Static and Dynamic Selection Thresholds Governing the Accumulation of Information in Genetic Algorithms Using Ranked Populations

John Milton miltonj77@bigpond.com
Faculty of Engineering and Information Technology, University of Technology,
Sydney, Broadway, New South Wales 2007, Australia

Paul J. Kennedy paulk@it.uts.edu.au
Faculty of Engineering and Information Technology, University of Technology,
Sydney, Broadway, New South Wales 2007, Australia

Abstract

Mutation applied indiscriminately across a population has, on average, a detrimental effect on the accumulation of solution alleles within the population and is usually beneficial only when targeted at individuals with few solution alleles. Many common selection techniques can delete individuals with more solution alleles than are easily recovered by mutation. The paper identifies static and dynamic selection thresholds governing accumulation of information in a genetic algorithm (GA). When individuals are ranked by fitness, there exists a dynamic threshold defined by the solution density of surviving individuals and a lower static threshold defined by the solution density of the information source used for mutation. Replacing individuals ranked below the static threshold with randomly generated individuals avoids the need for mutation while maintaining diversity in the population with a consequent improvement in population fitness. By replacing individuals ranked between the thresholds with randomly selected individuals from above the dynamic threshold, population fitness improves dramatically. We model the dynamic behavior of GAs using these thresholds and demonstrate their effectiveness by simulation and benchmark problems.

Keywords

Genetic algorithms, information theory, solution density, ranked populations, selection thresholds, selection pressure, ranking.

1 Introduction

Many researchers have shown that mutation can be detrimental to the success of a GA as it may alter optimal solution alleles present in the population to other, nonoptimal alleles (e.g., Milton et al., 2005; Galvan-Lopez and Poli, 2006; Wright and Richter, 2006; Ochoa, 2006). Poor selection choices can similarly lead a GA to delete individuals containing optimal solution alleles. These lost alleles can only be reintroduced by mutation or the addition of newly generated individuals that replace deleted individuals (Gonalves et al., 2005). This paper identifies selection thresholds in ranked populations that separate individuals with a high density of optimal solution alleles from those with a low density. We use some basic ideas from information theory to characterize the "density of optimal solution alleles" as "solution density." The thresholds differentiate

Downloaded from http://direct.mit.edu/evco/article-pdf/18/2/229/1494001/evco.2010.18.2.18203.pdf by guest on 17 June 2021

© 2010 by the Massachusetts Institute of Technology

those individuals with a lower solution density than the information source as sought by Milton et al. (2005). These individuals can be targeted for replacement instead of mutation to accelerate the accumulation of information by the GA.

The existence of both static (k_0) and dynamic (k_g) selection thresholds with $k_0 \leq k_g$ in a population of individuals is described. These thresholds govern the accumulation of information in a GA. We show that by replacing individuals ranked below the static threshold with randomly generated replacements the GA performs better than when mutation is used to maintain diversity. By replacing individuals ranked between the static and dynamic selection thresholds with randomly selected individuals from above the dynamic threshold, selection pressure may be controlled to achieve a balance between the rate of improvement and population diversity. If selection is implemented by ordering individuals in the population by fitness, the thresholds are easily visualized, but their existence is not dependent on this approach to selection.

The rest of this paper is organized as follows. Section 2 provides some general background to GAs, information theory, and related work. The idea of information in a GA population and the associated idea of solution density are then defined in Section 3. Section 4 develops a model of solution density in a GA subject to a static selection threshold and randomly generated replacements in lieu of mutation. Simulations to validate this model are presented. Section 5 expands the model and supporting simulation to remove some constraining assumptions so that a dynamic selection threshold and child replacements are accommodated by a more realistic algorithm. The knowledge gained from the simulations is then applied to a bit–trap benchmark as proposed by Harik (1999) with excellent results. Sections 7 and 8 provide suggestions for further research and concluding remarks.

2 Background

Genetic algorithms are optimization algorithms based on the principles of biological evolution. They are relatively straightforward to program but understanding how they work is challenging and has been a major research goal for some decades. Early work on evolutionary algorithms occurred in the late 1950s and early 1960s. Bremermann et al. (1966), Fraser (1957), and Box (1957) described artificial evolution systems. From the 1970s, work on evolutionary algorithms accelerated with the increasing availability of computers and diversified into a variety of branches including genetic algorithms (Rechenberg, 1973; Holland, 1975; Goldberg, 1989; Jong and Spears, 1991; Reeves, 1993; Whitley et al., 1995) which represents problems as strings of symbols; genetic programming (Koza, 1997; Poli, 2001), which evolves computer programs rather than strings of symbols; and real valued genetic algorithms (Rechenberg, 1973; Schwefel, 1981).

More recently, a significant volume of research (Rowe, 2001; Rowe et al., 2002, 2004, 2007; Poli et al., 2004; Toussaint, 2004; Mitavskiy, 2004; Borenstein and Poli, 2006) has examined genetic algorithm operators in detail through Markov chain analysis, groups, and other mathematical tools. Such analysis provides significant insight into the structure of search spaces as well as the nature of genetic operators such as crossover, mutation, and selection.

Even with this work, a rigorous approach to optimal GA design, akin to electronic circuit design or mechanical engineering designs has not been achieved (Jansen et al., 2005). A step toward this goal is the little models approach in Goldberg (2002) whereby GA behavior is simulated subject to constraints that simplify the model. Milton et al.

Downloaded from http://direct.mit.edu/evco/article-pdf/18/2/229/1494001/evco.2010.18.2.18203.pdf by guest on 17 June 2021

(2005) use such a constrained model and apply an information theoretic approach to show that mutation applied indiscriminately across the population has, on average, a detrimental effect on the accumulation of solution alleles within the population and that mutation is only beneficial when targeted at individuals with a lower solution density than the mutation source.

Selection is a genetic operator that retains part of the population for use as parents for generating a new population. Many implementations of selection have been proposed, including direct selection, proportional selection, uniform ranking (Schwefel, 1995), linear ranking (Baker, 1985), tournament selection (Blickle and Thiele, 1995), and Genitor (Whitley, 1989). With the exception of direct selection, each of these implementations is stochastic, using a probability of selection derived from the performance of individuals, rather than an absolute threshold performance. While this approach is biologically plausible, it leaves open the possibility of deleting individuals with more optimal solution alleles than can be easily recovered, for example using mutation. Gonalves et al. (2005) use arbitrary selection thresholds where the top 10% of the population are retained in the next generation and the bottom 20% are replaced by randomly generated individuals. Our work takes a similar approach, but we identify the selection thresholds by linking them to the algorithm used to generate the replacement individuals, rather than choosing them arbitrarily.

3 Solution Density

Information theory is characterized by a quantitative approach to the notion of information. It provides a framework for understanding how information can be transmitted and stored compactly and for calculating the maximum quantity of information that can be transmitted through a channel (Van der Lubbe, 1997). Information theory provides an interesting lens through which to view the mechanics underlying a GA. It gives us insight into the flow of alleles through a population and the differences in the structure of information between ranked and unranked populations.

Generally, the initial population of a GA is constructed using a memoryless information source, which randomly generates symbols and places them into positions (loci) of individuals ranging from position 1 to L. Information theory defines an information source as an algorithm that generates symbols in a stationary[1] stochastic sequence. A memoryless information source is one where the symbols are statistically independent (Van der Lubbe, 1997). Each individual generated this way represents a possible solution to the problem.

We define *ideal alleles* as those symbols in the appropriate loci that form an optimal solution. If more than one optimal solution to the problem exists, arbitrarily choose one of these solutions to represent the optimal solution. The concept of ideal alleles is used throughout this paper to simplify the explanation of ideas and observations. It is understood that ideal alleles cannot be easily identified in real problems.

We introduce the term *solution density* to refer to the frequency of ideal alleles in the population at a particular generation and denote it as ρ_g for generation g. Solution density is a measure of a population's fitness. Unless the information source has special knowledge of the problem that biases it to produce ideal alleles at a greater rate than other alleles, these ideal alleles will occur at the rate $1/A$ in the initial population where

[1] Stationary means that the probability of symbol generation does not change with time.

Downloaded from http://direct.mit.edu/evco/article-pdf/18/2/229/1494001/evco.2010.18.2.18203.pdf by guest on 17 June 2021

A is the number of possible alleles at each locus. We refer to A as the allele cardinality (binary, octal, hexadecimal, or other) and assume that it does not vary from locus to locus within genomes of the same problem. Thus, the solution density of the initial population can be given as

$$\rho_0 = \frac{1}{A}. \tag{1}$$

As with ideal alleles, the notion of solution density is used to explain ideas and observations. The solution density identified by a genetic algorithm operating on a real problem is the algorithm's best estimate of the solution in the presence of noise from various sources including the stochastic nature of the operators themselves, and false optima which deceive the operators into encoding misleading information into the evolving population.

The population has an entropy given by

$$H_g = -\sum_{l=1}^{L}\sum_{a=1}^{A} p_a(l,g)\log_2 p_a(l,g)$$

where $p_a(l,g)$ is the relative frequency of each allele a at locus l at generation g.

Weaver and Shannon (1949, p. 20) describe entropy as a measure of uncertainty and information received as the difference between the uncertainty at the receiver before the arrival of a signal and the uncertainty at the receiver after the arrival of a signal. In the context of a genetic algorithm, the population is the receiver while the signal is provided by the selection operator. Therefore the entropy of the population from generation to generation can be used to measure the accumulation of information by a population.

The action of selection increases the relative frequency of ideal alleles in the population. This reduces the uncertainty of the population so that

$$H_g > H_{g+1}.$$

Hence the information encoded into the population at generation g by the action of selection is given by

$$R_g = H_0 - H_g$$

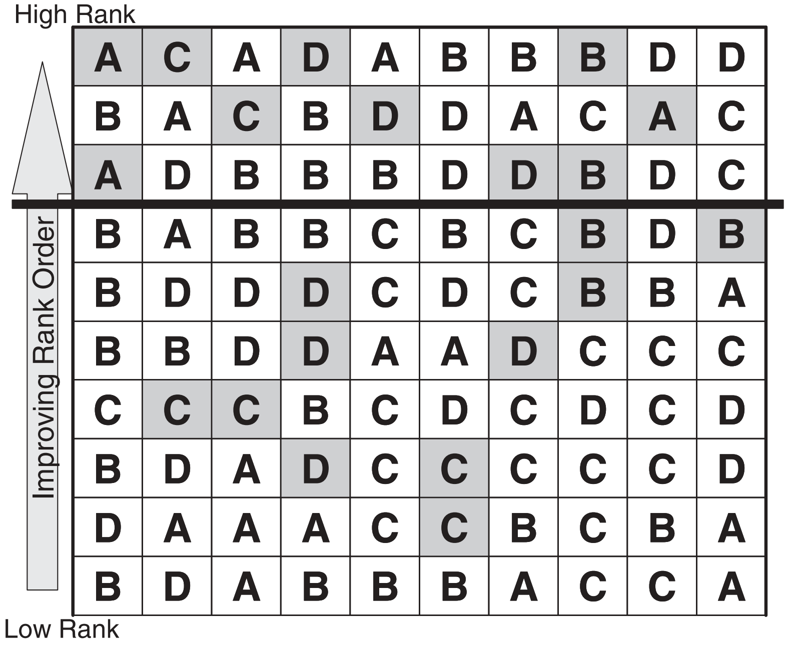
the difference between the uncertainty in the initial population and the uncertainty in the population at generation g.

This information is proportional to the solution density ρ_g since $max\{p_a(l,g)\}$ is the relative frequency of ideal alleles in locus l at generation g. Therefore, the solution density is related to the accumulated information by

$$\rho_g = \frac{1}{L}\sum_{l=1}^{L} max\{p_a(l,g)\}.$$

This means that the solution density (ρ_g) can be used to model information accumulation by a genetic algorithm. We prefer the use of solution density over entropy for our model as solution density is quicker to calculate and can be directly applied to the binomial model introduced in Section 4.

Downloaded from http://direct.mit.edu/evco/article-pdf/18/2/229/1494001/evco.2010.18.2.18203.pdf by guest on 17 June 2021



A	C	A	D	A	B	B	B	D	D
B	A	C	B	D	D	A	C	A	C
A	D	B	B	B	D	D	B	D	C
B	A	B	B	C	B	C	B	D	B
B	D	D	D	C	D	C	B	B	A
B	B	D	D	A	A	D	C	C	C
C	C	C	B	C	D	C	D	C	D
B	D	A	D	C	C	C	C	C	D
D	A	A	A	C	C	B	C	B	A
B	D	A	B	B	B	A	C	C	A

Figure 1: A population of 10 individuals (rows) having ten loci (columns) ranked by ideal allele from the solution (A,C,C,D,D,C,D,B,A,B) indicated in gray. The threshold separates individuals with more/less ideal alleles than the population average, as generated by the memoryless information source.

4 A Model of Solution Density in a GA Subject to Selection

When individuals, each with L loci, are ranked by the number of ideal alleles they contain, then a gradient from low to high fitness exists in the population. In this gradient a static threshold ($k_0 : 0 \leq k_0 \leq L$) exists, where individuals with more than k_0 ideal alleles have a solution density greater than that of the information source (Figure 1). Deleting any individual from above this static threshold represents lost information that cannot be easily recovered using the information source. Most of the selection implementations outlined in Section 2 run this risk. Similarly, applying mutation to any individual above this threshold will, on average, decrease the solution density of the population rather than increase it (Milton et al., 2005).

Ideal alleles λ will initially be distributed throughout the population with binomial probability distribution

$$p(\lambda|L, \rho_0) = \frac{L!}{\lambda!(L-\lambda)!}\rho_0^{\lambda}(1-\rho_0)^{(L\text{-}\lambda)}$$

that is similar to that shown in Figure 2. The binomial distribution describes the number of ideal alleles per individual $\{\lambda | 0 \leq \lambda \leq L\}$ where L is the number of loci per individual. The solution density ρ_g, of the population at generation g, is the ratio of the number of ideal alleles to the total number of alleles in the population.

Downloaded from http://direct.mit.edu/evco/article-pdf/18/2/229/1494001/evco.2010.18.2.18203.pdf by guest on 17 June 2021

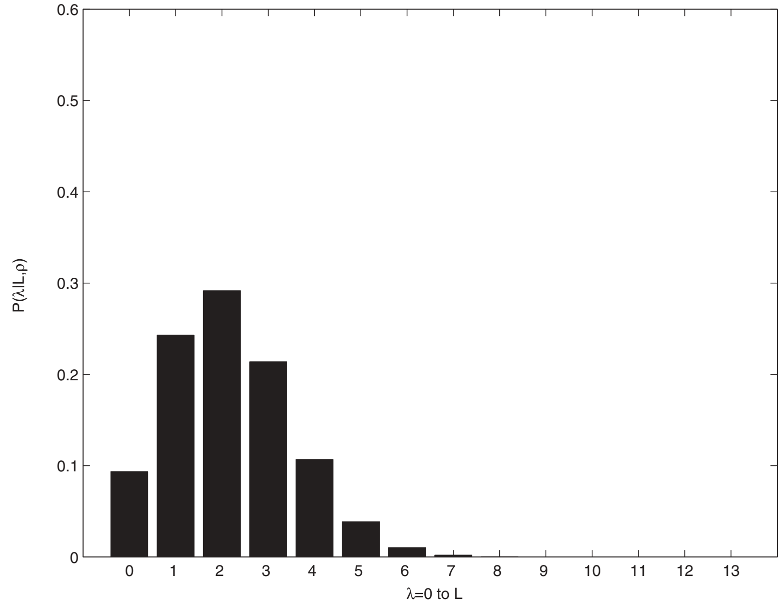


Figure 2: The binomial distribution $\mathbf{p}_0^L(0,\lambda)$ describes the number of ideal alleles per individual at generation 0.

Throughout this paper, binomial distributions are frequently used. To focus readers on the salient variables, the binomial distribution: $p(\lambda|L,\rho_g)$ shall be henceforth abbreviated as $\mathbf{p}(g,\lambda)$. Sometimes only parts of a binomial distribution are required. For example, $p(\lambda|L,\rho_g)$ for $\lambda = a$ to b. These partial distributions shall be abbreviated as $\mathbf{p}_a^b(g,\lambda)$. In each case a and b are in the range 0 to L.

Selection from a threshold k truncates the binomial distribution thus

$$\mathbf{p}_0^L(0,\lambda) \overset{\textbf{select}}{\Rightarrow} \mathbf{p}_{k+1}^L(0,\lambda).$$

An example of this kind of distribution is shown in Figure 3.

The population described by Figure 3 no longer has ideal alleles distributed binomially throughout the population. Instead, ideal alleles occur more frequently per individual in the surviving population than they did in the initial population. This would invalidate the continued use of binomial distribution equations to model the GA behavior. However, if crossover is now repeatedly applied to all of the individuals in this population, the distribution of ideal alleles across the population will return to a binomial distribution.

The minimum amount of crossover that achieves this return to a binomial distribution is referred to as *sufficient* crossover in this paper. Applying more crossover than this has no further effect on the distribution of ideal alleles in the population and is computationally intensive. Therefore the accurate identification of sufficient crossover is important to the efficient operation of the GA. Sections 4.2 and 5.2 compare the

Downloaded from http://direct.mit.edu/evco/article-pdf/18/2/229/1494001/evco.2010.18.2.18203.pdf by guest on 17 June 2021

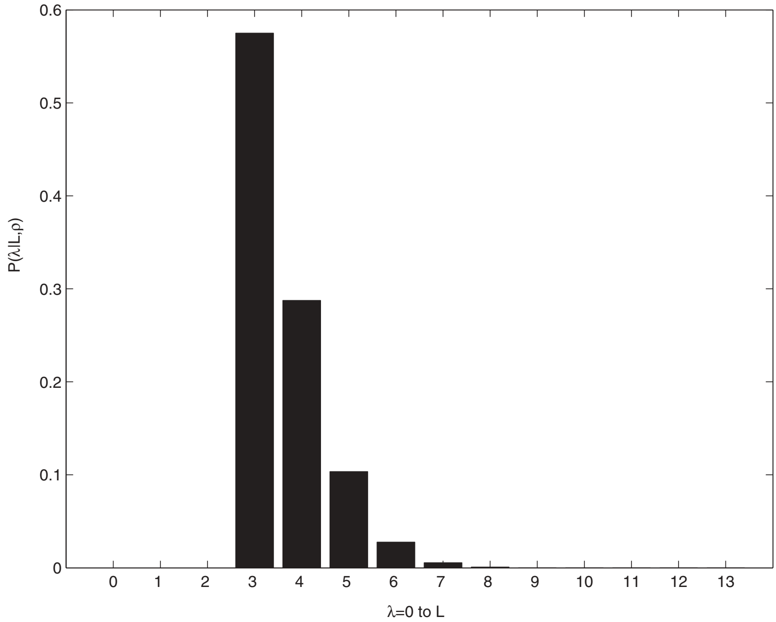


Figure 3: The truncated probability distribution $\mathbf{p}_3^{13}(0, \lambda)$ of the population that has had individuals with two or fewer ideal alleles deleted (selection threshold $k = 2$).

effect of insufficient to sufficient crossover on the fidelity of the model described and Appendix A provides the calculation for how much crossover is sufficient.[2]

Once sufficient crossover has been applied between all of the individuals in the population, the ideal alleles are again distributed binomially across the population, and the binomial distribution equations can continue to be used to model the growth of solution density ρ_g from generation to generation.

$$\mathbf{p}_0^L(0, \lambda) \overset{\textbf{select}}{\Rightarrow} \mathbf{p}_{k+1}^L(0, \lambda) \overset{\textbf{crossover}}{\Rightarrow} \mathbf{p}_0^L(1, \lambda).$$

Figure 4 illustrates the binomial distribution $\mathbf{p}_0^{13}(1, \lambda)$ of a population that has the same solution density as the population in Figure 3. Note that the peak of the distribution has moved to the right when compared to Figure 2, indicating that a greater percentage of the population contains more ideal alleles. Hence, the solution density of the population has risen. Figure 5 illustrates the trend resulting from repeating

[2] An alternative approach might be to use gene pool recombination. In gene pool recombination, for each locus the two alleles to be recombined are chosen independently from the gene pool defined by the selected parent population. The biologically inspired idea of restricting the recombination to the alleles of two parents for each offspring is abandoned (Muhlenbein and Voigt, 1995). This approach decorrelates loci and results in a binomial distribution of ideal alleles. However, it may also lose alleles as some may not be chosen. This adds another operator that probabilistically leaks ideal alleles from the genetic algorithm as does selection and mutation. In addition, gene pool recombination scales with increasing individual length (L), while crossover described in this paper does not. Hence for populations of significant genome length (L), crossover is more efficient than gene pool recombination at redistributing alleles through a population.

Downloaded from http://direct.mit.edu/evco/article-pdf/18/2/229/1494001/evco.2010.18.2.18203.pdf by guest on 17 June 2021

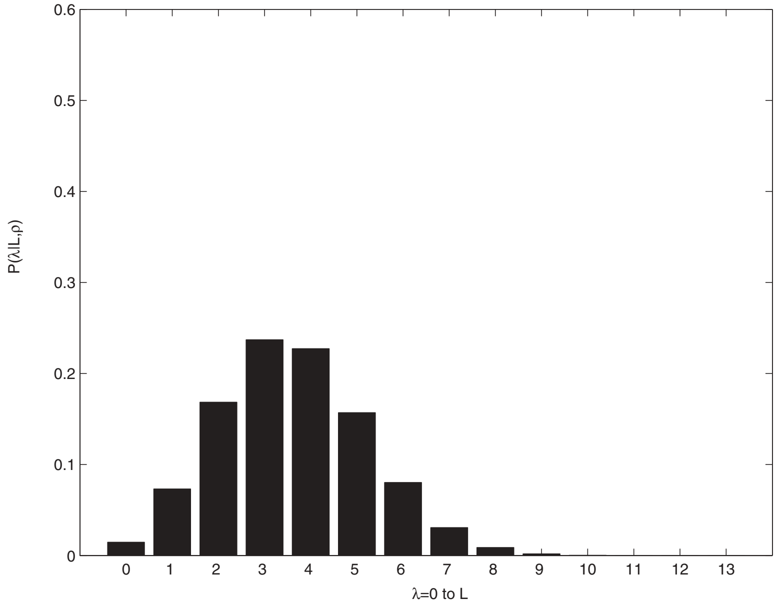


Figure 4: The truncated distribution of Figure 3 with alleles redistributed by sufficient crossover between all of the individuals in the population to return it to a binomial distribution.

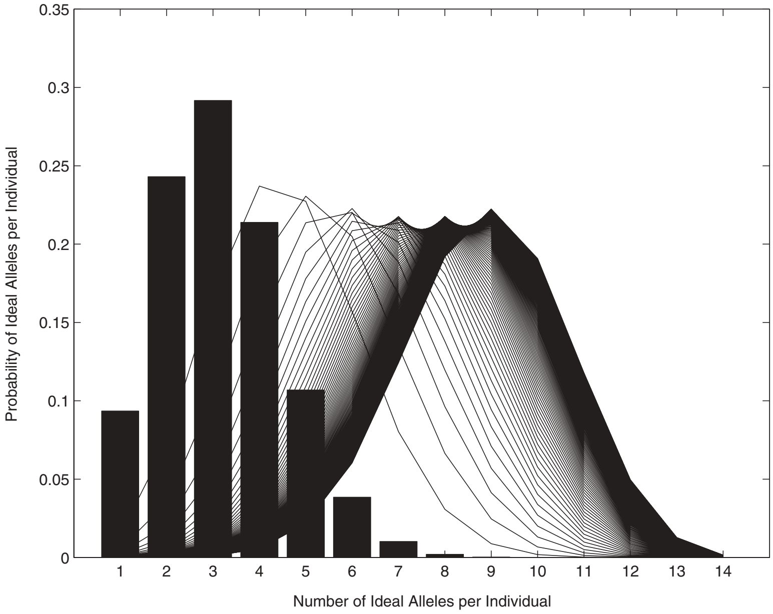


Figure 5: The movement of the probability density function from $\mathbf{p}(0, \lambda)$ (columns) to the region of higher solution density at $\mathbf{p}(g, \lambda)$ (lines) due to repeated selection and crossover.

Downloaded from http://direct.mit.edu/evco/article-pdf/18/2/229/1494001/evco.2010.18.2.18203.pdf by guest on 17 June 2021

this process each generation. This effect can be described algebraically to reveal the change in solution density and to identify the selection threshold k that optimizes the improvement in solution density for each generation.

The binomial distribution can also be used to model the change of solution density ρ_g from generation to generation when a univariate estimation distribution algorithm (UEDA; Muhlenbein and Paaß, 1996) is used to model successive generations. UEDAs use an estimate of a population's allele distribution, rather than an instance of a population, and modify this estimated distribution. This approach is popular as it does not require memory to store actual populations. An estimated distribution implicitly assumes that alleles are distributed throughout the population as represented by the distribution. Hence the distribution of ideal versus nonideal alleles must be binomially distributed in a UEDA and therefore the models described in this paper are applicable to UEDAs.

$$\mathbf{p}_0^L(0, \lambda) \overset{\textbf{UEDA}}{\Rightarrow} \mathbf{p}_0^L(1, \lambda)$$

Returning to the model, an expression for the expected solution density in the population at generation $g + 1$ which describes this change will now be constructed. First we need an expression describing the number of ideal alleles present in the population after selection. This expression must be further developed to include the number of ideal alleles that are added by the randomly generated replacement individuals. Finally, the expression for expected solution density must account for the probability that at least one individual survives to the next generation.

When individuals ranked above a threshold k are selected, the expected number of ideal alleles in the population at generation g is $N_g \sum_{\lambda=k+1}^{L} \lambda \mathbf{p}(g, \lambda)$ and the total number of alleles in the population at generation $g + 1$ is LN_{g+1}. Since the solution density is given by the ratio of ideal alleles in the population to the total alleles, the expected solution density at generation $g + 1$ for a population subject to selection only is

$$\mathbb{E}_s[\rho_{g+1}] = \frac{N_g \sum_{\lambda=k+1}^{L} \lambda \mathbf{p}(g, \lambda)}{LN_{g+1}}. \tag{2}$$

To this point we have not replaced the individuals deleted from the population. If randomly generated new individuals are now used to replace the deleted individuals and increase the diversity of symbols represented in the population (i.e., in lieu of mutation) then the solution density is further altered as follows. First, the number of individuals to be added to the population must be quantified. Next, the solution density associated with these individuals must be quantified. Then, this solution density must be added to the surviving population's solution density.

As the individuals with k or fewer ideal alleles (i.e., $\lambda \leq k$) were deleted, the number of individuals deleted and therefore the number of replacements required to maintain the population size so that $N_g = N_{g+1}$, is $N_g \sum_{\lambda=0}^{k} \mathbf{p}(g, \lambda)$. The solution density associated with these new individuals is $\frac{\sum_{\lambda=0}^{L} \lambda \mathbf{p}(0,\lambda)}{\sum_{\lambda=0}^{L} \mathbf{p}(0,\lambda)}$ which simplifies to $\sum_{\lambda=0}^{L} \lambda \mathbf{p}(0, \lambda)$ since the denominator $\sum_{\lambda=0}^{L} \mathbf{p}(0, \lambda) = 1$. Notice that as we are generating new individuals in the same way as for the initial population, we use the binomial distribution $\mathbf{p}(0, \lambda)$ rather than $\mathbf{p}(g, \lambda)$.

The solution density to be added to the surviving population is the product $N_g \sum_{\lambda=0}^{k} \mathbf{p}(g, \lambda) \sum_{\lambda=0}^{L} \lambda \mathbf{p}(0, \lambda)$, the number of randomly generated replacement

Downloaded from http://direct.mit.edu/evco/article-pdf/18/2/229/1494001/evco.2010.18.2.18203.pdf by guest on 17 June 2021

individuals multiplied by the number of ideal alleles associated with the randomly generated replacement individuals. Adding this term to the numerator of Equation (2) and simplifying gives

$$\frac{1}{L}\left[\sum_{\lambda=k+1}^{L}\lambda\mathbf{p}(g,\lambda)+\sum_{\lambda=0}^{k}\mathbf{p}(g,\lambda)\sum_{\lambda=0}^{L}\lambda\mathbf{p}(0,\lambda)\right].$$

However, the expected solution density in generation $g+1$ is valid only if at least one individual survives. Thus our final estimate of the expected solution density is

$$\mathbb{E}_{sr}[\rho_g]=\frac{1}{L}\left[\sum_{\lambda=k+1}^{L}\lambda\mathbf{p}(g,\lambda)+\sum_{\lambda=0}^{k}\mathbf{p}(g,\lambda)\sum_{\lambda=0}^{L}\lambda\mathbf{p}(0,\lambda)\right]\times\left[1-\left(\sum_{\lambda=0}^{k}\mathbf{p}(g,\lambda)\right)^{N_g}\right]. \quad (3)$$

Equation (3) describes the expected behavior in the change of solution density ρ_g of a population from generation to generation under the successive application of selection, random replacement, and crossover. This line of reasoning is critically dependent on the number of crossover operations. Insufficient crossover reduces the mixing of the ideal alleles and invalidates this analysis because the binomial distribution no longer applies. However, with sufficient crossover it is possible to model increasing solution density and estimate the number of individuals that exist at or below the threshold for any generation.[3]

4.1 Static Threshold

In order to use Equation (3) to model information flow in a GA, we need to determine a suitable value for the selection threshold k which will ensure a rise of solution density from generation g to $g+1$. For the solution density to increase, the ideal alleles lost when individuals are deleted must be less than the ideal alleles introduced by the randomly generated individuals replacing them. Hence if k is set to satisfy

$$\sum_{\lambda=0}^{k}\lambda\mathbf{p}(g,\lambda)<\sum_{\lambda=0}^{k}\mathbf{p}(g,\lambda)\sum_{\lambda=0}^{L}\lambda\mathbf{p}(0,\lambda) \quad (4)$$

then the solution density will rise and information will accumulate. The last factor on the right-hand side of Equation (4) is a constant. Assigning $K=\sum_{\lambda=0}^{L}\lambda\mathbf{p}(0,\lambda)$ and expanding both sides results in

$$0\mathbf{p}(g,0)+1\mathbf{p}(g,1)+\cdots+k\mathbf{p}(g,k)<K\mathbf{p}(g,0)+K\mathbf{p}(g,1)+\cdots+K\mathbf{p}(g,k).$$

Subtracting like terms on the left-hand side from like terms on the right-hand side gives

$$0<(K-0)\mathbf{p}(g,0)+(K-1)\mathbf{p}(g,1)+\cdots+(K-k)\mathbf{p}(g,k)$$

[3] Refer to Appendix A for the calculation of how much crossover is *sufficient*.

Downloaded from http://direct.mit.edu/evco/article-pdf/18/2/229/1494001/evco.2010.18.2.18203.pdf by guest on 17 June 2021

which is true for all $k \leq K$ since when the last factor equals zero the remainder of the right-hand side is positive. It may also be true for some $k > K$, but the degree to which this is true varies for different distributions $\mathbf{p}(g, \lambda)$. Consequently, a conservative bound on selection threshold which guarantees that information will accumulate is

$$k \leq \sum_{\lambda=0}^{L} \lambda \mathbf{p}(0, \lambda).$$

We can better quantify this bound on k by realizing that the initial solution density is $\rho_0 = 1/A$. This means that $\sum_{\lambda=0}^{L} \lambda \mathbf{p}(0, \lambda) = L/A$.

Therefore, if the selection threshold k is less than L/A, ideal alleles will accumulate and if k is greater than L/A, ideal alleles are unlikely to accumulate. This bound is the static selection threshold $k_0 = L/A$ which defines the boundary between information gain and information loss when using randomly generated replacement individuals. Replacing individuals having k_0 or fewer ideal alleles with new, randomly generated individuals will, on average, provide an increase in solution density. To summarize, the static selection threshold is given by

$$k_0 \leq \sum_{\lambda=0}^{L} \lambda \mathbf{p}(0, \lambda) = \frac{L}{A}. \tag{5}$$

4.2 Exploring the Static Threshold

We present a simulation study to examine the fidelity of the above equations modeling information flow, especially the accurate identification of the selection threshold and the influence of crossover. To achieve this, the results of a simulated GA with excessive crossover and little crossover are compared to the theoretical results derived above. The GA simulation uses the well-known royal road problem (Mitchell et al., 1992). We say that it is simulated because the GA knows the correct solution to the problem, unlike the usual case when the optimal solution is, of course, unknown. Each ideal allele is defined as the symbol 1 and hence the ideal evolution of the GA can be monitored and compared to the theoretical model. The simulation selects individuals for survival based on the number of 1s they contain. This means that the GA has perfect knowledge of the ranked order of individuals based on the number of ideal alleles each has. This unrealistic constraint is removed in Section 6.

The specific form of GA used in the simulation is shown in Algorithm 1.

Algorithm 1 Outline of genetic algorithm with randomly generated replacements

```
Data: Population Size, Termination Criteria, Genome Length = $L$, Cardinality = $A$
Define information source having cardinality $A$;
Calculate $k_0$;
for $n = 1$ to Population Size do
    Generate Individual of length $L$ using information source;
```

Downloaded from http://direct.mit.edu/evco/article-pdf/18/2/229/1494001/evco.2010.18.2.18203.pdf by guest on 17 June 2021

```
repeat
    Score each individual in the population using the objective function;
    Delete individuals in population with $k_0$ or less ideal alleles;
    Replace deleted individuals using information source;
    /* this paper uses $C = 310$ and $C = 10$ for comparison
       purposes                                               */
    for $c = 1$ to $C$ do
        Randomly select sections of length $L/2$ in 2 randomly selected
        individuals and exchange these sections;
until termination criteria = true;
```

Crossover is performed between randomly selected individuals. Each crossed-over section is six loci in length, selected from a random starting position in each parent, with wrap-around at genome ends. No mutation is applied as diversity is introduced into the population by replacing deleted individuals with new randomly generated individuals.

The simulation experiments were repeated for two scenarios corresponding to amounts of crossover at either end of the spectrum: (i) $C = 310$ crossover operations per generation and (ii) $C = 10$ crossover operations per generation. This first amount of crossover is chosen arbitrarily based on 10 times the population size.

The results for each scenario were averaged over 100 trials of the simulation to produce meaningful results. Parameters used for both the theoretical model and simulated GAs are an initial population size of $N = 31$ individuals, individual length $L = 13$ loci and allele cardinality of $A = 6$ alleles. With these parameters, the static selection threshold, k_0, is calculated using Equation (5) to be 2.1667.

Equation (3) defines the theoretical model and is used to predict the change in expected solution density for a variety of selection thresholds k over a number of generations (Figure 6). In Figures 6 to 8, different selection thresholds k are indicated by lines. For example, $k = 2$ is the graph of $\mathbb{E}[\rho_g]$ with a selection threshold of two ideal alleles. Dashed lines indicate the solution density of the information source used for generation and replacement of individuals.

Figure 7 shows the solution density of the population for 100 generations with 310 crossover operations per generation and Figure 8 shows the solution density with 10 crossover operations per generation.

4.2.1 Discussion

Where the number of crossover operations is sufficient to return the distribution of ideal alleles to a binomial distribution (Figure 7), then the model (Figure 6) is an excellent estimation of the simulated GA behavior. Indeed for the simulations where selection pressure (k) is set at less than four ideal alleles per individual ($k < 4$), the maximum mean squared error between the model and 100 experimental trials is only 0.0023. When fewer crossover operations are done (Figure 8) the model is less accurate, but for the simulations where $k < 4$, the maximum mean squared error between the model and 100 experimental trials is still only 0.0038.

When selection pressure exceeds the calculated selection threshold $k_0 = 2.1667$, the equations predict a collapse in information. That is, a leveling off or rapid decline in the solution density of the population occurs. For example, compare the line marked $k = 2$ in

Downloaded from http://direct.mit.edu/evco/article-pdf/18/2/229/1494001/evco.2010.18.2.18203.pdf by guest on 17 June 2021

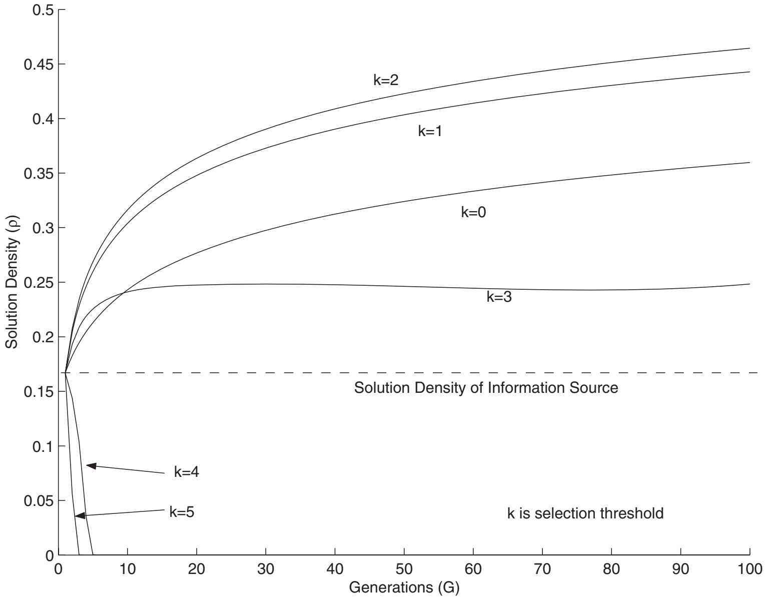


Figure 6: The expected solution density predicted by Equation (3) for a variety of selection thresholds k ($N = 31$, $L = 13$, $A = 6$).

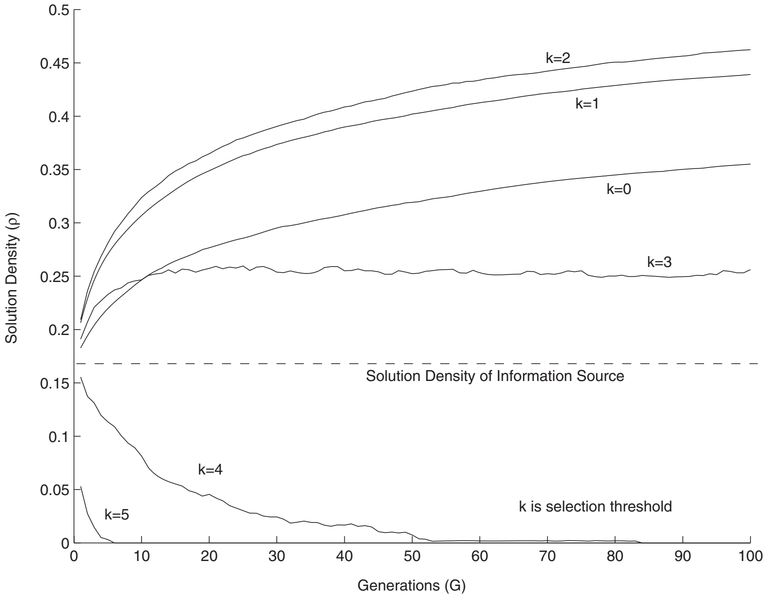


Figure 7: The average results of 100 trials where crossover has been performed between randomly selected pairs of individuals 310 times each generation, $C = 310$ ($N = 31$, $L = 13$, $A = 6$).

Downloaded from http://direct.mit.edu/evco/article-pdf/18/2/229/1494001/evco.2010.18.2.18203.pdf by guest on 17 June 2021

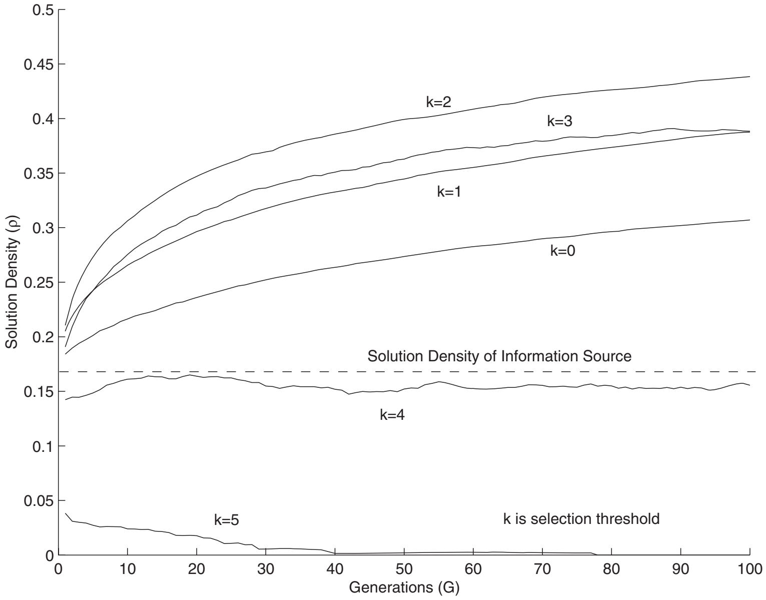


Figure 8: The average results of 100 trials where crossover has been performed between randomly selected pairs of individuals 310 times each generation, $C = 10$ ($N = 31$, $L = 13$, $A = 6$).

Figure 6 with $k = 3$ and $k = 4$ in Figure 6. This collapse is apparent in the simulations (see Figure 7 for $k = 2, k = 3$, and $k = 4$). As predicted, the highest possible selection pressure that does not exceed $k_0 = 2.1667$ provides the fastest improvement of solution density.

Low selection pressure in the simulation with sufficient crossover (i.e., Figure 7, $C = 310$) achieves slightly faster improvement in solution density than does low selection pressure in the simulation with insufficient crossover (Figure 8, $C = 10$). In other words, the simulations illustrated by Figure 7 $k = 0$, 1, and 2 are superior to the simulations illustrated by Figure 8 $k = 0$, 1, and 2.

High selection pressure (i.e., with $k = 3$, 4, or 5) in the simulation with sufficient crossover (Figure 7, $C = 310$) has a slower increase in solution density than does high selection pressure in the simulation with insufficient crossover (Figure 8, $C = 10$). This suggests that a little crossover is more robust to higher selection pressure than excessive crossover.

In both crossover scenarios, the maximum solution density reached is quite low. Indeed, it is less than 0.5, the starting point for an equivalent GA with a binary allele cardinality ($A = 2$).

One way to increase this maximum solution density lies with how individuals are replaced in the population. As we will show in the next section, rather than replacing individuals with random genomes, the maximum solution density may be increased if individuals are replaced with randomly selected survivors of the previous generation. That is, we use survivors as parents.

5 A Model with Parents

The model and simulations described in Section 4 are effective in replacing lost information but the overall improvement in solution density is quite low. The solution density

Downloaded from http://direct.mit.edu/evco/article-pdf/18/2/229/1494001/evco.2010.18.2.18203.pdf by guest on 17 June 2021

of 0.5 achieved after 100 generations only provides a very small probability that the optimal solution exists in the population. Nevertheless, Figure 7 shows that the average solution density of the population, while low, is still higher than that of the information source. Therefore, it seems sensible to use this population as a source of replacement individuals.

If we use individuals from the surviving population to replace deleted individuals, we are using them as parents for the following generation and the model equations require some revision. As before, the proportion of individuals deleted from the current population equals the proportion of replacement individuals required in the next population so that $N_g = N_{g+1}$. Hence, the proportion of replacement individuals in the next generation is given by $\sum_{\lambda=0}^{k} \mathbf{p}(g,\lambda)$. The ideal alleles that the randomly chosen parents add to the population is $\frac{\sum_{\lambda=k+1}^{L} \lambda \mathbf{p}(g,\lambda)}{\sum_{\lambda=k+1}^{L} \mathbf{p}(g,\lambda)}$. We can estimate the expected solution density in the next generation as

$$\mathbb{E}_{sp}[\rho_{g+1}] = \frac{1}{L}\left[\sum_{\lambda=k+1}^{L} \lambda \mathbf{p}(g,\lambda) + \frac{\sum_{\lambda=0}^{k} \mathbf{p}(g,\lambda) \sum_{\lambda=k+1}^{L} \lambda \mathbf{p}(g,\lambda)}{\sum_{\lambda=k+1}^{L} \mathbf{p}(g,\lambda)}\right]$$

which after some manipulation simplifies to

$$\mathbb{E}_{sp}[\rho_{g+1}] = \frac{1}{L}\left[\frac{\sum_{\lambda=k+1}^{L} \lambda \mathbf{p}(g,\lambda)}{\sum_{\lambda=k+1}^{L} \mathbf{p}(g,\lambda)}\right].$$

As the solution density of this information source (i.e., the surviving population) rises over successive generations, then the selection threshold for individuals replaced using the surviving population also increases. We denote this dynamic selection threshold k_g and define the expected solution density in generation $g+1$ as

$$\mathbb{E}_{sp}[\rho_{g+1}] = \frac{1}{L}\left[\frac{\sum_{\lambda=k_g+1}^{L} \lambda \mathbf{p}(g,\lambda)}{\sum_{\lambda=k_g+1}^{L} \mathbf{p}(g,\lambda)}\right]. \tag{6}$$

Equation (6) accounts for the use of randomly selected survivor parents, but it does not permit the introduction of new information to replace information lost during selection. This means that the diversity of the population may decrease, potentially resulting in premature convergence on a suboptimal solution or in the stalling of the algorithm as it runs out of useful information. In order to resolve this, we can replace individuals below the static threshold k_0 with randomly generated individuals and replace individuals between the static threshold k_0 and the dynamic threshold k_g with randomly selected survivor parents from above k_g. For the model to reflect this, we combine Equations (3) and (6) thus

$$\begin{aligned}\mathbb{E}_{srp}[\rho_{g+1}] = \frac{1}{L}\Bigg[&\sum_{\lambda=k_g+1}^{L} \lambda \mathbf{p}(g,\lambda) + \sum_{\lambda=0}^{k_0} \mathbf{p}(g,\lambda) \sum_{\lambda=0}^{L} \lambda \mathbf{p}(0,\lambda)\\ &+ \frac{\sum_{\lambda=k_0+1}^{k_g} \mathbf{p}(g,\lambda) \sum_{\lambda=k_g+1}^{L} \lambda \mathbf{p}(g,\lambda)}{\sum_{\lambda=k_g+1}^{L} \mathbf{p}(g,\lambda)}\Bigg] \times \left[1 - \left(\sum_{\lambda=0}^{k_g} \mathbf{p}(g,\lambda)\right)^{N_g}\right].\end{aligned} \tag{7}$$

In Equation (7), the first term in the first set of brackets represents the surviving solution density, the second term represents solution density introduced by randomly

Downloaded from http://direct.mit.edu/evco/article-pdf/18/2/229/1494001/evco.2010.18.2.18203.pdf by guest on 17 June 2021

generated individuals, and the third term represents the solution density introduced by parent individuals. This sum is multiplied by the probability that at least one individual survives.

5.1 Dynamic Threshold

Having described how the expected solution density changes from generation to generation, we need to quantify the threshold k_g that supports this accumulation of information. The threshold k_0 associated with randomly generated replacements is static since the replacement information source has a constant solution density. However, the threshold k_g associated with replacement by randomly selected survivor parents is dynamic because the solution density of the surviving population increases over generations. Combining Equation (5), which gave a static upper bound on the selection threshold, with Equation (1), which defined the solution density of the initial population, gives

$$k_0 \le \sum_{\lambda=0}^{L} \lambda \mathbf{p}(0, \lambda) = \frac{L}{A} = L\rho_0.$$

Since ρ_0 is the solution density at $g = 0$ and ρ_g is the solution density at $g > 0$, this suggests that, where $g > 0$ and $0 \le k_0 < k_g < L$,

$$k_g = L\rho_g. \tag{8}$$

Therefore, the selection threshold k_0 can be determined using Equation (5) and ρ_1 calculated using Equation (3). Analogously k_g may be calculated with Equation (8) and ρ_g, for $g \ge 2$, with Equation (7).

5.2 Exploring the Dynamic Threshold

We now present a second simulation study to examine the fidelity of the equations modeling information flow, the accurate identification of the selection threshold, and the influence of crossover. However, this time we use Equation (7) to model the effect of using randomly generated individuals and child individuals to replace low-performing individuals, instead of Equation (3).

In the first simulation, the GA had full knowledge of the solution. The motivation was to check the fidelity of the equations modeling the GA behavior. In this second simulation, we make the GA more realistic by reducing its knowledge of the solution. Instead, we use an estimate for solution density that can be derived by the GA from the population without knowledge of the solution (ideal alleles).

We define the *major schema* of the population as the genome comprising the most frequently occurring allele at each locus in the population. This major schema represents the best estimate of the ideal individual at generation g. We also define the *effective solution density* as the frequency of alleles forming the major schema at generation g which permits us to use effective solution density to estimate the selection thresholds (k_0 and k_g) for the simulation. We rank the population by ideal allele as before.

Downloaded from http://direct.mit.edu/evco/article-pdf/18/2/229/1494001/evco.2010.18.2.18203.pdf by guest on 17 June 2021

The selection operator now replaces individuals in generation g that are below the k_0 threshold estimated by effective solution density. Hence the bottom ranked

$$N_{g,k_0} = N_g \sum_{\lambda=0}^{k_0} \mathbf{p}(g, \lambda)$$

individuals are replaced with randomly generated individuals. We then replace the next $N_{g,k_g} - N_{g,k_0}$ individuals with randomly selected individuals from above the threshold k_g where

$$N_{g,k_g} = N_g \sum_{\lambda=0}^{k_g} \mathbf{p}(g, \lambda).$$

The details of the form of GA used in this second simulation are given in Algorithm 2. We ran this more realistic simulation against the same royal road problem as before. As before, the simulations were repeated for two scenarios corresponding to amounts of crossover at either end of the spectrum: (i) $C = 310$ crossover operations per generation and (ii) $C = 10$ crossover operations per generation. This first amount of crossover is chosen arbitrarily based on 10 times the population size.

Algorithm 2 Outline of genetic algorithm with randomly generated and parent replacements

```
Data: Population Size, Termination Criteria, Genome Length=$L$, Cardinality=$A$
Define information source having cardinality $A$;
Calculate $k_0$;
for $n = 1$ to Population Size do
    Generate Individual of length $L$ using information source;
repeat
    Determine major schema;
    Estimate $k_g$ using major schema;
    Score each individual in the population using the objective function;
    Rank population by Score;
    Delete bottom ranked $N_{g,k_g}$ individuals;
    Replace $N_{g,k_0}$ individuals using information source;
    Replace $N_{g,k_g} - N_{g,k_0}$ individuals using randomly selected individuals
    from above $N_{g,k_g}$;
    /* this paper uses $C = 310$ and $C = 10$ for comparison
       purposes                                              */
    for $c = 1$ to $C$ do
        Randomly select sections of length $L/2$ in 2 randomly selected
        individuals and exchange these sections;
until termination criteria = true;
```

Downloaded from http://direct.mit.edu/evco/article-pdf/18/2/229/1494001/evco.2010.18.2.18203.pdf by guest on 17 June 2021

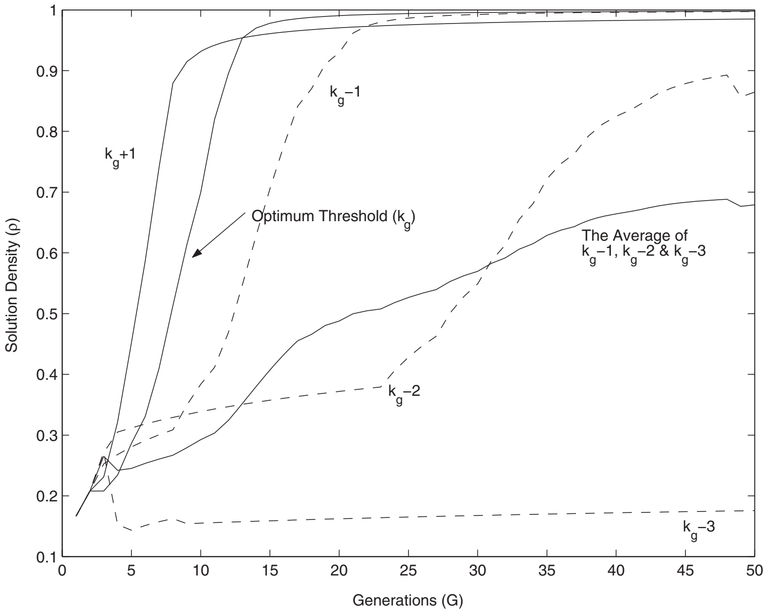


Figure 9: The expected solution density predicted by Equation (7) for the optimal dynamic selection threshold k_g and for $k_g + 1$, $k_g - 1$, $k_g - 2$, $k_g - 3$ and the average of these last three ($N = 31$, $L = 13$, $A = 6$).

The results for each scenario were averaged over 100 trials of the simulation to produce meaningful results. Parameters used for both the theoretical model and simulated GAs are a population size of $N = 31$ individuals, individual length $L = 13$ loci, and allele cardinality of $A = 6$ alleles.

The theoretical results are shown in Figure 9. The results of the simulated GA are shown in Figure 10 (310 crossovers per generation) and Figure 11 (10 crossovers per generation). The simulation marked k_g is where the selection threshold is set at the optimum level as indicated by the model. The simulations marked $k_g + 1, 2, \ldots$, are where the selection threshold was artificially increased by one, two, and so on, ideal alleles per individual for comparison purposes. We again compare the results of a simulated GA with the theoretical results.

5.2.1 Discussion

As before, the model (Figure 9) gives a good estimation of the simulated GA behavior when the number of crossover operations is sufficient (Figure 10). The maximum mean squared error between the model and the simulation for the line $k_g + 1$ and k_g is 0.0758 and 0.1483 respectively. This is greater than the maximum mean squared error in simulation 1 due to the relaxed assumptions of the simulation.

In this simulation, the threshold is set imperfectly and some individuals with low solution density survive to the next generation. This is most clearly seen where we deliberately lower the dynamic threshold to below the predicted optimum (Figure 10 line $k_g - 1$). This line resembles the average of lines $k_g - 1$, $k_g - 2$, and $k_g - 3$, from

Downloaded from http://direct.mit.edu/evco/article-pdf/18/2/229/1494001/evco.2010.18.2.18203.pdf by guest on 17 June 2021

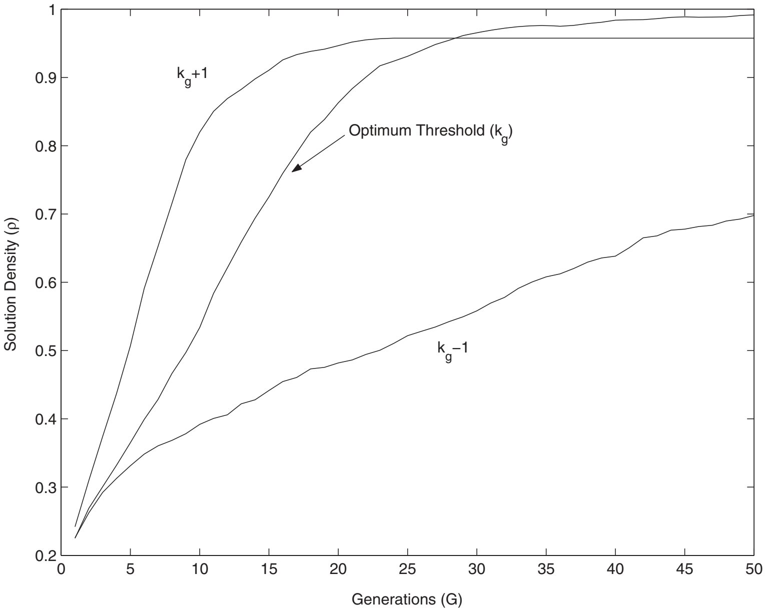


Figure 10: The average results of 100 trials where crossover has been performed between randomly selected pairs of individuals 310 times ($N = 31$, $L = 13$, $A = 6$).

the model shown in Figure 9. This occurs because the inaccurate threshold means that some individuals with much less than the targeted $k_g - 1$ ideal alleles survive to the next generation.

When selection pressure exceeds the calculated selection threshold k_g, the model predicts an increased rate of improvement in solution density, which reaches a lower maximum solution density than the optimum. To see this, compare the k_g and $k_g + 1$ lines in Figure 9. We see this behavior in the simulation with 310 crossovers (Figure 10, lines k_g and $k_g + 1$). Again, the simulation is affected by the inaccurate threshold setting.

When we look at the simulation with less crossover (Figure 11) the model (Figure 9) does not predict the maximum performance of the GA as well as before. This is especially evident for the line $k_g + 1$ which reaches a significantly lower solution density in the simulation than predicted by the model. This is because there is not sufficient crossover in this simulation to mix the ideal alleles introduced by the randomly generated replacements into the population. Instead, they remain in low scoring random individuals and are selected out the very next generation, leading to increased information loss and the stalling effect apparent in the flattening $k_g + 1$ line.

However, the reduced crossover also means that this simulation is less affected by the survival of individuals with low solution density as their deleterious alleles are not mixed through the remaining individuals to the degree that occurred when more crossover was applied. This is evidenced by the low crossover simulation (Figure 11) having a more rapid improvement in solution density when compared to the high crossover simulation (Figure 10), especially when a deliberately low threshold of $k_g - 1$ is set (compare the Figure 10, $k_g - 1$ line to the Figure 11, $k_g - 1$ line).

Downloaded from http://direct.mit.edu/evco/article-pdf/18/2/229/1494001/evco.2010.18.2.18203.pdf by guest on 17 June 2021

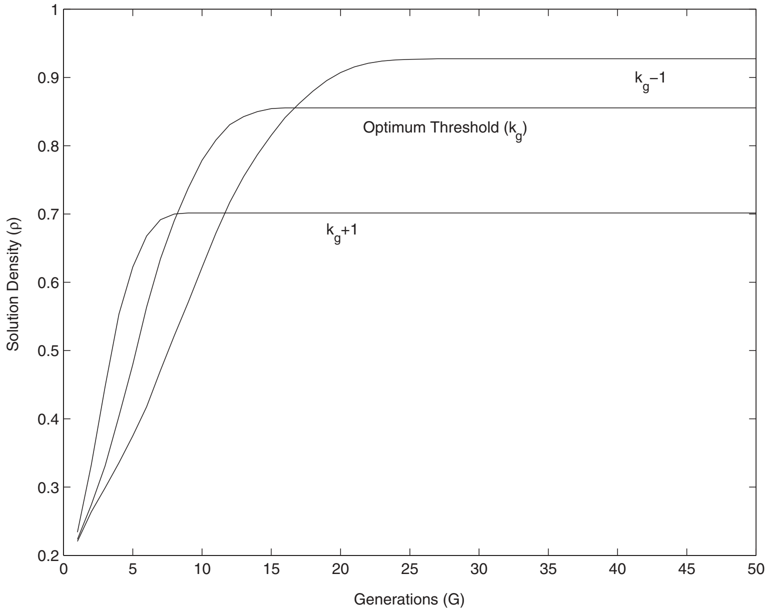


Figure 11: The average results of 100 trials where crossover has been performed between randomly selected pairs of individuals 10 times ($N = 31$, $L = 13$, $A = 6$).

6 Performance in Selected Benchmark Problem

We now develop a GA utilizing these ideas and apply it to a benchmark problem commonly used to test GAs. The algorithm has no knowledge of the optimum for either ranking or threshold setting. The maximum schema as described by Section 5 is used to locate the static (k_0) and dynamic (k_g) thresholds while the bit trap objective function score is used to rank individuals in the population.

We examine two differently sized problems. The first of the selected benchmarks is based on 10 concatenated 6-bit traps. The bit trap problem is a "deceptive" version of the counting 1s problem. In the bit trap problem, the fitness of an individual is the number of 1s it contains, unless it is all 0s, in which case the individual's fitness is $L + 1$. The problem is deceptive because the algorithm is rewarded incrementally for each 1 it adds to individuals, but the optimum solution consists of all 0s.

The results for this small benchmark are shown in Figure 12. In the second benchmark, we increase the problem size to 60 concatenated 6-bit traps. Results for this larger bit trap problem are shown in Figure 13.

The first benchmark of 10 concatenated 6-bit traps produces an $L = 60$ bit problem as described by Harik (1999).[4] A population size of $N = 439$ is used and five trials are completed using Algorithm 2. Note that all five trials in Figure 12 approach the maximum solution density very quickly. A total of 5,707 evaluations are completed in the $G = 13$ generations required to converge to the optimum. This result compares favorably with Harik's result for a 4-bit trap, 40-bit problem which required 4,000

[4] Although Harik (1999) used 10 concatenated 4-bit traps.

Downloaded from http://direct.mit.edu/evco/article-pdf/18/2/229/1494001/evco.2010.18.2.18203.pdf by guest on 17 June 2021

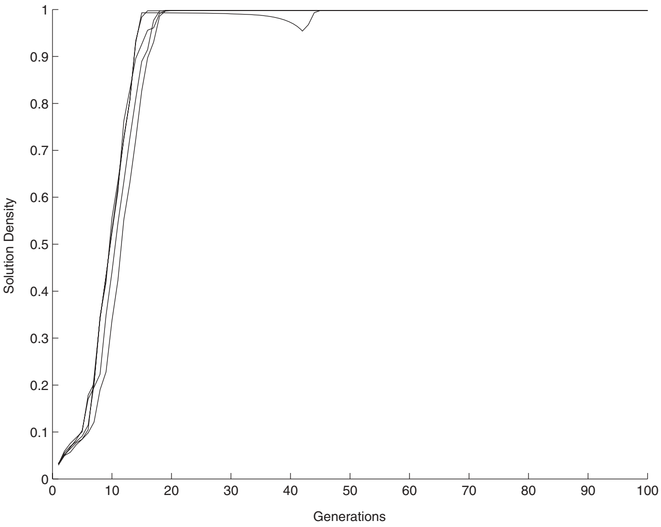


Figure 12: Five trials of a problem formed of 10 concatenated 6-bit traps. The GA has a length of $L = 60$ bits and a population size of $N = 439$. A total of 4,390 evaluations are completed in $G = 10$ generations.

evaluations and a population size of 500 with an extended compact genetic algorithm (ECGA) achieving a similar result (0.93 of the optimum).

As an $L = 60$ bit problem is relatively simple, we increase the problem size from $L = 60$ bits to $L = 360$ bit-concatenated 6-bit traps for the second benchmark (Figure 13). In this second benchmark, 22,120 evaluations are completed in the $G = 40$ generations required to converge to 0.95 of the optimum. This is an encouraging result for a problem described by Harik as "a difficult, partially deceptive problem."

7 Future Work

The primary challenge with the approach we have described is the ranking of individuals by ideal allele content. Accurate estimation of the dynamic threshold is important to keep the selection pressure at a high level, but this estimate will be compromised by inaccurate ranking of the population. Hence it is important to determine the sensitivity of the approach described by this paper to errors in the rank order. We will address this challenge by linking the entropy of ranked populations to selection thresholds.

8 Conclusions

This paper has identified both static (k_0) and dynamic (k_g) selection thresholds with $k_0 \leq k_g$ in ranked lists of individuals in a population. These thresholds are related to the information content of the information sources used to generate replacement individuals and the accumulation of information in the GA. By replacing individuals ranked below the static threshold with randomly generated replacements, the need for mutation is avoided while diversity is maintained. By replacing individuals ranked

Downloaded from http://direct.mit.edu/evco/article-pdf/18/2/229/1494001/evco.2010.18.2.18203.pdf by guest on 17 June 2021

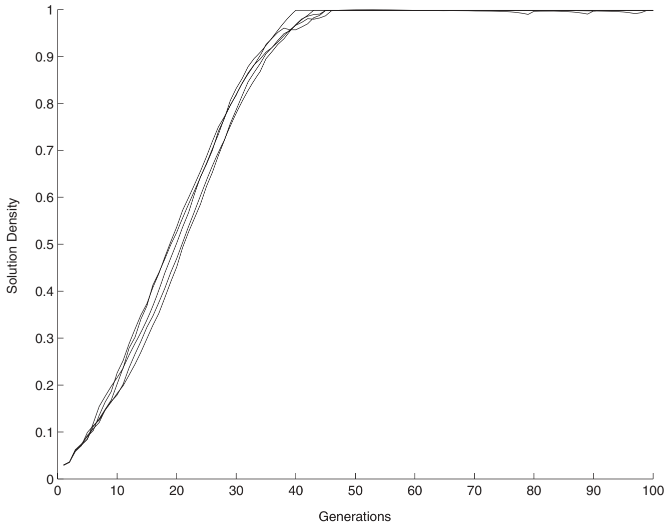


Figure 13: Five trials of a problem formed of 60 concatenated 6-bit traps. The GA has a length of $L = 360$ bits and a population size of $N = 553$. A total of 5,530 evaluations are completed in $G = 10$ generations.

between the static and dynamic selection threshold with randomly selected individuals from above the dynamic threshold, selection pressure may be controlled to achieve a balance between the rate of improvement and population diversity.

By modeling the change in a population's solution density when subject to varying amounts of crossover, it was shown that large amounts of crossover (or a UEDA) are superior to insufficient crossover when the location of the thresholds is uncertain. This is especially the case where the actual threshold is placed at, or slightly above, the optimum selection threshold. If the actual threshold is placed below the optimum threshold, then limited crossover provides better average performance as fewer deleterious alleles are spread through the population.

We make two recommendations to ensure that information accumulates in a GA and to ensure that selection pressure is controlled to achieve a balance between the rate of improvement and population diversity: (i) replace individuals below the static threshold with randomly generated replacements and (ii) replace individuals between the static and dynamic selection thresholds with randomly selected individuals from above the dynamic threshold.

References

Baker, J. E. (1985). Adaptive selection methods for genetic algorithms. In *Proceedings of the 1st International Conference on Genetic Algorithms*, pp. 101–111.

Blickle, T., and Thiele, L. (1995). A mathematical analysis of tournament selection. In *Proceedings of the 6th International Conference on Genetic Algorithms*, pp. 9–16.

Borenstein, Y., and Poli, R. (2006). Information perspective of optimization. *Lecture Notes in Computer Science Parallel Problem Solving from Nature (PPSN)9*, 4193:102–111.

Downloaded from http://direct.mit.edu/evco/article-pdf/18/2/229/1494001/evco.2010.18.2.18203.pdf by guest on 17 June 2021

Box, G. (1957). Evolutionary operation: A method of increasing industrial productivity. *Applied Statistics*, 6:81–101.

Bremermann, H. J., Rogson, M., and Salaff, S. (1966). *Global properties of evolution processes in natural automata and useful simulation*. Washington, DC: Spartan Books.

Fraser, A. (1957). Simulation of genetic systems by automatic digital computers. *Australian Journal of Biological Science*, 10:484–491.

Galvan-Lopez, E., and Poli, R. (2006). Some steps towards understanding how neutrality affects evolutionary search. *Lecture Notes in Computer Science Parallel Problem Solving from Nature (PPSN)9*, 4193:778–787.

Goldberg, D. (1989). *Genetic algorithms in search optimization and machine learning*. Reading, MA: Addison Wesley.

Goldberg, D. (2002). *The design of innovation*. Dordrecht, The Netherlands: Kluwer Academic.

Gonalves, J. F., de Magalhes Mendes, J. J., and Resende, M. G. C. (2005). A hybrid genetic algorithm for the job shop scheduling problem. *European Journal of Operational Research*, 167(1):77–95.

Harik, G. (1999). Linkage learning via probabilistic modeling in the ECGA. Tech. Rep. 99010, University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory (IlliGAL).

Holland, J. (1975). *Adaptation in natural and artificial systems*. Ann Arbor, MI: University of Michigan Press.

Jansen, T., Jong, K. A. D., and Wegener, I. (2005). On the choice of the offspring population size in evolutionary algorithms. *Evolutionary Computation*, 13(4):413–440.

Jong, K. D., and Spears, W. (1991). An analysis of the interacting roles of population size and crossover in genetic algorithms. *Parallel Problem Solving from Nature, Proceedings of 1st Workshop*, 496:38–47.

Koza, J. (1997). *Genetic programming*. New York: Marcel Dekker.

Kreyszig, E. (1983). *Advanced engineering mathematics*, 5th ed. New York: John Wiley.

Milton, J., Kennedy, P., and Mitchell, H. (2005). The effect of mutation on the accumulation of information in a genetic algorithm. In S. Zhang and R. Jarvis (Eds.), *AI 2005: Advances in Artificial Intelligence, 18th Australian Joint Conference on Artificial Intelligence, Lecture Notes in Artificial Intelligence (LNAI) 3809*, pp. 360–368. Berlin: Springer-Verlag.

Mitavskiy, B. (2004). Crossover invariant subsets of the search space for evolutionary algorithms. *Evolutionary Computation*, 12(1):19–46.

Mitchell, M., Forrest, S., and Holland, J. (1992). The royal road function for genetic algorithms: Fitness landscapes and GA performance. *Proceedings of the First European Conference on Artificial Life*, pp. 245–255.

Muhlenbein, H., and Paaß, G. (1996). From recombination of genes to the estimation of distributions I. Binary parameters. *Lecture Notes in Computer Science 1411: Parallel Problem Solving from Nature (PPSN)*, IV:178–187.

Muhlenbein, H., and Voigt, H.-M. (1995). Gene pool recombination in genetic algorithms. In *Proceedings of the International Conference on Metaheuristics*, Vol. 1484, pp. 53–62.

Ochoa, G. (2006). Error thresholds in genetic algorithms. *Evolutionary Computation*, 14(2):157–182.

Poli, R. (2001). General schema theory for genetic programming with subtree-swapping crossover. In J. F. Miller, M. Tomassini, P. L. Lanzi, C. Ryan, A. G. B. Tettamanzi, and

Downloaded from http://direct.mit.edu/evco/article-pdf/18/2/229/1494001/evco.2010.18.2.18203.pdf by guest on 17 June 2021

W. B. Langdon (Eds.), *Genetic Programming, Proceedings of EuroGP'2001*, Vol. 2038, pp. 143–159. Berlin: Springer-Verlag.

Poli, R., McPhee, N. F., and Rowe, J. E. (2004). Exact schema theory and Markov chain models for genetic programming and variable length genetic algorithms with homologous crossover. *Genetic Programming and Evolvable Machines*, 5(1):31–70.

Rechenberg, I. (1973). *Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der biologischen Evolution*. Stuttgart, Germany: Frommann-Holzboog.

Reeves, C. (1993). Using genetic algorithms with small populations. In S. Forrest (Ed.), *Proceedings of the 5th International Conference on Genetic Algorithms*, pp. 92–99.

Rowe, J. (2001). A normed space of genetic operators with applications to scalability issues. *Evolutionary Computation*, 9(1):25–42.

Rowe, J. E., Vose, M. D., and Wright, A. H. (2002). Group properties of crossover and mutation. *Evolutionary Computation*, 10(2):151–184.

Rowe, J. E., Vose, M. D., and Wright, A. H. (2004). Structural search spaces and genetic operators. *Evolutionary Computation*, 12(4):461–493.

Rowe, J. E., Vose, M. D., and Wright, A. H. (2007). Neighborhood graphs and symmetric genetic operators. *Lecture Notes in Computer Science: Foundations of Genetic Algorithms*, 4436:110–122.

Schwefel, H. (1981). *Numerical optimization of computer models*. New York: John Wiley.

Schwefel, H. (1995). *Evolution and optimum seeking*. New York: John Wiley.

Spears, W. M., and De Jong, K. A. (1991). On the virtues of parameterized uniform crossover. In R. Belew and L. Booker (Eds.), *Proceedings of the Fourth International Conference on Genetic Algorithms*, pp. 230–236.

Toussaint, M. (2004). Notes on information geometry and evolutionary processes. *Arxiv preprint nlin.AO/0408040*.

Van der Lubbe, J. (1997). *Information theory*. Cambridge, UK: Cambridge University Press.

Weaver, W., and Shannon, C. E. (1949). *The mathematical theory of communication*. Champaign, IL: University of Illinois Press.

Whitley, D. (1989). The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best. In *Proceedings of the 3rd International Conference on Genetic Algorithms*, pp. 116–123.

Whitley, D., Mathias, K., and Pyeatt, L. (1995). Hyperplane ranking in simple genetic algorithms. In *Proceedings of the 6th International Conference on Genetic Algorithms*, pp. 231–238.

Wright, A. H., and Richter, J. N. (2006). Strong recombination, weak selection, and mutation. In *GECCO '06: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, pp. 1369–1376.

Appendix Calculating Sufficient Crossover

Our estimates of expected solution density require sufficient crossover to return the distribution of ideal alleles in the population to a binomial distribution. This appendix provides an estimate of sufficient crossover. The calculations performed here determine the minimum number of crossover operations C that return a distribution of ideal alleles in a population subject to selection to a binomial distribution. We shall consider

Downloaded from http://direct.mit.edu/evco/article-pdf/18/2/229/1494001/evco.2010.18.2.18203.pdf by guest on 17 June 2021

only uniform crossover in the following analysis to avoid edge effects, such as defining length bias (Spears and De Jong, 1991).

The problem is to determine the number of crossover operations C required to change the truncated distribution $\mathbf{p}^L_{k+1}(g, \lambda)$ with selection threshold k (Figure 3) into the binomial distribution $\mathbf{p}(g+1, \lambda)$ (Figure 4) by exchanging S alleles per operation. To do this, we must first define an intermediate distribution ψ_c as the distribution after c crossover operations have been performed on $\mathbf{p}^L_{k+1}(g, \lambda)$ and before the binomial distribution $\mathbf{p}(g+1, \lambda)$ has been reached.

Each distribution ψ_c may be represented by a vector of length $L+1$, where each element of the vector represents the proportion of individuals in the population that contain $\{\lambda | 0 \leq \lambda \leq L\}$ ideal alleles. Hence, a matrix of probabilities Ψ_c can be constructed whereby each cell in the matrix represents the joint probability that two individuals, one with λ_1 and the other with λ_2 ideal alleles, are randomly selected for crossover. This joint probability matrix is given by $\Psi_c = \psi_c^T \psi_c$, where ψ_c^T is the transpose of the vector ψ_c.

Similarly, a hyper geometric distribution[5] $\mathbf{w} = w(\lambda_s | L, \lambda, S)$ exists that describes the distribution of ideal alleles $\{\lambda_s | 0 \leq \lambda_s \leq s\}$ among S alleles in each crossed-over section exchanged by the randomly chosen individuals (where s is the number of exchanged ideal alleles S or λ, whichever is the least).

Again, a probability matrix $\mathbf{W}$ can be constructed where each cell represents the joint probability that $0 \leq \lambda_s \leq s$ ideal alleles are among the S exchanged alleles. This matrix is given by $\mathbf{W} = \mathbf{w}^T \mathbf{w}$.

The probability that individuals with $\{\lambda | 0 \leq \lambda \leq L\}$ ideal alleles are chosen for crossover, and then sections containing $\{\lambda_s | 0 \leq \lambda_s \leq s\}$ ideal alleles are exchanged by crossover, can be found by taking the Kronecker tensor product $\Psi_c \otimes \mathbf{W}$. Each cell of the matrix $\Psi_c \otimes \mathbf{W}$ represents a transition probability from the distribution ψ_c to another distribution π formed by the exchange of λ_s ideal alleles between individuals containing λ ideal alleles.

By constructing each of these possible π distributions, then multiplying them by the appropriate transition probability from $\Psi_c \otimes \mathbf{W}$ and summing the resulting expected distributions $\mathbb{E}[\pi]$, the expected distribution $\mathbb{E}[\psi_1]$ of a single crossover operation is produced.

Repeating the process using $\mathbb{E}[\psi_1]$ in place of ψ_0 and counting how many iterations c are required before the intermediate Ψ_c distribution equals the required binomial distribution $\mathbf{p}(g+1, \lambda)$ then the sufficient number of crossover operations C can be determined. In addition, one can compare the number of crossover operations required for differing numbers of exchanged alleles S.

One difficulty with this approach is determining when the intermediate ψ_c equals the required binomial distribution $\mathbf{p}(g+1, \lambda)$. We calculated a mean square error $[\sum(\psi_c - \psi_{c+1})^2]^{\frac{1}{2}}$ and continued until this error was less than 10^{-9} in a single crossover operation. The standard χ^2 goodness of fit test with a confidence level of 99% (Kreyszig, 1983) was then applied to each intermediate distribution to decide when ψ_c equaled the binomial distribution $\mathbf{p}(g+1, \lambda)$ and hence identify C.

[5] The hypergeometric distribution models the number of ideal alleles λ_s in the S alleles, exchanged without replacement from the total ideal alleles λ in a parent individual with L loci. $w(\lambda_s | L, \lambda, S) = \frac{\binom{\lambda}{\lambda_s}\binom{L-\lambda}{S-\lambda_s}}{\binom{L}{S}}$.

Downloaded from http://direct.mit.edu/evco/article-pdf/18/2/229/1494001/evco.2010.18.2.18203.pdf by guest on 17 June 2021

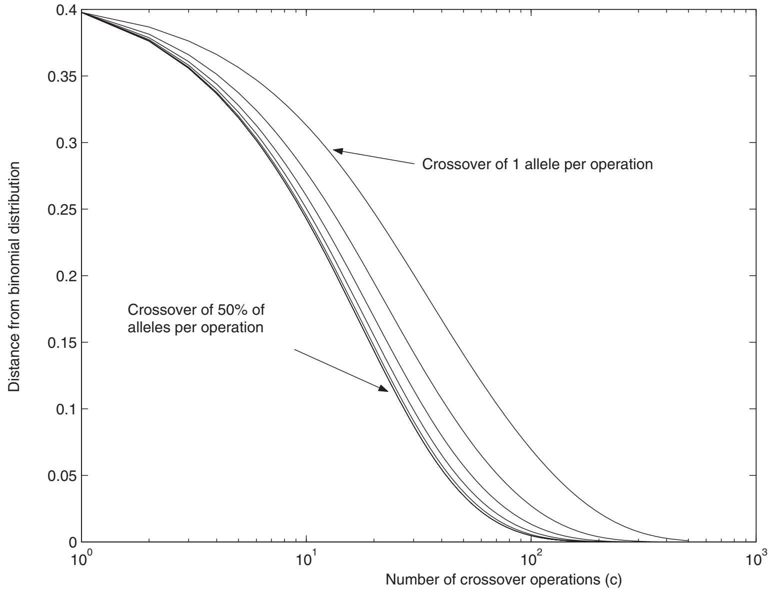


Figure 14: The distance between the distribution ψ_c and the distribution $\mathbf{p}(g+1, \lambda)$ for a population size of $N = 31$, genome length of $L = 13$, and alleles exchanged ($S = 1$ to 7) per crossover operation.

Figure 14 shows the change in the distance between the distribution ψ_c and the distribution $\mathbf{p}(g+1, \lambda)$ per crossover operation for the six cases where $\{S|S \in \mathbb{Z}_+ : 1 \leq S \leq L/2\}$. As illustrated by Figure 14, the most effective way to redistribute ideal alleles in a population altered by selection to a randomized (binomial) distribution is to use crossover section lengths $S = L/2$. The distribution is sufficiently close to a binomial distribution, as determined by a 99% χ^2 test, after $5N$ crossover operations (where N is the population size). Therefore, sufficient crossover occurs at approximately $C = 5N$ for crossover section lengths of $L/2$.

Downloaded from http://direct.mit.edu/evco/article-pdf/18/2/229/1494001/evco.2010.18.2.18203.pdf by guest on 17 June 2021