

$$\alpha(t) = \sum_{x\in\Omega} F(x)\phi(x,t) \tag{20}$$

So, effectively we have

$$\phi(z,t+1) = \sum_{x\in\Omega} p(x,t) \sum_{y\in\Omega} \phi(y,t) p(x,y\to z) \tag{21}$$

where $p(x,t) = F(x)\phi(x)/\sum_{x\in\Omega} F(x)\phi(x,t)$ is effectively a form of fitness proportionate selection where the function F is interpreted as a fitness function (although F may be a complicated function of the actual fitness f).

In the next section we study Equations (15), (18) and (21), and compare them to the evolution equations for standard selection with crossover, headless-chicken crossover and mutation.

3. THEORETICAL COMPARISON WITH OTHER OPERATORS

It is instructive to compare Equations (15) and (18) with the evolution equations for a GA with standard selection and crossover and for a GA with standard selection and headless chicken crossover. In both cases, for simplicity we will assume that genetic operators are applied with 100% probability.

In normal selection each parent is selected independently therefore $p(x,y,t) = p(x,t)p(y,t)$, and, so, the infinite population model for a selecto-recombinative generational GA becomes

$$\phi(z,t+1) = \sum_{x\in\Omega} p(x,t) \sum_{y\in\Omega} p(y,t) p(x,y\to z) \tag{22}$$

If, instead the second parent is randomly drawn from the population (as in Holland's work), we have $p(x,y,t) = p(x,t)\phi(y,t)$, and, so, the infinite population model for a selecto-recombinative generational GA becomes

$$\phi(z,t+1) = \sum_{x\in\Omega} p(x,t) \sum_{y\in\Omega} \phi(y,t) p(x,y\to z) \tag{23}$$

The evolution equation for a GA with standard selection and headless chicken crossover were derived [10]. This is

$$\phi(z,t+1) = \sum_{x\in\Omega} p(x,t) \sum_{y\in\Omega} \pi(y,t) p(x,y\to z) \tag{24}$$

where $\pi(y,t)$ is the probability of generating a random individual of type y at generation t. Since normally the algorithm used to initialize the population is also used to generate the random parent in headless-chicken crossover, in fact, $\pi(y,t)$ is not a function of t. Also, in most GAs the initialization algorithm draws individuals randomly and uniformly in Ω. So, $\pi(y,t) = \frac{1}{|\Omega|}$. Under these conditions we then have

$$\phi(z,t+1) = \sum_{x\in\Omega} p(x,t) \sum_{y\in\Omega} \frac{1}{|\Omega|} p(x,y\to z) \tag{25}$$

This equation is identical to the evolution equation for a GA under pure linear selection (Equation (18)). That is, a GA with normal selection and headless-chicken crossover is identical to a GA with pure-linear selection and ordinary crossover. We can also see that Equation (21) and Equation (23) are identical. However, also the similarity between Equations (22) and (25) and Equation (15) is striking, the only difference between these equations really being whether $p_\delta(y,t)$, $p(y,t)$, $\phi(y,t)$ or $\frac{1}{|\Omega|}$ is used. This allows us to better understand semi-linear selection.

Firstly, we can interpret semi-linear selection as a form of independent selection, but one where the two parents are chosen using different selection schemes: the first is selected with whatever the selection algorithm is, leading to the term $p(x,t)$ in Equation (15); the second is independently selected with a new form of selection, which leads to the term $p_\delta(y,t)$. What form of selection could this be? We note that if we selected a type randomly and uniformly out of those present in the population,[3] each type would be selected with a probability of 1 over the total number of types. However, this is exactly what Equation (16) computes. So, linear selection corresponds to selecting the first parent using ordinary selection of individuals and the second using random selection of types. This gives us also a way of implementing linear selection efficiently, without requiring the creation of a population P^2 of all possible pairs of individuals, as suggested in Section 1.

Secondly, we notice the similarity between $|\Omega|$, the number of types in the search space, and the denominator of Equation (16), $\alpha(t) = \sum_{y\in\Omega} \delta_y(t)$, which computes the number of types in the population. Although it is unlikely that $\alpha(t)$ will ever approach $|\Omega|$ in any realistic situation, in a large and diverse population the selection of random types as second parents to use in crossover leads to the introduction of considerable variation in the offspring. In such conditions, semi-linear selection effectively turns into pure-linear selection, and so it turns crossover into a form of headless chicken crossover (i.e., an adaptive macro mutation).

4. RESULTS

In this section we study the behavior of linear selection both by performing real runs and by integrating infinite population evolution equations. Because pure linear selection with crossover behaves exactly as standard selection with headless chicken crossover, we will treat these two cases as one. So, whenever we refer to linear selection in this section, we will mean semi-linear selection.

We consider two problems, both are functions of unitation. The first is the Zero-max problem – a version of One-max where the objective is to maximise the number of zeros. The second is the OneMix problem, recently introduced in [11]. This function is a mixture of the OneMax problem and a ZeroMax problem. Like these it is a function of unitation, u, which represents the number of 1s in a string. For unitation values bigger than $\ell/2$, where ℓ is the bit-string length, our new function is just OneMax. For lower unitation values, it is OneMax if u is odd, a scaled version of ZeroMax, otherwise. The new function, which we call *OneMix*, is formally defined as

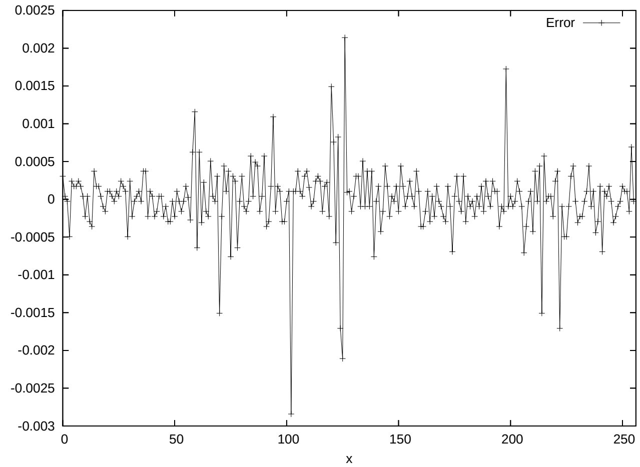
$$f(u) = \begin{cases} (1+a)(\ell/2-u)+\ell/2 & \text{if } u \text{ is even} \\ & \text{and } u < \ell/2 \\ u & \text{otherwise,} \end{cases}$$

where $a > 0$. With this constraint we ensure that the global optimum is the string $00\cdots 0$.

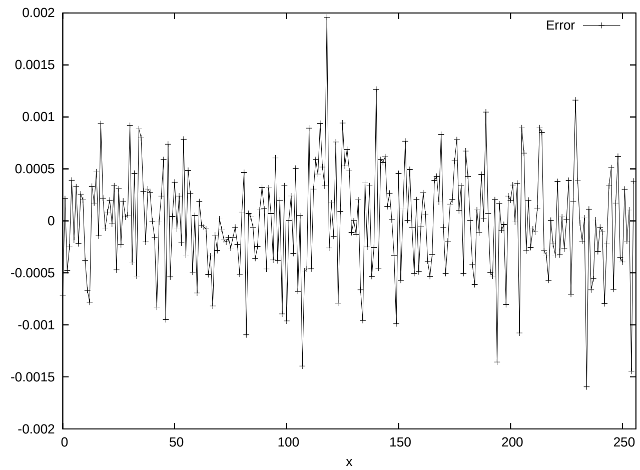
We chose these problems because they have radically different features. ZeroMax is an "easy" problem where both crossover type and mutation type search operators can do well, while the OneMix problem is known to be deceptive [11] for a GA with crossover while it is not for a GA based on mutation.

4.1 Integration of infinite population equations

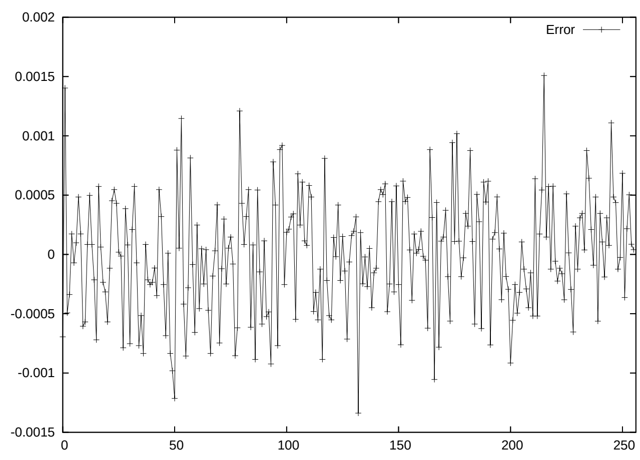
[3]This is not the same thing as selecting a random individual from the population, which would lead to a term of the form $1/M$ where M is the population size.



(a) $\phi(z,1)$



(b) $\phi(z,3)$



(c) $\phi(z,8)$

Figure 1: Differences between the value of $\phi(z,t)$ predicted integrating Equation (10) and that measured in one run of the GA with linear selection.

To corroborate Equation 10, the infinite population dynamics obtained by integrating such an equation was compared against the behavior shown in real runs of the a GA with linear selection and crossover on the following version of OneMix

$$f(u) = 1.3\ell - \begin{cases} 0.8\ell - 1.6u + \ell/2 & \text{if } u \text{ is even and } u < \ell/2 \\ u & \text{otherwise} \end{cases} \tag{26}$$

where $\ell = 8$. We use the same settings used in [11] but we treat it as a minimization problem. The population size was very large, $15,000$, compared to the search space size, 256, so as to well approximate the infinite population behavior.

As an illustration, Figure 1 presents the differences between the frequency of strings of each type, $\phi(x, t)$, estimated using Equation (10) and that measured in a run of the algorithm, at generations 1, 3, and 8. The x axis represents the individual while the y axis represents the differences i.e. $\phi_{\text{Equation}} - \phi_{\text{run}}$. It is apparent how the predictions are exact within experimental errors.

4.2 Experiments

The behavior of a GA with linear selection and crossover was compared against a GA with headless chicken crossover, a GA with normal selection and crossover, and a GA using Holland's selection and crossover. The comparison was made on the ZeroMax and OneMix (Equation 26) problems.

We used the following settings: chromosome length $\ell = 200$, population size 500, 10 independents runs and the crossover probability was varied from 10% to 100% in steps of 10% (i.e., $100\%, 90\% \ldots 10\%$).

Figures (2), (3), (4), and (5) show the results for the ZeroMax problem for the normal selection, Holland's selection, linear selection, and headless chicken crossover, respectively. Only the probabilities $p_{xo} = 10\%$, $p_{xo} = 5\%$, and $p_{xo} = 100\%$ were plotted with the purpose of making the figures more understandable.

Figure 2 shows the results for the linear selection. It is observed that probabilities $p_{xo} = 50\%$ and $p_{xo} = 100\%$ exhibit a similar performance and it is the best performance within the four selection schemes. Figure 3 presents the results for Holland's selection. In this case the best performance is obtained using $p_{xo} = 50\%$. The results for the linear selection can be seen in Figure 4. It is observed that the best behavior is obtained with $p_{xo} = 50\%$ and $p_{xo} = 10\%$. Finally, Figure 5 presents the performance of the headless chicken crossover. It is noted that GAs with probabilities $p_{xo} = 100\%$ and $p_{xo} = 50\%$ are making no progress in the search because of the high rate of crossover which effectively can be interpreted as high rate of mutation.

Comparing figures (2), (3), (4), and (5) it is observed that the best performance is obtained in the normal selection and the worst is the headless chicken crossover. It can be seen that linear selection with $p_{xo} = 100\%$ and Holland selection with $p_{xo} = 100\%$ have a similar behavior. Furthermore, linear selection with $p_{xo} = 50\%$ and $p_{xo} = 10\%$ and Holland selection with $p_{xo} = 10\%$ also exhibit a similar performance.

The results obtained in this experiment confirm what we have expected based on the theory presented in the previous section.

For the OneMix problem the crossover probability was varied from 100% to 20% by a step of 10% and from 20% to 10% by a step of 2% (i.e. $20\%, 18\% \ldots 10\%$). This was

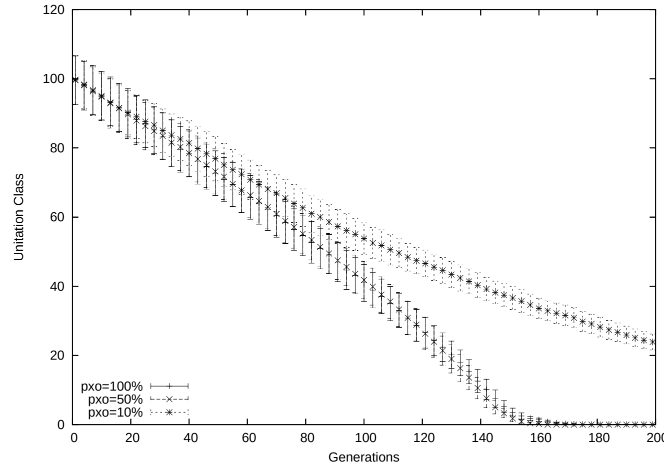


Figure 2: ZeroMax Problem Normal Selection.

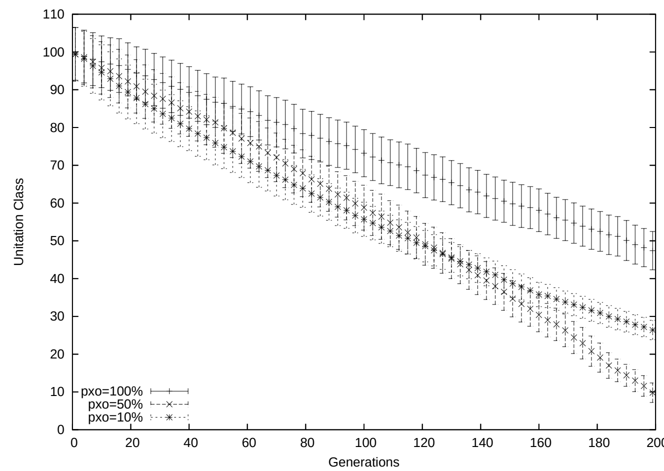


Figure 3: ZeroMax Problem Holland Selection.

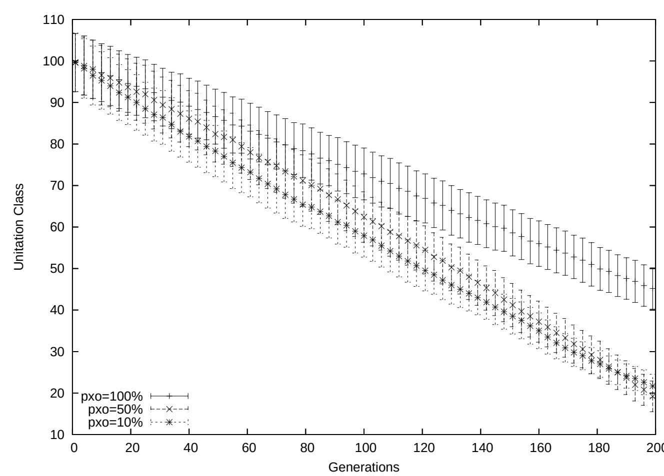


Figure 4: ZeroMax Problem Linear Selection.

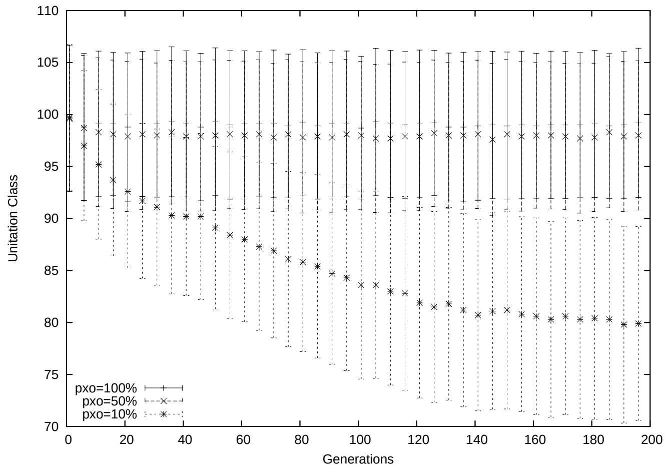


Figure 5: ZeroMax Problem Headless Chicken Crossover.

done to find the value of p_{xo} where there is the shift in convergence from the global optimum to the local optimum. The results are shown in Figures (6), (8), and (9). The plots present only the probability $p_{xo} = 90\%$, $p_{xo} = 50\%$, $p_{xo} = 30\%$, $p_{xo} = 18\%$, $p_{xo} = 16\%$, $p_{xo} = 14\%$, $p_{xo} = 12\%$, and $p_{xo} = 10\%$ in order to make them more understandable.

Figure 6 shows the results for the normal selection. With $p_{xo} = 10\%$ the algorithm converges towards the optimal unitation class. However, crossover rates $p_{xo} = 18\%$, $p_{xo} = 16\%$, and $p_{xo} = 14\%$ make the algorithm stay around unitation class 100 (the one of the initial population) showing almost no progress in the search. Rates $p_{xo} > 18\%$ have a tendency of driving the algorithm towards unitation class 200 while probabilities $p_{xo} < 14\%$ drive it towards unitation class 0. The runs with $p_{xo} = 90\%$ and $p_{xo} = 10\%$ present the highest tendencies towards $u = 200$ and $u = 0$, respectively.

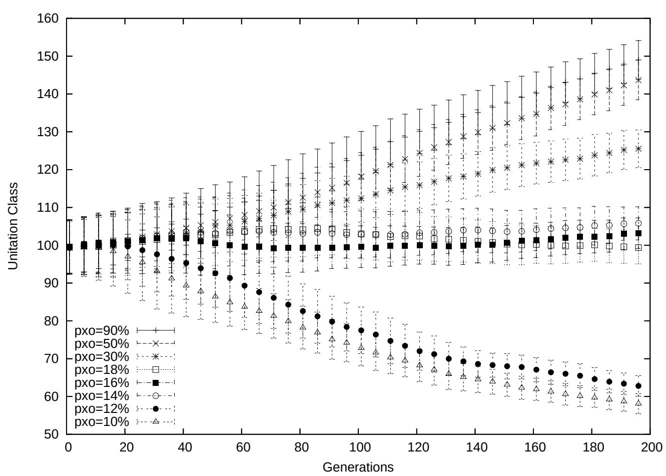


Figure 6: OneMix Problem Normal Selection.

The results for the Holland's selection are presented in Figure 7. It is observed that probabilities $p_{xo} = 10\%$, $p_{xo} = 12\%$, $p_{xo} = 14\%$, and $p_{18}\%$ are driving the algorithm to look below the unitation class 100 and probabilities $p_{xo} = 14\%$ and $p_{xo} = 12\%$ have a tendency of driving the algorithm towards the 0 unitation class. All other probabilities make the algorithm look above the 100 unitation class which is a local optimum.

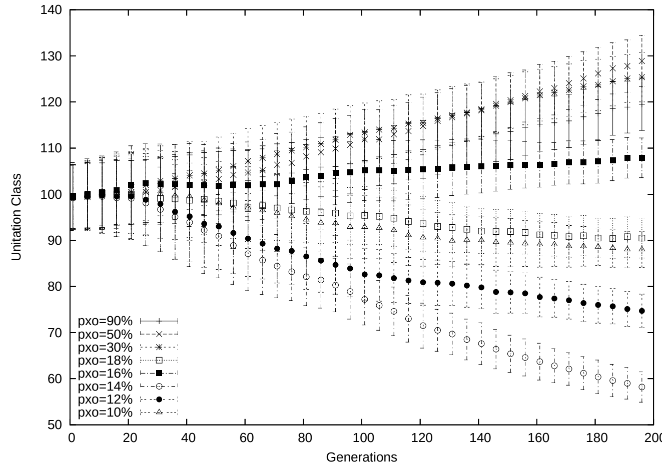


Figure 7: OneMix Problem Holland Selection.

Figure 8 shows the results for the linear selection. Probabilities $p_{xo} = 50\%$ and $p_{xo} = 12\%$ are the ones that show the highest drive towards unitation classes 0 and 200, respectively. All runs with $p_{xo} > 16\%$ have a tenendency towards the 200 unitation class, while for $p_{xo} < 16\%$ tends to the 100 unitation class.

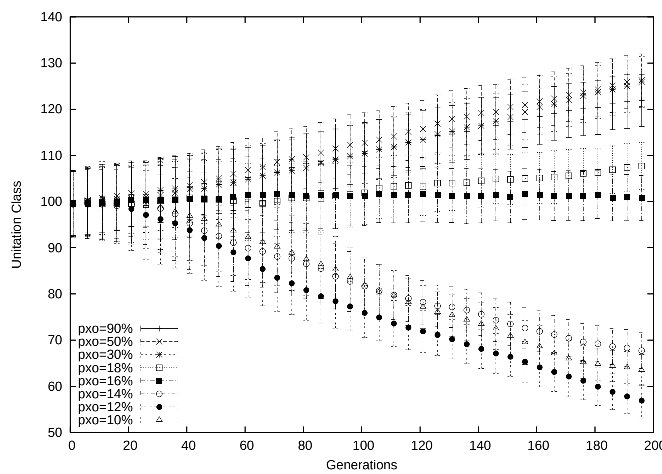


Figure 8: OneMix Problem Linear Selection.

Figure 9 shows the results for the Headless Chicken Crossover. Here, there is no crossover rate that drives the population towards the 200 unitation class, but rates $p_{xo} \geq 30\%$ always keep the population around the 100 unitation class, showing no improvement in the search.

Comparing Figures (6), (7), (8), and (9) it is observed that the best performance is obtained by the linear selection with $p_{xo} = 12\%$ and that the worst behavior is found in the normal selection with $p_{xo} = 90\%$. It is also noted that the headless chicken runs were never attracted to the local optimum which is consistent with the hypothesis that OneMax is deceptive for crossover.

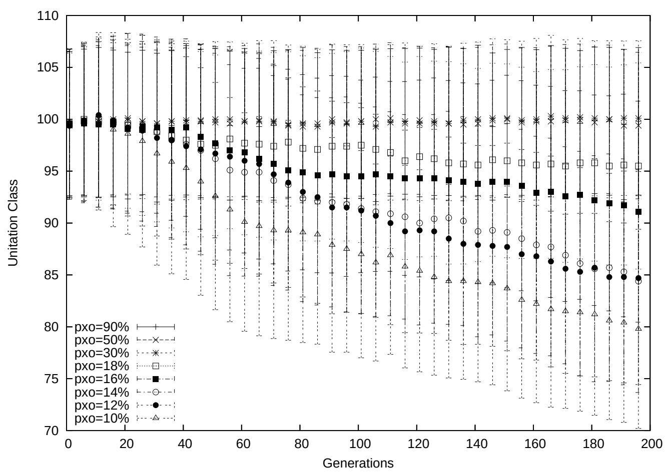


Figure 9: OneMix Problem Headless Chicken Crossover.

Also It is observed that more settings of the headless chicken crossover were attracted towards the 0 unitation class. While normal selection presented the fewer settings towards the 0 unitation class and instead it was more attracted to the 200 unitation class.

5. CONCLUSIONS

We have considered forms of selection where parents are not selected independently. The studied theoretically three different forms of selection — pure linear selection, semi-linear selection and Holland's selection — in conjunction with crossover and found, surprisingly, that two such forms actually correspond to a preexisting form of selection (originally defined by Holland) with crossover and standard selection with headless-chicken crossover. One form, semi-linear selection, where the parents are jointly selected with a probability proportional to the average of their selection probabilities, however, provided, in conjunction with crossover, novel features that are somehow in between those of a crossover-based and a mutation-based GA with normal selection.

Experimental results have shown that semi-linear selection behaves in the middle between algorithms driven by crossover or those driven by mutation. This results confirmed the findings of the theory.

6. REFERENCES

[1] P. J. Angeline. Subtree crossover: Building block engine or macromutation? In J. R. Koza, K. Deb, M. Dorigo, D. B. Fogel, M. Garzon, H. Iba, and R. L. Riolo, editors, *Genetic Programming 1997: Proceedings of the Second Annual Conference*, pages 9–17, Stanford University, CA, USA, 13-16 July 1997. Morgan Kaufmann.

[2] T. Blickle and L. Thiele. A mathematical analysis of tournament selection. In L. J. Eshelman, editor, *Proceedings of the Sixth International Conference on Genetic Algorithms (ICGA'95)*, pages 9–16, San Francisco, California, 1995. Morgan Kaufmann Publishers.

[3] T. Blickle and L. Thiele. A comparison of selection schemes used in evolutionary algorithms. *Evolutionary Computation*, 4(4):361–394, 1997.

[4] D. E. Goldberg and K. Deb. A comparative analysis of selection schemes used in genetic algorithms. In G. J. E. Rawlins, editor, *Foundations of Genetic Algorithms*. Morgan Kaufmann Publishers, 1991.

[5] J. H. Holland. *Adpatation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1975.

[6] T. Jones. Crossover, macromutation, and population-based search. In L. Eshelman, editor, *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 73–80, San Francisco, CA, 1995. Morgan Kaufmann.

[7] T. Motoki. Calculating the expected loss of diversity of selection schemes. *Evolutionary Computation*, 10(4):397–422, 2002.

[8] R. Poli. Tournament selection, iterated coupon-collection problem, and backward-chaining evolutionary algorithms. In *Proceedings of the Foundations of Genetic Algorithms Workshop (FOGA 8)*, 4th January 2005.

[9] R. Poli and W. B. Langdon. Backward-chaining evolutionary algorithms. *Artificial Intelligence*, 170(11):953–982, Aug. 2006.

[10] R. Poli and N. F. McPhee. Exact GP schema theory for headless chicken crossover and subtree mutation. In *Proceedings of the 2001 Congress on Evolutionary Computation CEC2001*, pages 1062–1069, COEX, World Trade Center, 159 Samseong-dong, Gangnam-gu, Seoul, Korea, 27-30 May 2001. IEEE Press.

[11] R. Poli, A. H. Wright, N. F. McPhee, and W. B. Langdon. Emergent behaviour, population-based search and low-pass filtering. In *2006 IEEE World Congress on Computational Intelligence, 2006 IEEE Congress on Evolutionary Computation*, pages 395–402, Vancouver, 16-21 July 2006.

[12] K. Sastry and D. E. Goldberg. Modeling tournament selection with replacement using apparent added noise. In *Proceedings of ANNIE 2001*, volume 11, pages 129–134, 2001.

[13] A. Sokolov and D. Whitley. Unbiased tournament selection. In H.-G. Beyer, U.-M. O'Reilly, D. V. Arnold, W. Banzhaf, C. Blum, E. W. Bonabeau, E. Cantu-Paz, D. Dasgupta, K. Deb, J. A. Foster, E. D. de Jong, H. Lipson, X. Llora, S. Mancoridis, M. Pelikan, G. R. Raidl, T. Soule, A. M. Tyrrell, J.-P. Watson, and E. Zitzler, editors, *GECCO 2005: Proceedings of the 2005 conference on Genetic and evolutionary computation*, volume 2, pages 1131–1138, Washington DC, USA, 25-29 June 2005. ACM Press.

[14] C. R. Stephens and R. Poli. EC theory – "in theory": Towards a unification of evolutionary computation theory. In A. Menon, editor, *Frontiers of Evolutionary Computation*, pages 129–156. Kluwer, Boston, MA, 2004.