Symmetry Breaking in Population Based Optimisation

Adam Prügel-Bennett
ECS, University of Southampton, SO17 1BJ, United Kingdom

25th January 2004

Abstract

This paper argues that the performance of evolutionary algorithms working on hard optimisation problems depends strongly on how the population breaks the 'symmetry' of the search space. The splitting of the search space into widely separate regions containing local optima is a generic property of a large class of hard optimisation problem. This phenomenon is discussed by reference to two well studied examples, the Ising perceptron and K-SAT. A finite population will quickly concentrate on one region of the search space. The cost of crossover between solutions in different regions of search space can accelerate this symmetry breaking. This, in turn, can dramatically reduce the amount of exploration, leading to sub-optimal solutions being found. An analysis of symmetry breaking using diffusion model techniques borrowed from classical population genetics is presented. This shows how symmetry breaking depends on parameters such as the population size and selection rate.

Keywords

Symmetry breaking, landscapes, replica symmetry, diffusion equations, crossover, evolutionary algorithms.

1 Introduction

A major challenge in developing a theory of evolutionary algorithms (EAs) is to provide guidance for their design. For example, to determine parameters such as population size or the selection pressure, as well as suggesting improved algorithms through the use of structured populations, mating taboo or niching strategies, etc. In this paper, we argue that a crucial element in addressing these issues is to understand how the population in an EA breaks the 'symmetry' of the search space.

The symmetry breaking this paper addresses occurs when a population concentrates on one region of the search space due to chance. If the EA was run again it might end up concentrated at another region of search space with a similar fitness. This chance or spontaneous symmetry breaking is a finite population effect—in an infinite population, regions that are similar in fitness value and size would be explored by approximately the same proportion of the population. In problems with exact symmetries, this spontaneous symmetry breaking of the population might be beneficial. However, in hard optimisation problems the search space is frequently shattered into many regions containing local optima. Spontaneous symmetry breaking will focus the search to a particular region of the search space. When this occurs too rapidly it can be detrimental as it will reduce the amount of exploration that an EA performs.

In the rest of this paper, we build up the argument outlined above in the following series of steps

1. A toy problem is presented for which an EA using crossover substantially outperforms other optimisation algorithms

2. We argue that this does not happen in most real problems because there are many separate regions in search space with good solutions which provide contradictory 'gradient' information to the population.

3. We illustrate how the search space can be split into symmetric regions by considering two well known problems with an explicit symmetry in the search space. Explicit symmetries can cause problems for evolutionary algorithms using crossover.

4. Some recent results are reviewed suggesting that splitting of the search space into regions is a generic property of a large class of combinatorial optimisation problems. We illustrate the nature of this splitting on two NP-hard problems, the Ising perceptron and K-SAT.

5. Finally, three toy problems, related to that discussed in step 1, but which have an explicit symmetry or near symmetry, are considered. Simulation results are presented for these model to illustrate how population based search algorithms behave in spaces with symmetry. An analytical model of the symmetry breaking is developed using stochastic differential equations.

The aim of this paper is to gain a greater understanding of the performance of evolutionary algorithms on typical hard optimisation problems. The toy problems we consider are clearly artificial, but are designed to capture some of the features of real problems. It is the author's view that controlling the balance between breaking the search space symmetries and maintaining diversity is crucial to determining optimal parameters, understanding mating taboos, niching strategies and population structures. This is a speculative view based on the author's experience. Empirical evidence supporting this view is difficult to obtain because it requires a knowledge of the landscape structure in large optimisation problems which is hard to ascertain. Studies using barrier trees discussed in section 5.1 seem to support the thesis but are far from conclusive. The strength of the case being made rests on the plausibility of the chain of arguments being made. This paper marshals together evidence to support this chain of reasoning.

The existence of exact symmetries in optimisation problems and the difficulties that these can cause for evolutionary algorithms with crossover are well known. We briefly revisit this problem to make contact with these well established ideas. However, the main focus of the current paper is on problems with approximate symmetries arising from the splitting of the search space into regions containing local optima, which we argue is a generic feature of a large class of hard optimisation problems. Whereas rapid symmetry breaking in problems with exact symmetries is often desirable, we argue that this is not so in problems with approximate symmetries. If the symmetry breaking is too rapid the population will have little time to explore different regions and so might end up in a sub-optimal region. If this line of argument is correct then a key issue in designing an evolutionary algorithm is to ensure that symmetry breaking is retarded. Many of the arguments given below should apply to any population based algorithm that uses crossover, however, to make the argument more concrete we give examples using a simple genetic algorithm (GA).

The next five sections of the paper expand on each of the steps in the argument outlined above. The final section draws together the threads of the previous sections to argue why symmetry breaking is such an important issue when considering population based algorithms.

2 Where EAs Do Well

We begin by showing there are classes of problems where evolutionary algorithms do well. A problem is presented where a genetic algorithm with crossover substantially outperform any other generic optimisation algorithm that the author is aware of. This example suggests that evolutionary algorithms might be expected to be very effective on problems with rough fitness landscapes. Unfortunately, this is rarely the case, but it raises the question of why EAs do not perform so well on real problems. We will argue that it is the splitting of the search space into regions that make many optimisation problems hard for EAs.

There are several problems published in the literature where genetic algorithms with crossover can substantially outperform other optimisation algorithms [1, 2, 3, 4, 5, 6]. Here, we describe the Hurdle problem first proposed in [5]. The problem is to optimise a binary string, $\boldsymbol{x} = (x_1, x_2, \ldots, x_L)$, with $x_i \in \{0, 1\}$. The fitness is a function of the Hamming distance from some target string $\boldsymbol{x}^*$. For convenience we will take the target string to be the string of all ones, so that the Hamming distance is equal to the number of zeros. The fitness function is given by

$$F(H) = L - H - 2\left[H \text{ is odd}\right]$$

where H is the Hamming distance and where we have used square brackets to denote the indicator function (Iverson's convention)

$$\left[\textit{predicate}\right] = \begin{cases} 1 & \text{if } \textit{predicate} \text{ is true} \\ 0 & \text{if } \textit{predicate} \text{ is false.} \end{cases}$$

The fitness function for the Hurdle problem with $L = 20$ is shown in figure 1.

The fitness landscape consists of an average coarse-grain 'gradient' sloped towards the global optimum which is hidden by local roughening. (There is no real gradient as we are dealing with a discrete search space, however, we will informally use the term gradient to denote a tendency of the fitness function to increase as we approach the optimum solution.) Figure 1 suppresses information on the multiplicity of solutions. There are exponentially many more solutions for $H \approx L/2$ than for $H \approx 0$. The landscape may be more accurately pictured as consisting of a hill with a very narrow summit at $H = 0$ (there is also a dip at $H = L$). The closer we are to the summit the more likely it is that

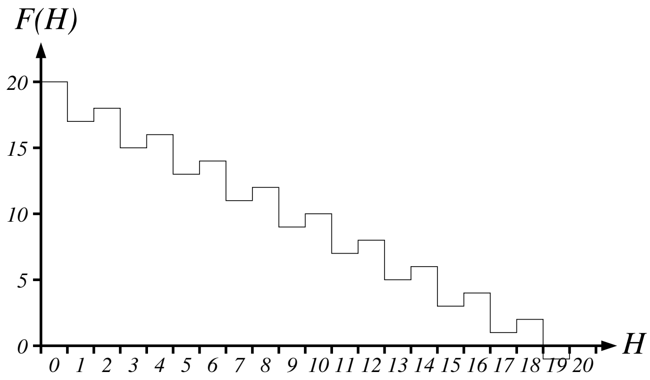


Figure 1: Fitness as a function of the Hamming distance for the Hurdle problem with $L = 20$.

a path will lead down hill. Imposed on this structure is a constant undulation hiding the gradient information.

To get a feel for the difficulty of this problem we consider how long it would take for a hill-climber to solve it. We start the hill-climber by selecting a random string. With high probability, the random string will be at a Hamming distance $H \approx L/2$ away from the global optimum. If the hill-climber was permitted to only change a single string variable, x_i, at a time, it would immediately get stuck. However, if occasionally the hill-climber could mutate two string variables in one move, then it could progress from one local optimum to another towards the global optimum. The probability of a double mutation moving the hill-climber from a state H to a state $H-2$ is equal to $H(H-1)/(L(L-1))$. This probability is smallest when making the last jump from a state $H = 2$ to the global optimum. A hill-climber will typically take longer going from the state $H = 2$ to $H = 0$ than it takes to move from wherever it started to the state $H = 2^*$. To obtain a rough estimate for the average first-passage time to reach the global optimum we therefore need only consider the last step. The average number of double mutations attempted in order to make this last step is $L(L-1)/2$. Thus the average first-passage time scales as $\Theta\left(L^2\right)$. We can generalise the model so the Hamming distance between local optima is n. In the following we assume $n = O(1)$. In this case, the average first-passage time is $\Theta\left(L^n\right)$. Simulated annealing (with single mutations) does even worse on this problem than a hill-climber. An argument based on assuming a succession of partial equilibria suggests that simulated annealing takes $\Theta\left(L^{n+1}\right)$. The scaling is subtle, but the reason why the first-passage time for simulated annealing is longer that hill-climbing is easy to see. Consider the case $n = 1$. In simulated annealing the stochastic hill-climber must move from a state $H = 2$ to $H = 1$. This is improbable because it involves mutating one of the two zeroes in the string and because it costs fitness so it is likely to be rejected. Having made this first improbable move the stochastic hill-climber must make a second improbable move from $H = 1$ to $H = 0$ rather than the much more probable step back to $H = 2$. Thus, as well as the entropy barrier, the stochastic hill-climber faces a fitness barrier, which the hill-climber, making larger mutations, could jump. Full details of this analysis are given

*The mean transition times measured in terms of the number of double mutation from $H = 4$ to $H = 2$ is $L(L-1)/(4\cdot 3)$ and from $H = 6$ to $H = 4$ it is $L(L-1)/(6\cdot 5)$, etc. Thus the mean transition time from $H = L/2$ to $H = 2$ is $L(L-1)\left(1/(4\cdot 3) + 1/(6\cdot 5) + \cdots\right) < L(L-1)\left(5/6 - \log(2)\right) \approx 0.14L(L-1)$ where the bound becomes tight when $L \to \infty$.

in [5].

In contrast, a genetic algorithm with crossover can relatively easily combine two strings to reach the global optimum. For example, suppose we have a population which contains the two strings shown in figure 2.

1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1
1	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1

Figure 2: Example of two strings a Hamming distance 2 away from the state of all ones.

If we perform uniform crossover on these two strings the probability of the offspring being the global optimum is $1/16$—that is, independent of L and (provided L is large) much greater than the probability of $2/(L(L-1))$ for a double mutation to make the same jump. Crossover facilitates the search because it focuses the search on the sites where the strings disagree. On average, an offspring will be at the same Hamming distance from the optimal solution as its parents. This contrasts sharply with mutation which, on average, moves the solution to the state where $H = L/2$. Single-point crossover will be slightly slower than uniform crossover (it would not be able to reach the global optimum for the strings in figure 2 in a single step), however, it still focuses the search and will be a much more effective search operator than mutation for long strings provided the gap, n, is small. Note, however, that in some problems the type of crossover can sometimes make a significant difference when trying to piece together a good solution [6].

No mathematical analysis of the performance of a GA on the Hurdle problem has been carried out. However, empirically a GA with crossover was found to substantially outperform a hill-climber. Furthermore, the first-passage time appeared to scale sub-quadratically[†], and at the same rate as a GA acting on the ONESMAX problem (i.e. with $F(H) = L - H$) where there is an uninterrupted gradient to the global solution. Furthermore, the first-passage time appeared to have the same scaling behaviour when the gap, n, between local optimum is greater than two (we examined the cases $n = 2$, 3, and 4). The main requirement for a GA to solve this type of problem efficiently is that the population is sufficiently diverse that the zeroes do not occur at the same sites for all members of the population. This requirement is relatively simple to satisfy for this problem, at least, if L is not excessively large. Mutations will maintain the diversity and prevent fixation of a zero at any one site. In a related problem of a two well potential, it has been shown that almost all the search is performed by the crossover operator [2, 3]. Mutation tends to be detrimental to the search, except in so far as it maintains diversity. Larger populations will have greater diversity and can thus bear a lower mutation rate. In consequence, the optimal mutation rate for this related problem was shown to be inversely proportional to the square root of the population size [3].

In summary, it is easy to construct a GA which significantly outperforms a hill-climber on the Hurdle problem. Furthermore, the performance of the GA was found to be fairly robust to changes of parameters such as the population size, selection strength or mutation rate.

[†]This scaling behaviour was observed for problems of size up to $L \approx 500$, for a GA with a population of 100.

3 Hard Optimisation Problems

We might expect from the Hurdle problem example that EAs with crossover would perform very well on many hard optimisation problems with roughened fitness landscapes. Unfortunately, the substantial performance advantages of an EA on the Hurdle problem is seldom found in real world optimisation problems. EAs rarely outperform other optimisation techniques without considerable tuning of parameters and the use of sophisticated search operators embedded within the EA. In this paper, we argue that for a large class of important problems, including many of the classic combinatorial optimisation problems, the reason why EAs do not perform as well as the Hurdle problem might suggest is that the search space is split into a large number of regions each containing a local optimum. Regions containing good quality solutions can be quite distant from each other. There is therefore not a single overall gradient for a population to follow.

An initial population will lie in many different regions of search space. As the population evolves, fitter members will replace less fit members. Crossover between solutions belonging to different regions will often produce poor quality offspring as the building blocks from the parents are trying to solve the problem in different ways. This gives a strong advantage to members of the population in the most populous region since they are more likely to cross with each other. Crossover thus produces a strong pressure for 'symmetry breaking'. Any initial small random preference of the population for one region of the search space is rapidly multiplied until all members of the population lie in that same region. This symmetry breaking may have nothing to do with the quality of solutions in that region—if the EA was run again it is probable that the population would finish focused on another region of search space. Once the population lies in a single region, exploration of the search space beyond the region is strongly inhibited because of the cost of crossover. Furthermore, the initial symmetry breaking can cause a significant loss of diversity within a very few generations.

It should be noted that the eventual focusing of the population around one region is a natural consequence of the population correlating as it evolves. Even without any selection pressure or crossover the population will correlate due to genetic drift. This is very pronounced in small populations. Selection will increase the correlation because good members of the population are being replicated and replace the less fit, but also less correlated, poor members. The crossover induced symmetry breaking described above just accelerates this process. Crossover is often regarded as a global search operator as if allows much larger jumps in terms of Hamming distance than mutation. However, in a correlated population crossover only produces changes on sites where the parents differ. It therefore acts as a focused search operator. The average position and correlation of a population is unchanged by most common crossover operators [7]. Thus, once a population has become focused in a region of the search space, crossover does not tend to shift it. Since the population typically evolves to a region of search space with an unusually high incidence of good quality solutions any search operator that produces a large random mutation is likely to arrive in a region of average fitness which will typically be much less fit than most of the members in the current population. In many problems there exists search operators which exploit some structure of the problem which may allow large steps in the search space. Such operators are often highly advantageous precisely because they prevent the population from becoming stuck in one region of search space.

Before studying the behaviour of EAs on fitness landscapes with many similar good quality regions, we return to the question of why these types of landscapes occur so often. We start by considering problems with an explicit symmetry before considering the emergence of hidden 'symmetries' that seem to be a hall mark of many hard optimisation problems.

4 Explicit Symmetries

Explicit symmetries provide a useful model of how the search space can split into similar regions. Two well known problems with explicit symmetries are the graph-colouring problem and feed-forward neural networks. We consider the graph colouring problem where we assign colours to vertices and are penalised for each edge whose vertices have the same colour. In this problem, there is an explicit colour permutation symmetry. Figure 3 illustrates two colourings of a graph that are identical up to a permutation of the colours.

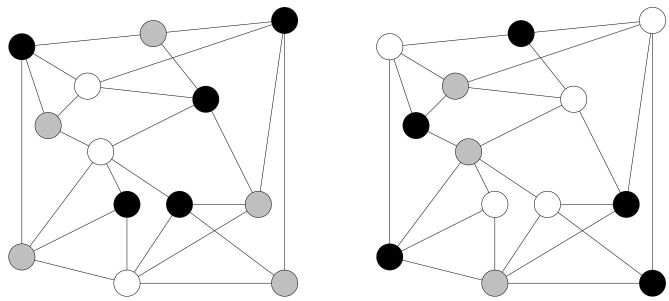


Figure 3: Example of two coloured graphs which differ by a permutation of the colours.

A naive crossover operator can take two perfectly coloured graphs, such as those shown in figure 3, and produce a poor quality offspring containing many edges with identically coloured vertices. Although crossover is reversible so that it could take two poor quality solutions and produce a very high quality solution, the probability of this happening is very low because there are so many more poor quality solutions than good solutions. The size of the permutation symmetry grows very rapidly with the number of colours used to colour the graph. For large graphs the number of colours needed to achieve a colouring with no colour conflicts can be very high. For example, a commonly used test problem for graph colouring are random instances of graphs from the class $\mathcal{G}_{1000,0.5}$, consisting of graphs with 1000 vertices where the probability of an edge is 0.5. The lowest number of colours so far found which gives a colouring of a typical instance of $\mathcal{G}_{1000,0.5}$ with no edge connecting identically coloured vertices is 83 [8, 9]. The number of permutation symmetry for 83 colours is $83! \approx 3.9 \times 10^{124}$. There are a large number of real problems which have so many symmetric regions that for any practical EA it is very unlikely for two members of the population to be in the same symmetric region. For these problems there will be a large cost for using crossover.

In practice, EAs using a naive crossover operator work extremely poorly on the graph-colouring problem. However, the permutation symmetry is only a consequence of representing this problem as a colouring problem. We can view the graph-colouring problem as a partitioning problem where we try to partition the vertices to minimise the number of edges between the vertices within each partition. A crossover operator based on this idea was proposed by Galinier and Hao and, when used in combination with a local search operator, is currently the best known technique for graph-colouring [8, 9].

Multi-layer perceptron (MLP) neural networks are another well known problem exhibiting a permutation symmetry. We consider an MLP with one hidden layer. An example of this is shown in figure 4.

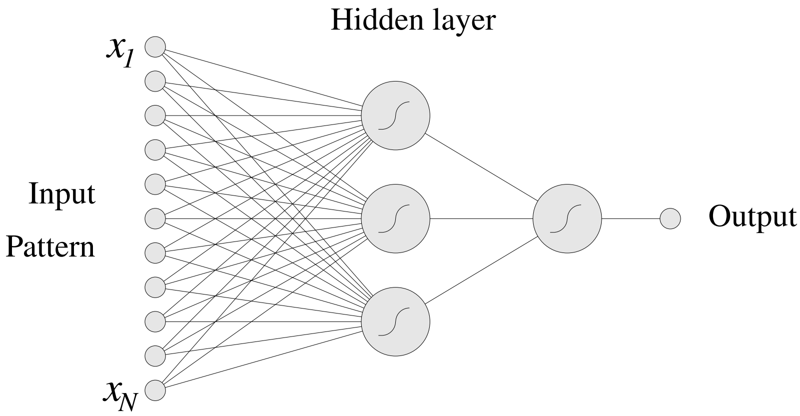


Figure 4: Example of a multi-layer perceptron.

The neurons in the hidden layer can each be characterised by a dividing plane in the space of input patterns (their response depends on the distance of the pattern from the dividing plane). However, we can obtain an equivalent neural network by permuting the neurons in the hidden layer and their corresponding output weights. Again, EAs with naive crossover perform poorly on this type of problem because they mix potentially good dividing planes—this has been called the 'problem of competing conventions' [10].

These problems illustrate what we mean by symmetry breaking. If we were to run an EA on these problems we would expect the population to focus on one of the symmetry regions (in contrast, an infinite population would explore each symmetry region equally). Rapid symmetry breaking appears desirable in these problems as it would reduce the cost of crossover. However, even in these cases, symmetry breaking could cause a rapid loss of diversity since members of the population in other symmetric regions are lost, with little interchange of genetic material. Thus symmetry breaking may initially have a detrimental effect even in problems with exact symmetries. The difficulties that arise for algorithms using crossover in problems with explicit symmetries are well known. The point of introducing them here is that they illustrate clearly why there can be a cost of crossover. In addition, they make it clear what we mean by symmetry breaking in a population.

However, our concern in this paper is with problems where the search space is split into many regions which are not identical, although some regions may appear similar to the search algorithm. Which region the population ends up in is often down to chance. The term 'symmetry breaking' when applied to problems with, at best, approximate symmetries might be confusing, however, we persist in doing so because of its close analogy to symmetry breaking in systems with exact symmetry. We could view the symmetry breaking in a different light by considering an ensemble of runs. Regions with a similar fitness and size will be visited approximately an equal number of times in the ensemble. However, a single run of an EA will only visit a few regions and will break this ensemble symmetry. In the next section, we argue that the splitting of the search space into separate regions is a generic property of a large class of optimisation problems.

5 Hidden 'Symmetries'

Detailed studies of some hard optimisation problems have revealed that the splitting of the search space into many regions containing locally optimal solutions occur precisely when these problems become hard—at so called phase transitions [11]. The splitting is not an exact symmetry of the system, but rather a random[‡] shattering of the search space. In an analogous way that a piece of glass might shatter, the search space can split into regions of very different size. Thus the regions are not all similar, however, there are typically a very large number of regions so that for any particular region there will be a number of roughly similar regions. One of the best studied problems of this type is the binary or Ising[§] perceptron. This is known to be an NP-hard problem [12]. We take some time to describe this system as the splitting of the search space into many regions can be easily understood.

A perceptron is a model of a simple neuron (or classifier machine) which consists of a single weight vector $\boldsymbol{w} = (w_1, w_2, \dots, w_N)$. In the Ising perceptron the weights are 'Ising variables', i.e. $w_i \in \{-1, 1\}$. We describe the Ising perceptron in relation to the normal perceptron where the weights can take any value. The task of the perceptron is to correctly classify a set of N-dimensional input vectors or patterns $\boldsymbol{x}_\alpha$, with $\alpha = 1, \dots, P$ labelling the different patterns. The patterns are to be classified into two classes. We denote the class associated with pattern $\boldsymbol{x}_\alpha$ by $c_\alpha \in \{-1, 1\}$. A pattern is correctly classified if

$$\mathrm{sgn}(\boldsymbol{x}_\alpha^T \boldsymbol{w}) = c_\alpha$$

where $\mathrm{sgn}(z)$ equals 1 if $z \geq 0$ and -1 otherwise. That is, the weight vector $\boldsymbol{w}$ must lie in the same direction as the pattern $\boldsymbol{x}_\alpha$ if $c_\alpha = 1$ or in the opposite direction if $c_\alpha = -1$. The length of the weight vector is not important to the classification. We will assume for the normal perceptron $|\boldsymbol{w}|^2 = N$ so that the set of binary weights is a subset of the continuous weights. We can view the weights of a normal perceptron to be a point on an N-dimensional hyper-sphere. The weights of the Ising perceptron live on an N-dimensional unit hyper-cube. The solution space for the normal and Ising perceptron with three weights is illustrated in figure 5.

We can view the input patterns $\boldsymbol{x}_\alpha$ as defining a dividing plane perpendicular to itself and going through the origin. The weight vector must be on one side or other of this dividing plane depending on the class c_α. We show an example of a pattern vector and its corresponding dividing plane in figure 6(a). The patterns cut the hyper-sphere into allowed and disallowed halves. Any part of the hyper-sphere which is not in a disallowed

[‡]We can imagine our problem is an instance from a class of problems (e.g. all 3-SAT problems with 100 variables). No point in the search space is privileged when we consider the ensemble of all problems. However, selecting an instance shatters the symmetry of the search space. The instances of the problem constitutes the set of elementary events in a probability space.

[§]The name comes from the Ising model for a magnet consisting of spins that can take two states.

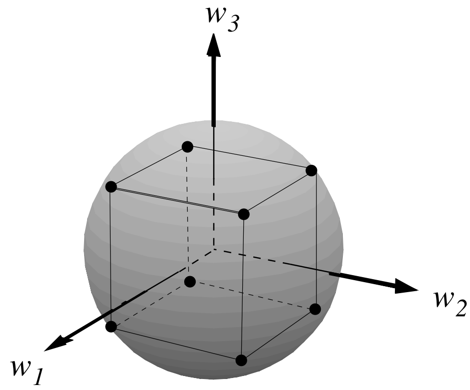


Figure 5: Weight space for the Ising perceptron (cube) and normal perceptron (sphere) for a 3-dimensional perceptron with weights $\boldsymbol{w} = (w_1, w_2, w_3)$.

half of any pattern corresponds to a perceptron which correctly classifies all the input patterns. This is illustrated in figure 6(b). The allowed region of weight space will form a convex cone on the hyper-sphere.

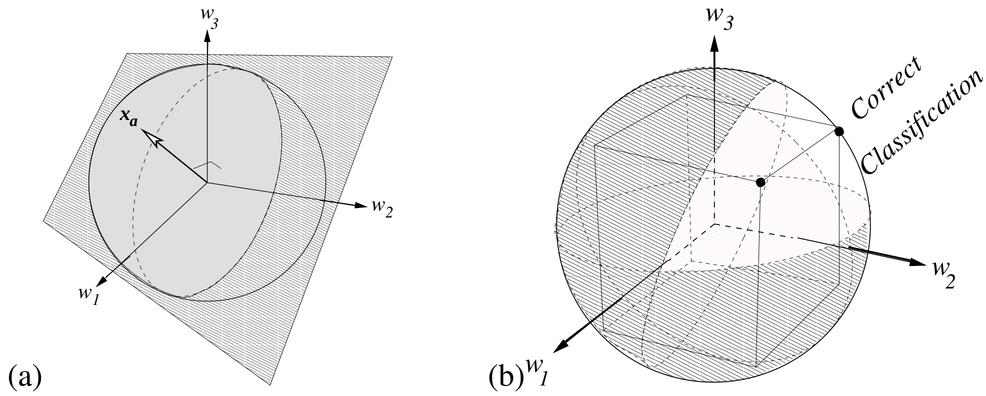


Figure 6: (a) A dividing plane defined by an input pattern $\boldsymbol{x}_\alpha$ splitting the sphere into two. If $c_\alpha = 1$ the weight vector must be on the same side of the dividing plane as the vector $\boldsymbol{x}_\alpha$. Otherwise it must be on the other side. (b) Illustration of the region in weight space that correctly classifies the patterns.

The Ising perceptrons which correctly classify the patterns correspond to those vertices on the hyper-cube which are in the allowed region of weight space. It becomes difficult to visualise the allowed region of weight space in high-dimensional space. However, we can schematically represent weight space for a normal perceptron as a plane and the weights of the Ising perceptron as a lattice. This picture under-represents the connectivity of the weight space, but it is sufficient to illustrate the shattering of weight space into regions. Figure 7(a) shows this schematic representation of weight space.

With each new pattern to be classified the allowed region of weight space is reduced. If the classification task is not linearly separable, then eventually, as we show the perceptron more and more patterns, the allowed region of weight space will vanish. When there are a few patterns almost all of the Ising perceptrons which correctly classify the patterns are connected. That is, you can move from one allowed solution to another via a path on

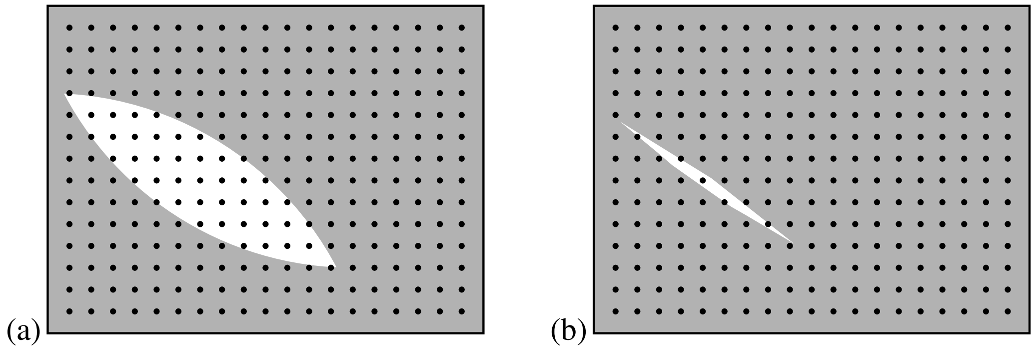


Figure 7: Schematic representation of weight space and the region of weight space that correctly classifies the patterns. (a) Case when there are relatively few patterns to classify. (b) Case when there are more patterns to classify and the solutions become disconnected.

the lattice (or along edges of the hyper-cube) that only goes through allowed solutions. However, as we increase the number of patterns we typically reach a situation where the solutions become disconnected. This causes the shattering of the search space into many regions. This is schematically illustrated in figure 7(b). The shattering occurs, despite the fact that the allowed region is convex, because the points on the hyper-cube are discrete. In high dimensional space the allowed region on the hyper-sphere will typically be broad in some directions and narrow in other directions. As a consequence the allowed points on the hyper-cube can be at a large separation from each other. When the allowed region vanishes there are disconnected regions of the hyper-sphere with a similar low number of wrongly classified patterns. Thus splitting of the search space can occur even in the normal perceptron where the search space is connected.

The shattering of the search space happens just when the problem typically becomes hard to solve [13]. This is an example of a phase transition from typically easy problems (almost all instances can be solved in polynomial time) to typically hard problems (almost all instances require exponential time). This transition coincides with what is called replica symmetry breaking which is a mathematical correlate of this phenomenon that appears in performing a mean field calculation of the thermodynamic properties of this system [14, 15, 16]. Replica symmetry breaking is a mathematical artifice associated with the typical behaviour of a landscape—it is not related to the symmetry breaking that occurs in population based algorithms.

The Ising perceptron is not alone in exhibiting this kind of behaviour. One well known problem which has recently been shown to have similar behaviour is K-SAT, an archetypal NP-complete problem [11, 17, 18, 19, 20]. K-SAT consists of a Boolean formula in conjunctive normal form, where each clause contains K variables or negated variables. For example, in 3-SAT we might have the following formula consisting of three clauses and four variables

$$f(\boldsymbol{x}) = (x_1 \vee \neg x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \neg x_4) \wedge (\neg x_1 \vee \neg x_3 \vee \neg x_4).$$

Our task is to choose the Boolean variables $x_i \in \{F, T\}$, such that the formula $f(\boldsymbol{x})$ is true. When we increase the number of clauses it will eventually become impossible to satisfy all the clauses. The task of finding the maximum number of satisfied clauses is known as MAX-K-SAT.

We can again visualise the solutions $\boldsymbol{x}$ as lying on a hyper-cube. The clauses in the formula impose constraints which restrict the solution on the hyper-cube. For example, the clause $x_1 \vee \neg x_2 \vee x_3$ requires the solution to lie on either the half cube with $x_1 = T$, or the half cube with $x_2 = F$, or the half cube with $x_3 = T$. Thus, each clause specifies an allowed region of the hyper-cube. An allowed solution is one where the allowed regions for all clauses intersect. Since the allowed regions for each clause are not convex, their intersection can consist of disconnected regions. K-SAT, for $K > 2$, was the first class of problems to be proved NP-complete. For problems with N variables and M clauses there is a characteristic ratio $\alpha_c = M/N$, below which nearly all instances are satisfiable and above which nearly all instances are unsatisfiable. In 3-SAT (in the limit $N \to \infty$), the critical ratio occurs at $\alpha_c \approx 4.267$ [19]. When M is such that $\alpha_d = 3.921 < M/N < \alpha_c$ most instances of the problem contain many local optima. In this region 3-SAT becomes typically hard. As with the Ising perceptron this transition happens exactly when the parameters defining the problem are chosen so that replica symmetry breaking occurs. The 2-SAT problem also exhibits a transition from satisfiability to unsatisfiability, but there exists a polynomial time algorithm for solving 2-SAT, so that the transition does not mark the point when the problem becomes hard to solve. However, 2-SAT does not exhibit replica symmetry breaking. Extrapolating between 2-SAT and 3-SAT, Monasson *et al.* showed empirically that instances of the problem became typically hard to solve at the same parameter values when replica symmetry breaking occurs in the mean field calculation [11, 17].

Replica symmetry breaking is known to occur in many physical models of disordered systems. The best studied such system is the spin-glass [21] (where it is known that finding the lowest energy state is NP-hard). This can be seen as a problem with random pairwise epistasis. In spin-glasses, replica symmetry breaking is associated with the splitting of state space into 'pure states'. These are similar but separate regions of state space containing local optima. The expected transition times for a thermodynamic system to move between pure states grow exponentially with the size of the system. These models suggest that a generic property for a large class of hard problems is the splitting of search space into disconnected regions containing locally optimal solutions. It is this belief that is the motivation for this paper. In figure 8, a schematic illustration of the replica symmetry breaking transition and the concomitant shattering of the solution space is shown.

Figure 8 is a caricature of what happens to the search space in real problems. It fails to represent the high dimensionality of most search spaces. The number of regions that the space splits into after replica symmetry breaking is typically exponentially large in the number of variables in the problem. At the same time, the proportion of solution space corresponding to very good solutions is an exponentially small fraction of the total number of possible solutions. Finally, the splitting up of search space can happen on different levels, with regions of good solutions clustering together into super-regions, which in turn can also form clusters, etc. The study and characterisation of landscapes is a large and developing field—see for example [22]. Nevertheless, figure 8 provides an intuitive picture of how the solution space can split as the difficulty of the problem increases.

A population searching a landscape such as that illustrated in figure 8 will be attracted towards the different fitness peaks. After an initial exploration period the population will concentrate on a single peak. This is what we are calling symmetry breaking—it breaks

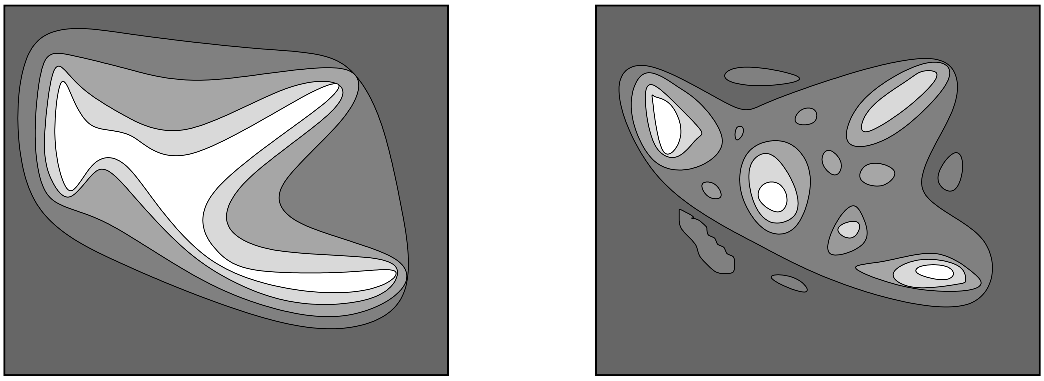


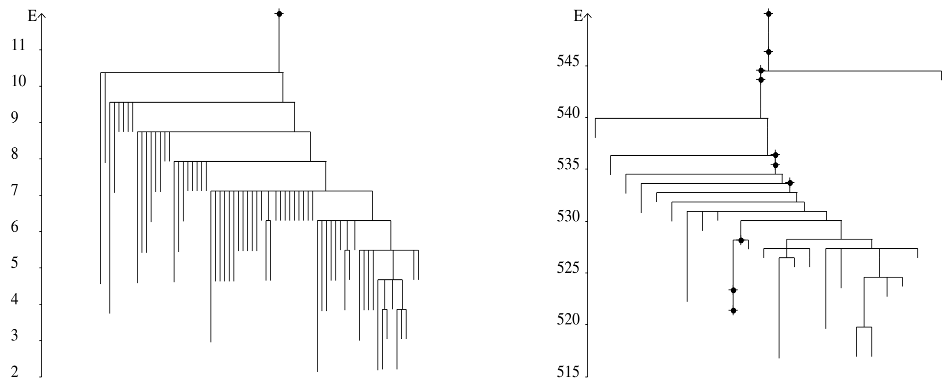
Figure 8: Schematic representation of a replica symmetry breaking transition. The lighter the search space the better the solutions. The left hand diagram represents the solution space before the replica symmetry breaking transition. The right hand diagram shows the solution space when replica symmetry breaking has occurred. The transition typically occurs when the complexity of the problem reaches some critical value.

the symmetry or approximate symmetry of the search space. It is a phenomenon that occurs in finite populations and, as we will see, it can happen very rapidly. The important issue for understanding the performance of EAs is whether the peak it concentrates on contains the best solution.

5.1 Barrier Trees

One recently developed tool for visualising complex search spaces is the barrier tree [23, 24], where the fitness landscape is represented as tree, in which the leaves of the tree represent the local optima and nodes on the tree represent the fittest saddle-points between subtrees. The nodes of the the tree are drawn as horizontal lines. To conform with the usual convention of drawing Barrier trees we consider problems where we minimise some energy. The low energy (fit) solutions lie at the bottom of the tree. The vertical position of the nodes of the tree give their fitness. In generating a barrier tree it is necessary to specify a neighbourhood relation. We have used a Hamming neighbourhood in the examples given below. Although, in principle, crossover could allow a member of the population to jump from one branch of the tree to unrelated one, this is observed to be a very rare event. Two examples of barrier trees are shown in figure 9. This shows the barrier trees for a Ising perceptron and a MAX-3-SAT (to obtain complex trees with a relatively small number of variables we have chosen the parameter controlling the complexity of the problem to be well above the hardness transition point). The barrier tree shows the shattering of the search space into regions containing local optima. Although, the local optima are not all the same—the fitter (lower energy) optima tend to have larger basins of attraction—the optima at around the same fitness are found to be roughly similar. That is, they seem to contain roughly the same number of states and have similar basins of attraction.

Recently, barrier trees have been used to visualise how optimisation algorithms solve hard optimisation problems [25]. To do this each configuration of the search space is mapped to a point on the barrier tree depending on which local optima are accessible from the configuration without going through any configuration less fit than the starting



(a) (b)

Figure 9: Barrier tree for (a) an Ising perceptron with 17 variables classifying 1000 patterns, the scale shows the number of wrongly classified patterns, and (b) a MAX-3-SAT problem with 17 variables and 4675 clauses the scale showing the number of unsatisfied clauses. The tree is truncated above its root. The small circles represent members of a population. Initially all members of the population are above the root—as illustrated in (a). Figure (b) shows the population of a GA after 70 iterations.

point. Configurations with the same fitness that can reach the same set of local minima are mapped to the same point on the tree. The vast majority of configurations lie above the root of the tree (i.e. it is possible to get to all the local minima without having to go over a barrier with energy greater than the energy of the starting point). The branch above the root has been truncated as it is just a single vertical line. Figure 9 illustrates a typical run of a GA where the small circles mark members of the population. Initially all members of the population have a fitness above the root of the tree (as we have truncated the tree, they are all mapped to the same point). When we run a genetic algorithm on these problems the members of the population descend the tree. Frequently the population will get stuck for a long time in a sub-optimal branch of the tree—an example where this happens is shown in figure 9(b). Note that although the population is concentrated around a local optimum, some members of the population will have higher energy than the saddle point. Most of these members had much fitter parents in the basin of attraction of the local optimum. The set of states at a minimum tend to lie in narrow valleys so that crossover and mutation can easily produce children much less fit than the parents. Which branch of the barrier tree the GA ends up in varies from run to run, although there is a very strong preference for the GA to get trapped in one of the low lying minima, but not necessary the global optimum. As we increase the population size we arrive at the global optimum more often.

These preliminary investigations seem to support the picture presented in this paper that the population becomes focused on one region of the search space. This can significantly delay the discovery of the global minimum. However, a systematic study of how a GA explores these problems still needs to be carried out. Unfortunately, to compute the barrier tree it is necessary to enumerate the whole search space, thus these studies are confined to rather small problems. In problems with many variables the number of local optima will typically be very much larger making the search space considerably more difficult for an optimisation algorithm to explore.

6 Crossover and Symmetries

The splitting of the solution space into different regions causes a particular difficulty for search algorithms using crossover. Crossing two solutions from different regions is likely to lead to an offspring which is far less fit than its parents. This slows down the initial gain in fitness, but much more significantly, it can cause a rapid symmetry breaking of the population. To examine the consequence of this we present some empirical simulations on models with different symmetries. We then present a mathematical analysis of symmetry breaking using stochastic differential equation techniques.

6.1 Empirical Studies

To study how symmetry breaking occurs in evolutionary algorithms we consider a simple generational genetic algorithm on some modifications of the Hurdle problem described in section 2. These models are an attempt to caricature what happens when the population reaches a split in the barrier tree. The population may initially explore both branches of the tree, but will rapidly end up with most members of the population focused around one branch. The first model we consider is the 'Symmetric Hurdle problem' which has a fitness dependent on the Hamming distance from a reference $\boldsymbol{x}^*$ given by

$$F(H) = \begin{cases} L - H - 2\left[H \text{ is odd}\right] & \text{if } H \leq L/2 \\ H - 2\left[L - H \text{ is odd}\right] & \text{if } H > L/2. \end{cases}$$

An example of the Symmetric Hurdle problem is shown in figure 10.

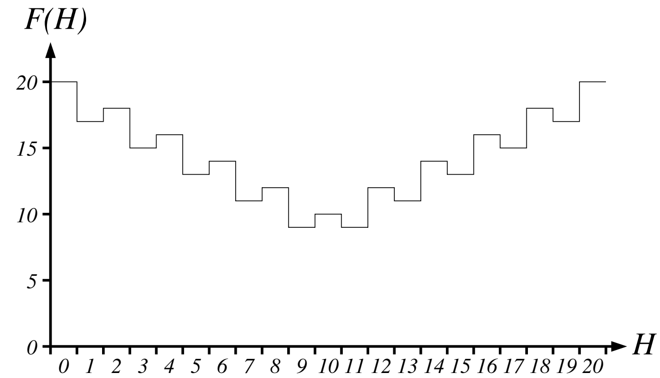


Figure 10: Fitness for the Symmetric Hurdle problem as a function of the Hamming distance from a reference string $\boldsymbol{x}^*$ for strings of length $L = 20$. Compare this with the original Hurdle problem shown in figure 1.

There are only two symmetric regions in this problem depending on whether H is greater than or less than $L/2$. Initially, the population will split into two halves, each half evolving towards one of the two global optima at either $H = 0$ or $H = L$. Eventually one sub-population becomes larger than the other, after which the symmetry is rapidly broken.

Once the symmetry is broken the population evolves in almost the same way as it would on the original Hurdle problem. This is a difficult problem for a GA without crossover because of the local roughening of the landscape.

We report results using scaled Boltzmann selection where the probability of selecting an individual, i, with fitness F_i is

$$p_i = \frac{e^{\beta F_i/\sigma}}{\sum_{j=1}^{P} e^{\beta F_j/\sigma}}$$

where σ is the standard deviation of the fitness distribution in the population. Boltzmann selection allows us to change the selection pressure by modifying β. We consider a selection rate $\beta = 1$ giving quite a strong selection pressure. (A comparison in the performance between Boltzmann, Ranking and Tournament selection is given in [26]). Figure 11 shows the observed mean first-passage time for the Hurdle problem and the Symmetric Hurdle problem.

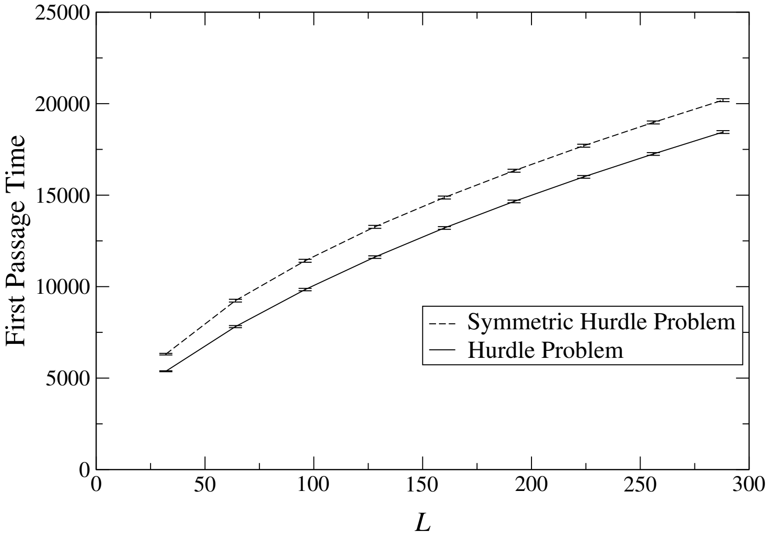


Figure 11: First passage time to (one of) the optimum string(s) in terms of the number of function evaluations for the Hurdle problem and the Symmetric Hurdle problem as a function of the string length L. The population size is $P = 1000$. A scaled Boltzmann selection with $\beta = 1$ and a mutation rate of $1/L$ is used. The results are averaged over 1000 runs.

The first passage time is observed to scale similarly for the normal and Symmetric Hurdle problem with the Symmetric Hurdle problem taking a little over 10% longer than the Hurdle problem. This does not seem to depend strongly on the population size or even the selection strategy—we found similar results for scaled proportional selection. Symmetry breaking is very rapid, it typically happens in about a tenth of the time it takes for the GA to find one of the global optima. Rapid symmetry breaking is not however always desirable, particularly in problems with only approximate symmetry. In these cases the different regions might not contain the same quality solution. Rapid symmetry breaking can prevent a proper exploration of the search space. To illustrate this, we consider a modification of the Symmetric Hurdle problem to give it a slight asymmetry, so that,

$$F(H) = \begin{cases} L - H - 2\left[H \text{ is odd}\right] & \text{if } H \leq L/2 \\ 0.1\,L + 0.8\left(H - 2\left[L - H \text{ is odd}\right]\right) & \text{if } H > L/2. \end{cases}$$

This ensures that the fitness of the state $H = L$ is ten percent less than that at the global optimum at $H = 0$. An example of this problem is shown in figure 12.

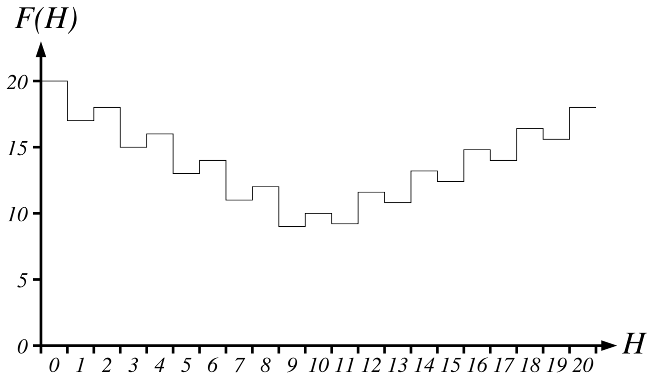


Figure 12: Fitness for an Asymmetric Hurdle problem as a function of the Hamming distance from $\boldsymbol{x}^*$ for strings of length $L = 20$.

In figure 13 we show the empirically measured probability of the population splitting to the fitter region (i.e. ending up with all members of the population with $H < L/2$) plotted against the selection strength. Although the region with $H < L/2$ has a small selection advantage this might not be enough to overcome any stochastic bias towards the states with $H > L/2$. If the sub-population with $H > L/2$ becomes sufficiently large then it will become favoured because of the cost of crossover. For very small selection strength random drift will be important so that the population might end up in either region. As we increase the selection strength the selective advantage of the region with $H < L/2$ will be preferred. However, for very strong selection, the population will have insufficient time to determine which region is fitter. Thus the probability of reaching the global optimum will drop off as we increase the selection strength beyond some optimal value. Increasing the population size reduces the genetic drift and thus slows down the symmetry breaking allowing a better estimate of the quality of the different regions. Therefore, we are more likely to reach the global optimum as we increase the population size.

The Symmetric Hurdle problem is simple because it only possesses two symmetric states. We argued earlier that in hard problems there are typically a large number of similar regions. To explore what happens in problems with a large number of regions we introduce a second problem in which the string variables can take on many different values $x_i \in \{0, 1, 2, \ldots, Q-1\}$. We denote the number of variables in a string with value k by n_k, i.e.

$$n_k = \sum_{i=1}^{L} \left[x_i = k \right].$$

A problem with Q similar regions is given by the fitness function

$$F(\boldsymbol{x}) = \sum_{k=0}^{Q-1} \frac{n_k^2}{1 + \eta k}.$$

We call this the 'Asymmetric Pott's problem' due to its similarity to Pott's models used in statistical physics. The parameter η controls the degree of asymmetry. The fitness is maximised when all the string variables, x_i, are equal to 0.

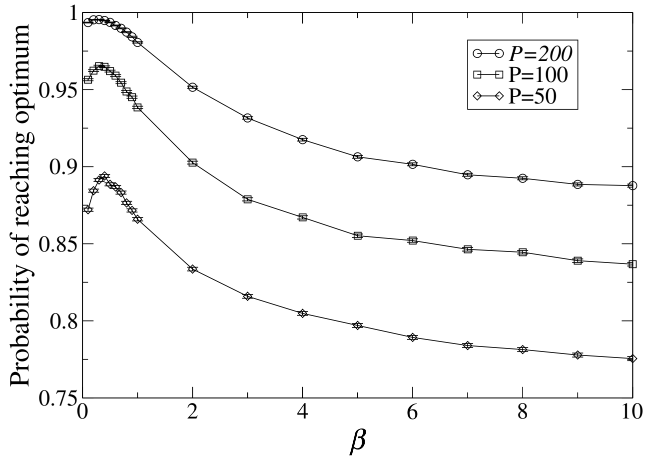


Figure 13: Probability of the population splitting to the fitter region ($H < L/2$) against the selection strength for the Asymmetric Hurdle problem, with $L = 64$. The data points show averages of $100\,000$ runs.

Figure 14 shows a histogram of the probability of ending in one of the regions A_k where all members of the population have a majority of their bit equal to k. As we increase the population size we observe that the population is more likely to end up in one of the fitter regions. The Asymmetric Pott's problem has a rich set of behaviours. However, it is much harder to characterise the performance of an EA on this problem than on those with only two regions. In the following we will restrict our attention to these simpler problems, except for a passing remark on the asymptotic survival time for the population to be in many different regions given no selective bias.

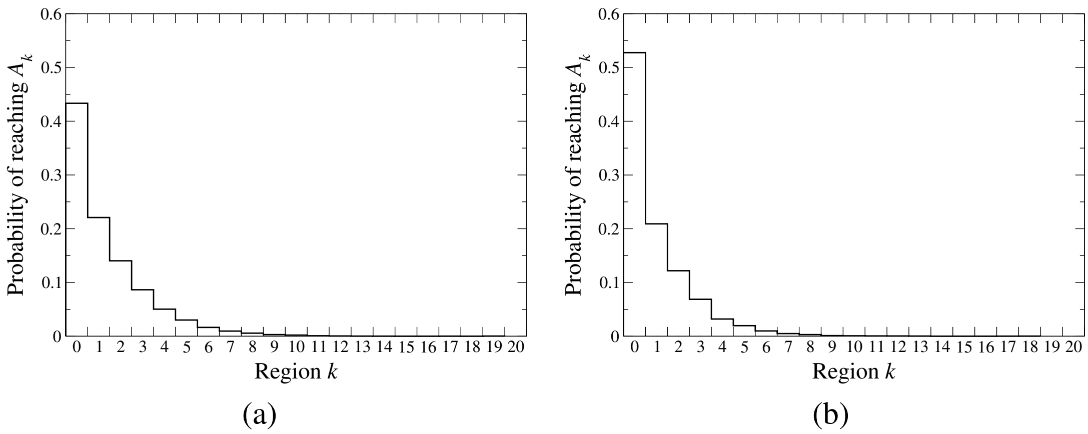


Figure 14: Histogram showing the probability of ending in state A_k for a population of size (a) 50 and (b) 100. The problem is the Asymmetry Pott's problem with $Q = 20$ and $\eta = 0.05$. The results shown are for a GA using scaled Boltzmann selection with $\beta = 1$ and a mutation rate of 1 bit per string on average. The histogram are computed from $10\,000$ runs.

6.2 Analysis of Symmetry Breaking

In this section, we present an analysis of the symmetry breaking that occurs in the Symmetric Hurdle problem or any other problem with two symmetric states. Our objective is to gain a qualitative understanding of how quantities such as the symmetry breaking time (i.e. the time for the whole population to move into a single region) and the probability of the population ending up in a particular state depend on the parameters of the GA (e.g. the population size and selection strength) rather than give an accurate model of the dynamics.

The symmetry breaking of a population into two states is analogous to the problem of fixation of alleles at a single locus. This is a classic problem in population genetics. Consider a population contains two alleles A and A'. The frequency of occurrence of the alleles is determined by selection and random genetic drift. The problem we are interested in, determining the time for a population to completely break the symmetry of the search space, is analogous to determining the fixation (or take-over) time for one of the alleles with a possible preference for the homozygous state. Fixation times have been modelled using stochastic differential equations (diffusion models) since the 1920's [27, 28, 29]. The great modern exponent of this type of analysis was Motoo Kimura. Many of the results in this section are taken, with minor modifications, from Kimura's work. The articles used can all be found in a selected collection of his works [30]. We will make the same assumptions that are usually made in the population genetics literature (see, for example, [31, 32]).

Symmetry breaking through random drift

We begin by considering the simplest case of symmetry breaking. That is when there is no selective pressure causing symmetry breaking so that it is produced through random drift alone. This would be the case in an asexual population, although with no crossover symmetry breaking is not important to the dynamics.

We consider a population of P individuals split between two regions, A and A'. We denote the number of individuals in region A by P_A. The proportion of the population in region A we denote by $X = P_A/P$. Our goal is to model how X varies with time. Since X is a stochastic variable we consider the probability mass function $\mathbb{P}\big(X(t) = x\big)$, where $X(t)$ denotes the proportion of individuals in region A at generation t. We assume all members of the population are equally fit so that the fluctuations in the proportion of the population in the two regions are caused by random sampling. The probability of there being $P'_A = PX(t+1)$ individuals in region A at the next generation is given by the binomial distribution

$$\mathbb{P}\big(P'_A\big) = \binom{P}{P'_A} X(t)^{P'_A} \, \big(1 - X(t)\big)^{P-P'_A}$$

since the probability of choosing a member of the population in region A is proportional to $X(t)$. On average, therefore, $X(t+1) = P'_A/P = X(t)$—i.e. the number of individuals in region A remains unchanged on average. However, $X(t+1)$ is a stochastic variable with variance $X(t)(1 - X(t))/P$. In this model we have made some simplifying assumptions, for example, we have neglected the fitnesses of the individuals. Including these effects would introduce additional sources of fluctuations. Many of these effects can be compensated for by replacing the true population size with an 'effective population size'—we discuss this in appendix A. Symmetry breaking in this model is caused by 'random genetic drift' due to stochastic sampling. (The term genetic drift is unfortunate in the context of diffusion models, where the term drift usually refers to the non-random part of the dynamics, however, the term is sufficiently established that it is preferable to live with the possible confusion rather than use a non-standard nomenclature.) The dynamics of $X(t)$ is described exactly by a discrete Markov chain. However, this does not give a closed form solution so provides little insight into what is happening. Instead we approximate the dynamics by a stochastic differential equation. We assume that the change in $X(t)$ is given by

$$\mathrm{d}X(t) = \sqrt{\frac{X(t)\,(1 - X(t))}{P}}\,\mathrm{d}W(t) \tag{1}$$

where $\mathrm{d}W(t)$ is an infinitesimal Gaussian noise with mean zero and variance $\mathrm{d}t$.

There are, in fact, two simplifying assumptions being made in equation (1). The first is to approximate a Binomial process by a Wiener or Gaussian process with the same mean and variance. This has the effect of allowing $X(t)$ to exist on a continuous set of points rather than a discrete set. This is usually a fairly innocuous approximation. The second approximation is to replace finite jumps $\Delta X(t)$, by infinitesimal jumps and at the same time take the continuous time limit. This is a reasonably good approximation, provided the variance in the noise is sufficiently small—this is fortunately the case in the problem we are considering. Technically, equation (1) is to be interpreted in the sense of Itô [33].

We can obtain a solution for the probability, $\mathbb{P}\big(x,t|x',t'\big)$, that $X(t) = x$ at time t given that $X(t') = x'$ at an earlier time t' in terms of the Kolmogorov equations. To calculate the time taken until all member of the population are in state 1 we consider the backward Kolmogorov equation for $u_1(x,t) = \mathbb{P}\big(1,0|x,-t\big)$ (i.e. the probability that having ended up in state $X = 1$ at time $t = 0$ the system was in state $X = x$ at time $-t$) given by

$$\frac{\partial u_1(x,t)}{\partial t} = \frac{x(1-x)}{2P}\frac{\partial^2 u_1(x,t)}{\partial x^2} \tag{2}$$

with boundary conditions $u_1(0,t) = 0$, and $u_1(1,t) = 1$. Using separation of variables this reduces to a Sturm-Liouville eigenvalue problem. This was solved by Kimura [34], giving

$$u_1(x,t) = x + \sum_{i=1}^{\infty} \frac{(-1)^i}{2}\big(\mathrm{P}_{i-1}(1-2x) - \mathrm{P}_{i+1}(1-2x)\big)\,\mathrm{e}^{-i(i+1)t/(2P)}$$

where $\mathrm{P}_i(x)$ are the Legendre polynomials. The distribution of times for the population to end up in the state $X(t) = 0$, starting from a state $X(0) = x$, is equal, by symmetry, to $u_0(x,t) = u_1(1-x,t)$. To model the symmetry breaking that occurs in the Symmetric Hurdle problem we assume the initial condition $X(0) = 1/2$ (more accurately we should average over $x = P_A/P$ where P_A is binomial distributed with mean $1/2$, however, we ignore this elaboration, as it leads to a small correction only). The probability that the population has completely split into either region A or region A' at time t is given by

$u(t) = u_1(1/2, t) + u_0(1/2, t)$, which reduces to

$$
\begin{aligned}
u(t) &= 1 - \sum_{i=0}^{\infty} \frac{(-1)^i\,(2i-1)!!\,(4i+3)}{2^{i+1}\,(i+1)!} \mathrm{e}^{-(2i+1)(i+1)t/P} \\
&= 1 - \frac{3}{2}\mathrm{e}^{-t/P} + \frac{7}{8}\mathrm{e}^{-6t/P} - \cdots .
\end{aligned}
$$

The higher order terms in $u(t)$ drop off very rapidly with t, so for $t > P$, $u(t)$ is well approximated by $1 - 3\mathrm{e}^{-t/P}/2$. The probability density for the symmetry breaking time, $\mathbb{P}\big(t_{SB}\big) = \mathrm{d}u(t)/\,\mathrm{d}t$, computed from the formula given above is shown in figure 15. Also shown is the simulated result for the model with a population of 100. It is apparent from figure 15 that, in this case, the diffusion approximation gives a very accurate estimate of the behaviour of the model for a finite population.

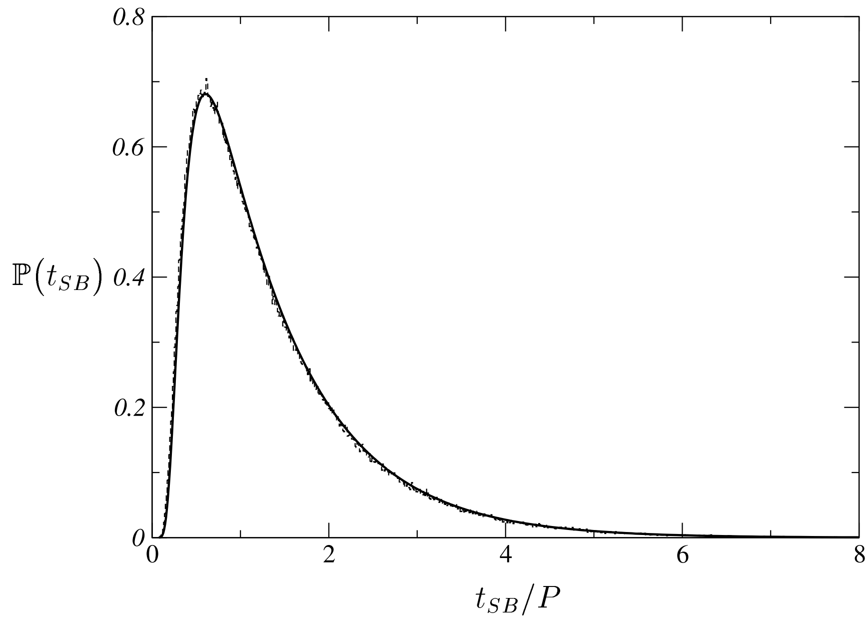


Figure 15: Probability density for the time taken for a population to break the symmetry of the solution space under random sampling. The solid line shows the theoretical results while the dotted line shows the histogram computed from 1 000 000 runs for a population of size 100, where t_{SB} is the number of generations for the whole population to be in either state A or A'.

The average time for the population to break the symmetry of the search space is $P\log(4) \approx 1.39P$ generations. This model is a reasonable estimate of the time it would take for a GA without crossover to break the symmetry in the Symmetric Hurdle problem. The loss of a member of the population from a region will be even more rapid in systems with many symmetric regions. For example, if we have Q symmetric regions, the probability that there is at least one member in k of the regions falls off asymptotically as $\exp\big(-k(k-1)t/(2P)\big)$ (see [34]). That is, in systems with a high degree of symmetry (such as the Asymmetric Pott's problem with $\eta = 0$), the population will rapidly concentrate on only a few regions due to sampling alone.

Some of the results in this section can also be obtained by considering the evolution as a coalescing random walk [35]. In this view we consider the relationship between individuals as we look backwards in time. Individuals that have a common ancestor some time in the past are guaranteed to be in the same region. Those that do not have a common ancestor might still be in the same region if their ancestors in the initial population started in the same region. The probability that two members of the population do not have a common ancestor within t previous generations is approximately given by $P\exp(-t/P)$. As a consequence, the expected time (in the past) for a population of size P to have just k distinct ancestors is $2P/k \pm O(1)$. For a population that can be in Q symmetric regions, this gives a good approximation to the expected time for the population to be in k regions provided $k \ll Q$ (if $k \approx Q$ then we have to take into account the possibility that some of the k ancestors started in the same region). For large, Q, the expected symmetry breaking time is thus $2P \pm O(1)$, while the expected time for all the members of the population to have two common ancestors is $P \pm O(1)$. Thus, the initial symmetry breaking is very rapid. On average, the population will spend half of its time in only two regions before it breaks the full symmetry.

Crossover induced symmetry breaking

For a GA with crossover there is a selective advantage to members in the region of search space containing the most number of members, since they will be more likely to cross with members in the same region of solution space. This selective advantage drives the symmetry breaking much more rapidly than in random drift. To obtain an intuition for the speed of this symmetry breaking we assume that the population is randomly paired so that on average $P\,X^2$ offspring have both parents from region A and $2\,P\,X(1-X)$ offspring have mixed parents. We assume that half of the offspring with mixed parents end up in region A. Furthermore, we suppose that there is a small selective advantage of $(1+\epsilon)$ to those offspring that have parents from the same region. In this model, there will be four groups of individuals after crossover: on average $P\,X^2$ individuals in region A whose parents both came from region A; $P\,X(1-X)$ in region A with parents from A and A'; $P\,X(1-X)$ in region A' with parents from A and A'; and $P\,(1-X)^2$ individuals in region A' with both parent from region A'. After selection the first and last groups get increased by a factor $(1+\epsilon)$ relative to the second and third groups. Thus, the average size of the population after crossover and selection is

$$\begin{aligned}
\mathbb{E}\left(X'\right) &= \frac{(1+\epsilon)X^2 + X(1-X)}{(1+\epsilon)(X^2+(1-X)^2) + 2X(1-X)} \\
&= \frac{X+\epsilon\,X^2}{1+\epsilon(1-2X(1-X))}
\end{aligned}$$

where the denominator is a normalisation term. The average change in the proportion of individuals in region A (i.e. those from the first and second group) is

$$\begin{aligned}
\mathbb{E}\left(\Delta X\right) &= \mathbb{E}\left(X'\right) - X = \frac{\epsilon X(1-X)(2X-1)}{1+\epsilon(1-2X(1-X))} \\
&\approx \epsilon X(1-X)(2X-1) - O(\epsilon^2)
\end{aligned}$$

In the Symmetric Hurdle problem ϵ will change in time as the population becomes fitter. How this happens will depend on the type of selection used. We avoid these complications and assume that ϵ remains fixed (or that ϵ denotes the average selection advantage of the offspring from parents inhabiting the same region). Crossover will produce additional fluctuations due to the stochastic process of pairing the individuals. The main contribution of these fluctuations would be to reduce the "effective population size" by a factor of two (see appendix A for a discussion of the effective population size, P_{eff}). We can model the symmetry breaking process, using the approximation scheme described above, by the stochastic differential equation

$$\mathrm{d}X(t) = \epsilon\, X(t)(1-X(t))(2X(t)-1)\,\mathrm{d}t + \sqrt{\frac{X(t)\,(1-X(t))}{P_{\text{eff}}}}\,\mathrm{d}W(t). \tag{3}$$

The corresponding backwards Kolmogorov equation is

$$\frac{\partial u(x,t)}{\partial t} = \epsilon\, x(1-x)(2x-1)\frac{\partial u(x,t)}{\partial x} + \frac{x(1-x)}{2\,P_{\text{eff}}}\frac{\partial^2 u(x,t)}{\partial x^2} \tag{4}$$

where we assume that $X(0)$ is either 0 or 1. Consequently the boundary condition on $u(x,t)$ are $u(0,t) = u(1,t) = 1$. There is no solution for $u(x,t)$ in terms of known functions. However, we can obtain a solution for the average time until symmetry breaking. To calculate this we use a formula derived by Kimura and Ohta [36]. The average time until symmetry is broken, assuming that initially $X(0) = x$, is given by

$$T(x) = \int_0^\infty t\,\frac{\partial u(x,t)}{\partial t}\,\mathrm{d}t.$$

Differentiating equation (4), multiplying by t, and integrating, we obtain

$$\int_0^\infty t\frac{\partial^2 u(x,t)}{\partial t^2}\,\mathrm{d}t = \epsilon\, x(1-x)(2x-1)\frac{\mathrm{d}T(x)}{\mathrm{d}x} + \frac{x(1-x)}{2\,P_{\text{eff}}}\frac{\mathrm{d}^2T(x)}{\mathrm{d}x^2}.$$

The left hand side can be evaluated using integration by parts

$$\begin{aligned}\int_0^\infty t\frac{\partial^2 u(x,t)}{\partial t^2}\,\mathrm{d}t &= \left[t\frac{\partial u(x,t)}{\partial t}\right]_0^\infty - \int_0^\infty \frac{\partial u(x,t)}{\partial t}\,\mathrm{d}t \\ &= -\big(u(x,\infty) - u(x,0)\big).\end{aligned}$$

Assuming $x \neq 0$ or 1 then $u(x,0) = 0$. Given sufficient time we expect that we will eventually reach either $X = 0$ or $X = 1$ so $u(x,\infty) = 1$. Thus the average symmetry breaking time starting from a population with $P_A = x\,P$ is given by

$$\epsilon\, x(1-x)(2x-1)\frac{\mathrm{d}T(x)}{\mathrm{d}x} + \frac{x(1-x)}{2\,P_{\text{eff}}}\frac{\mathrm{d}^2T(x)}{\mathrm{d}x^2} = -1$$

subject to the boundary conditions $T(0) = T(1) = 0$. This equation can be solved directly be integration giving

$$T(x) = 2\,P_{\text{eff}}\int_x^1 \mathrm{e}^{2\epsilon\,P_{\text{eff}}\,\eta(1-\eta)}\left(c + \int_{1/2}^{\eta}\frac{\mathrm{e}^{-2\epsilon\,P_{\text{eff}}\,\xi(1-\xi)}}{\xi(1-\xi)}\,\mathrm{d}\xi\right)\mathrm{d}\eta \tag{5}$$

where c is chosen so that $T(0) = 0$. (Although the integrals are convergent the integrands diverge at the end of the integration domain. To compute the integrals numerically it is helpful to remove these divergences as described in appendix B.)

The time for the system to break the symmetry, given that it started with $X(0) = 1/2$, is given by $T(1/2)$. In figure 16, we show the average symmetry breaking time as a function of the selective advantage multiplied by the effective population size. Also shown are simulation results for a population undergoing selection and crossover. Each member of the population is either in state A or state A'. Those in state A have a selective advantage of $1 + \epsilon$ over those in state A'. Under crossover a child from parents in the same state remain in the same state as the parents, while children with mixed parents have an equal chance of being in either state. For this system the effective population size is $P_{\text{eff}} = P/2$. We show simulations results for a population of size $P = 1000$. For large ϵ there is a small discrepancy between the theory and simulation results that occurs because we ignored terms of order ϵ^2. The error is more pronounced for small populations, so that with a population of $P = 100$ the simulation results are nearly 50% larger than the theory when $\epsilon P_{\text{eff}} = 50$ (i.e. when $\epsilon = 1$).

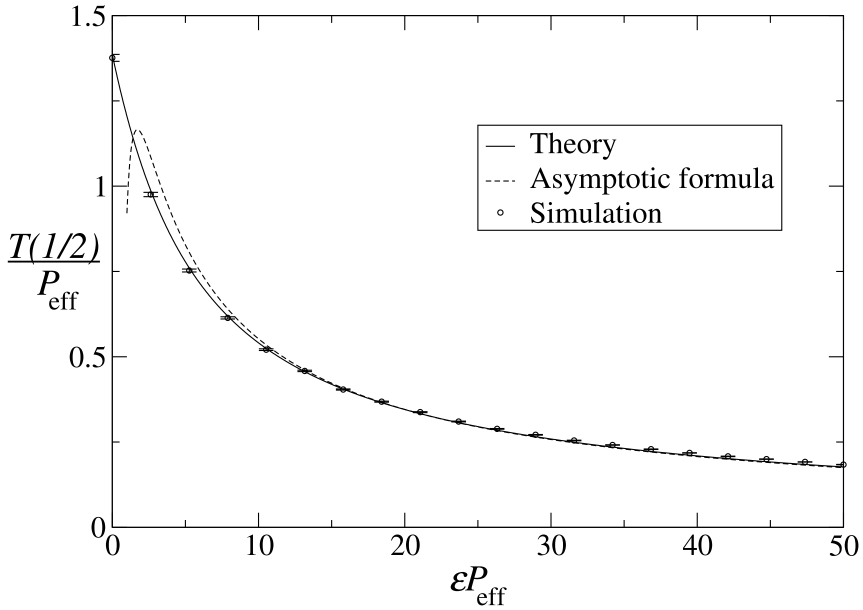


Figure 16: Average time for a population to break the symmetry. The theoretical results are obtained by numerically integrating equation (5). The asymptotic formula is derived in appendix C. Simulation results are for population of size 1000, averaged over 10 000 runs. Error bars show the expected error in the means.

Note that abscissa in figure 16 is a function of ϵP_{eff} so that even a small selective advantage, ϵ, will produce a large speed up in symmetry breaking when the effective population size, P_{eff}, is large. In appendix C, we derive an asymptotic formula for the symmetry breaking time when $\epsilon\, P_{\text{eff}}$ becomes large. The symmetry breaking time is approximately given by $(2\log(\epsilon\, P_{\text{eff}}) + 3(1 - \log(2)))/\epsilon$. If there is a significant selective advantage to offspring whose parents come from the same region then symmetry breaking can occur in only a few generations.

Asymmetric Regions

For understanding the performance of an EA on hard optimisation problems a more interesting situation is when two regions, A and A', of search space have slightly different fitnesses. This is the case for the Asymmetric Hurdle problem shown in figure 12. There are now two selective biases influencing symmetry breaking: the selective bias due to the cost of crossover and favouring the largest population; and a selective bias caused by the difference in fitness between the two regions.

The average proportion of the population in state A after crossover and selection is now given by

$$\mathbb{E}\big(X'\big) = \frac{(1+s)\,\big((1+\epsilon)X^2 + X(1-X)\big)}{(1+s)(1+\epsilon)X^2 + (2+s)X(1-X) + (1+\epsilon)(1-X)^2}.$$

The average change in the proportion of the population in state A after crossover and selection is

$$\begin{aligned}\mathbb{E}\big(\Delta X\big) &= \frac{s\,X(1-X) + \epsilon\,X(1-X)(2X-1) + s\,\epsilon\,X^2(1-X)}{1 + s\,X + \epsilon(1-2X+2X^2) + s\,\epsilon\,X^2} \\ &\approx \epsilon X(1-X)(2X-1) + s\,X(1-X).\end{aligned}$$

where the final approximation is valid provided $s, \epsilon \ll 1$.

Both types of selective advantage depend on the selection strength. To encode this dependence we write $s = r\epsilon$, where r is the ratio of the selective advantage due to the difference in fitness of the two regions to the selective advantage of the bigger sub-population caused by the cost of crossover. The ratio, r, will be determined, in part, by the problem and, in part, by how we evolve the population. For example, reducing the crossover probability or imposing a mating taboo between dissimilar individuals would increase r. We expect that the larger r the more likely we are to finish up in the fitter region.

We can model symmetry breaking in a problem with asymmetric regions by the stochastic differential equation

$$\mathrm{d}X(t) = \epsilon\,X(t)\,\big(1-X(t)\big)\,\big(2X(t)-1+r\big)\,\mathrm{d}t + \sqrt{\frac{X(t)\,(1-X(t))}{P}}\,\mathrm{d}W(t) \tag{6}$$

(again we should replace P by P_{eff} to model the additional fluctuations that arise from crossover—however, to prevent the equations becoming too difficult to read we will just write P). This is the same as equation (3) except it has an additional term $\epsilon\,r\,X(t)\,(1-X(t))\,\mathrm{d}t$, which introduces a selective advantage to members of the population in region A. Equation (6) is a powerful abstraction of the symmetry breaking problem in landscapes with two regions with dissimilar fitnesses. All the details of the EA are encoded into three parameters; a selection strength ϵ, the selective ratio r, and the (effective) population size P. Again we can borrow from the work of Kimura to solve this problem [34, 36]. We are particularly interested in the probability that the population ends up entirely in region A. This is rather easy to obtain from the backwards Kolmogorov equation. We choose as our boundary condition that $X = 1$ at time $t = 0$ and let $u(x,t)$ be the probability that $X = x$ at time $-t$. The quantity $u(x) = u(x,\infty)$ is the probability that the population finishes completely in region A (i.e. $X(t) = 1$ for some t) given that it started with a proportion x in region A. To obtain an expression for $u(x)$ we note that

$$\int_0^\infty \frac{\partial u(x,t)}{\partial t}\,\mathrm{d}t = u(x,\infty) - u(x,0) = u(x)$$

since $u(x,0) = 0$ (provided $x \neq 0$). Differentiating and integrating the backward diffusion equation with respect to t we obtain an equation for $u(x)$

$$\epsilon\,\big(x(1-x)\,\big(2x-1+r\big)\big)\,\frac{\mathrm{d}u(x)}{\mathrm{d}x} + \frac{x(1-x)}{2P}\frac{\mathrm{d}^2u(x)}{\mathrm{d}x^2} = 0$$

subject to the boundary condition that $u(0) = 0$ and $u(1) = 1$. We can solve this directly by integration giving

$$u(x) = \frac{\int_0^x \mathrm{e}^{2P\,\epsilon\left(\xi(1-\xi)-r\xi\right)}\,\mathrm{d}\xi}{\int_0^1 \mathrm{e}^{2P\,\epsilon\left(\xi(1-\xi)-r\xi\right)}\,\mathrm{d}\xi} = \frac{\operatorname{erf}\Big(\sqrt{\frac{P\,\epsilon}{2}}(2x+r-1)\Big) + \operatorname{erf}\Big(\sqrt{\frac{P\,\epsilon}{2}}(1-r)\Big)}{\operatorname{erf}\Big(\sqrt{\frac{P\,\epsilon}{2}}(1+r))\Big) + \operatorname{erf}\Big(\sqrt{\frac{P\,\epsilon}{2}}(1-r)\Big)}$$

where $\operatorname{erf}(x)$ is the error function given by

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x \mathrm{e}^{-t^2}\,\mathrm{d}t.$$

For small, x, the error function can be approximated using the first few terms of the Taylor expansion $\operatorname{erf}(x) = (2/\sqrt{\pi})(x - x^2/(3\cdot 1!) + x^5/(5\cdot 2!) - \cdots)$.

We might expect that the probability of ending up in region A depends strongly on the initial bias in the population. For a random initial population, the number of individuals starting in region A is binomial distributed with mean $P/2$ and variance $P/4$. For large populations the distribution for the initial proportion of individuals in region A can be approximated by a Gaussian distribution with mean $1/2$ and variance $1/(4P)$. Averaging over this distribution we find the probability of the population ending up region A is

$$\begin{aligned}\mathbb{P}\big(X(\infty)=1\big) &\approx \int_{-\infty}^{\infty} u(x)\mathrm{e}^{-2\,P(x-1/2)^2}\frac{\mathrm{d}x}{\sqrt{\pi/(2P)}}\\ &= \frac{\operatorname{erf}\Big(\sqrt{\frac{P\,\epsilon}{2}}\frac{r}{\sqrt{1+\epsilon}}\Big) + \operatorname{erf}\Big(\sqrt{\frac{P\,\epsilon}{2}}(1-r)\Big)}{\operatorname{erf}\Big(\sqrt{\frac{P\,\epsilon}{2}}(1+r))\Big) + \operatorname{erf}\Big(\sqrt{\frac{P\,\epsilon}{2}}(1-r)\Big)}.\end{aligned}$$

Had we just replaced x by $1/2$ in our expression for $u(x)$ we would have obtained a similar expression to the above, but with r in place of $r/\sqrt{1+\epsilon}$ in the first error function. For most parameter settings, this correction makes very little difference, showing that the initial bias in a random population typically has little influence on the final outcome.

In the limit of very small selection advantage (so that $P\,\epsilon \ll 1$), we can expand the error functions giving $\mathbb{P}\big(X(\infty)=1\big) \approx 1/2$. This is to be expected since, in this regime, symmetry breaking is driven by the random drift. If, as seems more likely, $P\,\epsilon \gg 1$ (and $r < 1$) then

$$\mathbb{P}\big(X(\infty) = 1\big) \approx \frac{\operatorname{erf}\left(\sqrt{\frac{P\,\epsilon}{2}}\frac{r}{\sqrt{1+\epsilon}}\right) + 1}{2}.$$

If $P\,\epsilon\,r \ll 1$ then again $\mathbb{P}\big(X(\infty) = 1\big) \approx 1/2$ while if $P\,\epsilon\,r \gg 1$ (i.e. we have a large population size, selection rate, or fitness bias towards one region) $\mathbb{P}\big(X(\infty) = 1\big) \approx 1$.

The ratio r of the selection advantage of region A to the selection advantage of the larger population is determined by many factors. If the selection strength is very large then symmetry breaking occurs before the population has explored the landscape so that there will only be a very small selection advantage to those individuals in region A. This temporal component of the selection pressure has not been modelled at all in this diffusion equation approach. We can decrease the cost of crossover (and hence increase r), by reducing the crossover probability (which we have so far taken to be 1) or biasing the crossover so that it contains more genetic material from one parent than the other.

7 Conclusions

We have argued that a generic feature of a large class of optimisation problems is the splitting of the search space into regions containing local optima. This can cause problems for search algorithms using crossover. Crossing strings from different regions can cause so much disruption that the offspring of parents from different regions are not viable. This introduces a strong selection advantage to the largest population, which accelerates symmetry breaking and causes the population to become focused around one region of the search space. Rapid symmetry breaking prevents a thorough exploration of the fitness landscape, which makes it more likely for the population to find a sub-optimal solution.

We have explored some simple models with symmetry, which illustrates how finding the global optimum can be sensitive to the choice of parameters. Increasing the population size, decreasing the crossover probability, biasing the crossover operator can also slow down symmetry breaking and thus increase the likelihood of an EA finding the global optimum. However, these measures also tend to slow down the rate of exploitation. To gain a better qualitative understanding we have modelled symmetry breaking using stochastic differential equations and their associated diffusion equations. Although the diffusion equations are abstracted from real EAs they address precisely the issues of the symmetry breaking time and the probability of finishing in a fitter region which are relevant to optimisation. The behaviour of the system is succinctly captured by very few parameters. Unfortunately, there is not a one-to-one correspondence between the parameters of an EA and the parameters of the stochastic differential equations. For example, changing the selection strength of an EA will affect the selective advantage of the fitter region and the cost of crossing individuals from different regions of the search space. However, it will also change the effective population size, and can indirectly affect the ratio of the selection advantage of a fitter region to the cost of crossover, by speeding up symmetry breaking and thus giving the population less time to explore the different regions. Additionally, we have neglected higher order corrections in our diffusion equations which would change the quantitative behaviour of the solutions. Nevertheless, the diffusion equations capture the most important features driving symmetry breaking. Although, caution is needed in interpreting their consequences, they can provide insights on a wide range of problems and for many different evolutionary algorithms. Their ubiquity, in the slightly different setting of classical population genetics, is a testimony to their potential utility.

There are other strategies for mitigating the cost of crossover between solutions in different regions of parameter space. Two popular strategies are niching and demes. Niching can be implemented by preventing or suppressing the crossover of solutions that are dissimilar to each other. It is also necessary to prevent a large sub-population from creating a disproportionate number of offspring. With niching the population can explore the search space without having to pay a high price for crossing solutions from different regions. The aim is to create 'species' that can explore different parts of the solution space. Those species that are exploring less fertile regions will have a selective disadvantage and will eventually become extinct. Although this is an intuitively appealing strategy, it requires careful choice of parameters to ensure that speciation takes place. This is difficult to achieve because we usually do not know much about how the search space has been shattered. Furthermore, in hard problems, there is likely to be a hierarchy of regions. That is, regions within regions. As the evolution progresses we require symmetry breaking to occur at different scales. This might require the parameters controlling niching to vary over time.

A second popular technique which mitigates the cost of crossover is to split the population into sub-populations or demes. The demes evolve independently except for an occasional migration of individuals between demes. Because the demes are small populations they will quickly break the symmetry in the solution space. Thus crossover within demes is relatively benign. However, because there are many demes, different regions of solution space will be explored. Migration causes a selection pressure between demes so that demes exploring unproductive parts of search space will eventually be replaced by individuals from demes in better parts of the search space. Thus symmetry breaking will be much slower than would occur in an unstructured population increasing the likelihood that the population ends up in a region of search space with good quality solutions. The argument about the virtue of demes in promoting evolution has a long history. In his classic 1933 paper, Sewall Wright, using his knowledge of breeding and particular applying his study of Shorthorn bulls, proposed that demes could dramatically accelerate evolution [29]. Fisher never accepted this argument believing that small sub-populations could not maintain a high fitness [37]. This is a viewed shared by many mathematical biologists today [38]. The idea developed in this paper, that rapid symmetry breaking can limit exploration, can, perhaps, be viewed as a restatement of Wright's Shifting Balance theory of evolution in a slightly different language. The empirical evidence that deme GAs are superior to normal GAs is also controversial. There are some strong advocates claiming deme GAs are superior [39], however, preliminary trials with deme GAs showed no clear advantage on MAX-3-SAT and the Ising perceptron once selection pressure has been controlled for.

This brings us back to the statement made in the introduction that symmetry breaking provides a new perspective for understanding the performance of evolutionary algorithms. Much of the argument we have just made uses common notions about EAs, such as exploration versus exploitation, diversity, etc. However, thinking in terms of symmetry breaking allows us to refine some of these ideas and may in the long term provide new methods for improving the performance of EAs on hard optimisation problems.

Acknowledgements

I would like to acknowledge the help of Gen Zhang who motivated the author to write this paper and who performed some of the preliminary simulations. I would also like to thank Will Benfold for performing the simulations involving deme GAs discussed in the conclusion and Jonathan Hallam for providing the barrier tree graphs. Finally I would like to thank the Portuguese FCT for funding the Maths Encounters in Madeira, where some of this work was carried out.

Appendix

A Other Fluctuations and the Effective Population Size

In deriving the random drift term we considered only drift due to sample fluctuations. For example, we assumed that all members of the population have the same fitness. When members of the population have different fitnesses, then the sample fluctuations will be greater. Similarly, crossover will produce additional fluctuation because of the stochastic nature of pairing members of the population to produce children. These fluctuations can often be compensated for by replacing the real population size with a smaller "effective population size". More precisely, when computing the variance in the proportion of the population after crossover and selection we often get a number of terms proportional to $X(t)(1 - X(t))$. If the coefficient of these terms are c_i then the effective population size is given by

$$P_{\text{eff}} = \frac{1}{\sum_i 1/c_i}.$$

To illustrate how different fitnesses might change the population size we consider the case when the fitnesses are Gamma distributed. That is, we assume the fitness, f_i, of member i is drawn from a probability distribution

$$p(f_i) = \frac{f_i^{\alpha-1}\,\mathrm{e}^{-f_i}}{\Gamma(\alpha)}$$

where α encodes the degree of variation in the fitness. The mean and variance of the Gamma distribution are both equal to α, so that α is equal to the mean fitness squared divided by the variance. Thus as $\alpha \to \infty$ all the fitnesses become effectively equal, while as $\alpha \to 0$ the fitnesses vary dramatically. We assume the probability of selecting an individual i is equal to $p_i = f_i / \sum_{j=1}^{P} f_j$ (note that f_i is the biological fitness, i.e. proportional to the expected number of offspring, rather than the value of the objective function). The probabilities of selection, p_i, are distributed according to a Dirichlet distribution. If we assuming independent sampling then the probability that we choose individual, i, a total of n_i-times is given by the multinomial distribution

$$\mathbb{P}\big(\{n_i\}\big) = P! \prod_{i=1}^{P} \frac{p_i^{n_i}}{n_i!} \left[\sum_{j=1}^{P} n_j = P \right].$$

Averaging over p_i we obtain a Dirichlet-multinomial distribution for n_i. It is a straightforward, but tedious, exercise to calculate the variance in $X(t+1)$ given that there was a proportion $X(t)$ in region A at time t. We find

$$\mathbb{E}\left(X^2(t+1)\right) - \mathbb{E}\left(X(t+1)\right)^2 = \frac{X(t)\left(1-X(t)\right)}{P_{\text{eff}}} \tag{7}$$

where

$$P_{\text{eff}} = \frac{P\alpha + 1}{\alpha + 1} \tag{8}$$

is the effective population size. If $\alpha \to \infty$ then $P_{\text{eff}} = P$, which is what we expect when all members of the population are selected with equal probability. If $\alpha \to 0$ then $P_{\text{eff}} = 1$ so that the dynamics is determined by the fittest individual in the population.

In deriving equation (8), we assumed that the fitnesses are random and distributed according to a Gamma distribution. Neither assumption holds in real EAs. The effective population size will depend on the details of the selection and on the problem. For a given system we can probably obtain a more accurate model for the fluctuations which would give a different equation for the effective populations size.

Crossover will also introduce fluctuations which can be compensated for by changing the effective population size. As an example we consider a system where the populations can be in one of two regions A or A'. We denote the proportion of the population in region A by X. Under crossover we select two parents and create a child. This is repeated P times where the choice of parents is independent. We assume that if both parents come from the same region the offspring will be in the same region. If the offspring has parents from different regions there is an equal chance of the offspring ending up in either of the two regions. We can write an indicator function for the i^{th} offspring to be in region A as

$$\left[i \in A\right] = \left[\alpha_i \in A\right]\left[\beta_i \in A\right] + \zeta_i\left(\left[\alpha_i \in A\right]\left[\beta_i \in A'\right] + \left[\alpha_i \in A'\right]\left[\beta_i \in A\right]\right)$$

where α_i and β_i are the parents of offspring i and ζ_i is a Bernoulli variable with probability of one half of being 1. Averaging over all possible parents and both possible outcomes for ζ_i,

$$\mathbb{E}\left(\left[i \in A\right]\right) = X^2 + \frac{1}{2}\left(X(1-X) + (1-X)X\right) = X.$$

The proportion of individuals in region A after selection is given by

$$X' = \frac{1}{P}\sum_{i=1}^{P}\left[i \in A\right].$$

On average $\mathbb{E}\left(X'\right) = X$ so that crossover does not change the proportion of individuals in the two regions on average. However

$$\begin{aligned}
\mathbb{E}\left((X')^2\right) &= \mathbb{E}\left(\left(\frac{1}{P}\sum_{i=1}^{P}\left[i \in A\right]\right)^2\right) \\
&= \mathbb{E}\left(\frac{1}{P^2}\sum_{i=1}^{P}\left[i \in A\right]^2 + \frac{1}{P^2}\sum_{i=1}^{P}\sum_{\substack{j=1 \\ j \neq i}}^{P}\left[i \in A\right]\left[j \in A\right]\right)
\end{aligned}$$

However since $[i \in A]$ is an indicator function $[i \in A]^2 = [i \in A]$ and thus

$$\begin{aligned}\mathbb{E}\left((X')^2\right) &= \mathbb{E}\left(\frac{1}{P^2}\sum_{i=1}^{P}[i \in A] + \frac{1}{P^2}\sum_{i=1}^{P}\sum_{\substack{j=1\\ j\neq i}}^{P}[i \in A]\,[j \in A]\right)\\ &= \frac{X}{P} + \left(1 - \frac{1}{P}\right)X^2\end{aligned}$$

and consequently

$$\mathbb{E}\left((X')^2\right) - \mathbb{E}\left(X'\right)^2 = \frac{X(1-X)}{P}.$$

This fluctuation is identical to that arising from random sampling in selection. In consequence, the effective population size is halved when we include crossover.

B Removing the Divergence in the Integral

We can directly remove the divergence in the integrand for $T(x)$ in equation (5). Doing so we obtain

$$\begin{aligned}T(x) &= 2\,P\left(x\log(x) + (1-x)\log(1-x)\right)\\ &\quad +2\,P\int_x^1 \mathrm{e}^{2\epsilon P\eta(1-\eta)}\left(c + \int_{1/2}^{\eta}\frac{\mathrm{e}^{-2\epsilon P\xi(1-\xi)} - 1}{\xi(1-\xi)}\,\mathrm{d}\xi\right)\mathrm{d}\eta\\ &\quad +2\,P\int_x^1\left(\mathrm{e}^{2\epsilon P\eta(1-\eta)} - 1\right)\left(\log(\eta) - \log(1-\eta)\right)\,\mathrm{d}\eta.\end{aligned}$$

When $\epsilon = 0$ the integral vanishes leaving $T(1/2) = 2\,P\log(2)$. This is the average symmetry breaking time for the model whose dynamics is due to sampling fluctuations alone.

C Asymptotic Behaviour

We would like to understand the behaviour of the symmetry breaking time (equation (5)) when ϵP becomes large. Obtaining the asymptotic behaviour of equation (5) directly is difficult because the equation involves a complicated double integral. Instead we divide up the dynamics into three epochs as shown in figure 17. Our approach is easy to understand although rather messy in practice. We estimate the average time spent in each epoch. In the first and third epoch the dynamics is assumed to be dominated by the diffusion term while in the second epoch the dynamics is assumed to be dominated by the selection term. In this approximation we must determine the average duration in each epoch.

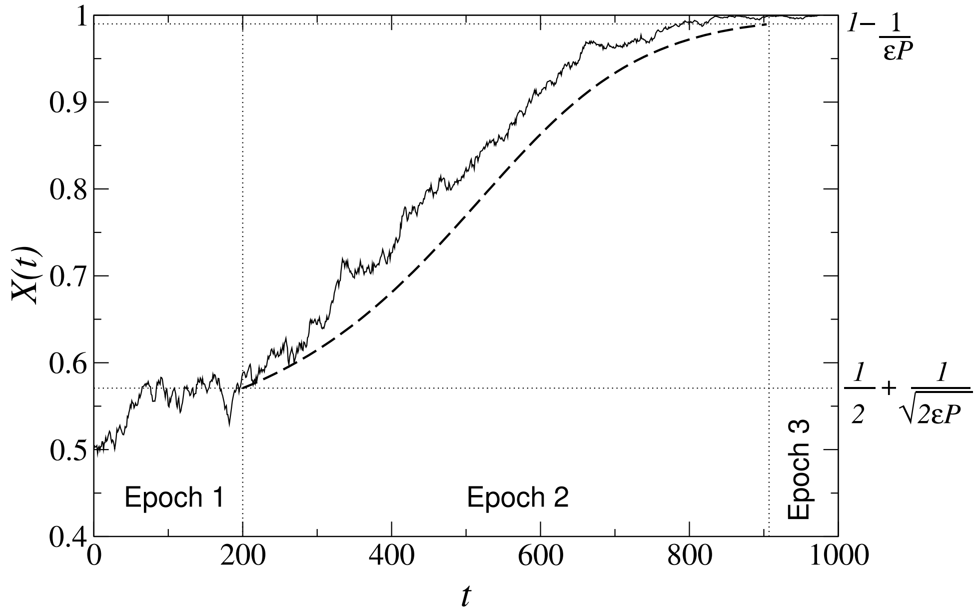


Figure 17: The proportion of the population in region A as a function of time. The simulation is for a population of 10 000 individuals with $\epsilon = 0.01$. The dashed line shows the behaviour when we ignore the diffusion term.

Epoch 1

In the initial epoch (assuming a random initial population) $X(t) \approx 1/2$ so that the selection term $\epsilon X(t)(1 - X(t))(2X(t) - 1)\,\mathrm{d}t$ will be very small compared to the diffusion term. We cannot directly compare the magnitude of the diffusion and selection terms as this depends on the time scale. On short time scales the diffusion term always dominates while on long time scales the selection term will dominate. We will assume that the first epoch ends when $X(t)$ is sufficiently far from $1/2$ that the selection term is likely to prevent the diffusion term taking $X(t)$ back to $X = 1/2$. Assume that $X(t) = 1/2 + \delta_1$. The typical change in $X(t)$ in time Δt due to diffusion alone (assuming δ_1 is small) is of order $\sqrt{\Delta t/(4P)}$. Thus the average time taken for $X(t)$ to move a distance δ_1 by diffusion $\Delta t = 4P\delta_1^2$. This is the typical time for diffusion to take $X(t)$ from $1/2 + \delta_1$ back to its start at $X = 1/2$. For this not to happen the change in $X(t)$ due to selection should be of order δ_1. The change in $X(t)$ due to selection in time Δt when $X(t) = 1/2 + \delta_1$ (again assuming δ_1 is small) is approximately $\epsilon\,\delta_1\,\Delta t/2$. Thus, selection will be sufficient to prevent diffusion from reversing symmetry breaking when $\epsilon\,\delta_1\,\Delta t/2 \approx \delta_1$ or substituting in our estimate for Δt we obtain $\delta_1 = 1/\sqrt{2P\epsilon}$. With the average time spent in epoch 1 equal to $T_1 = \Delta t = 2/\epsilon$.

Epoch 2

Under the selection term alone $X(t)$ would approach 1 (or 0) exponentially, but it would never quite reach the limit. Complete symmetry breaking in finite time is accomplished by the diffusion term. We argue below that when $X = 1 - 1/(\epsilon P)$ (or $X = 1/(\epsilon P)$ if $X \to 0$) we enter epoch 3. To calculate the duration of epoch 2 we ignore the diffusion term giving a time

$$
\begin{aligned}
T_2 &= \frac{1}{\epsilon} \int\limits_{1/2+1/\sqrt{2\epsilon P}}^{1-1/(\epsilon P)} \frac{\mathrm{d}x}{x(1-x)(2x-1)} \\
&= \frac{1}{\epsilon}\left(2\log(\epsilon P) - 3\log(2) - \frac{5}{\epsilon P} \pm O\left((\epsilon P)^{-2}\right)\right).
\end{aligned}
$$

Epoch 3

To estimate when the diffusion term becomes important again (the beginning of epoch 3), we assume that we are at a point $X = 1 - \delta_2$ and ask what the typical time it would take to reach $X = 1 - \delta_2/2$. Under selection (assuming δ_2 is small) this takes a time $\Delta t = \log(2)/\epsilon$. Under diffusion the typical time, Δt, it takes to travel a distance $\delta_2/2$ is approximately given by $\sqrt{\delta_2\,\Delta t/P} \approx \delta_2/2$ or $\Delta t = P\delta_2/4$. Diffusion becomes faster than selection when $\delta_2 = 4\log(2)/(P\epsilon)$. This estimate is crude. In particular, we did not take into account that the diffusion rate effectively halves, so we have overestimated the speed of diffusion. In addition, diffusion is equal likely to decrease $X(t)$. We will take the start of epoch 3 to occur when $X = 1 - 1/(\epsilon P)$ (or $X = 1/(\epsilon P)$ if $X \to 0$). Finally we have to estimate the time taken in epoch 3. This is harder to estimate as the diffusion rate depends on how close to $X = 1$ we currently are. From the argument given above we would expect the time to travel a distance δ would be $P\delta$ so that the typical diffusion time would be of order $1/\epsilon$—we take $T_3 = 1/\epsilon$. Times for completing both epoch 1 and epoch 3 will have large variances as they are dependent on diffusion.

Putting together our three estimates for the duration of each epoch, the expected symmetry breaking time in the limit when ϵP becomes large is

$$
\frac{T}{P} \sim \frac{2\log(\epsilon P) + 3(1-\log(2))}{\epsilon P} \pm O\left(\frac{1}{(\epsilon P)^2}\right).
$$

There was considerable arbitrariness in choosing the epoch boundaries δ_1 and δ_2. Changing the values of these boundaries by a multiplicative factor does not change the leading order term but will change the term $3(1-\log(2))/(\epsilon P)$. From figure 16, it is apparent that the asymptotic formula above provides an accurate estimate of the symmetry breaking time for $\epsilon P > 10$—much more accurate, in fact, than we could legitimately expect from the crudeness of the approximations we have made.

References

[1] J. L. Shapiro and A. Prügel-Bennett, "Genetic algorithms dynamics in two-well potentials with basins and barriers," in *Foundations of Genetic Algorithms 4* (R. K. Belew and M. D. Vose, eds.), (San Francisco), pp. 101–116, Morgan Kaufmann, 1997.

[2] A. Rogers and A. Prügel-Bennett, "The dynamics of a genetic algorithm on a model hard optimization problem," *Complex Systems*, vol. 11, no. 6, pp. 437–464, 2000.

[3] A. Rogers and A. Prügel-Bennett, "A solvable model of a hard optimization problem," in *Theoretical Aspects of Evolutionary Computing* (L. Kallel, B. Naudts, and A. Rogers, eds.), Natural Computing, (Berlin), pp. 207–221, Springer, 2001.

[4] T. Jansen and I. Wegener, "On the analysis of evolutionary algorithms — a proof that crossover can really help," in *Proceedings of the 7th Annual European Symposium on Algorithms (ESA'99)* (J. Nešetřil, ed.), (Berlin), pp. 184–193, Springer, 1999.

[5] A. Prügel-Bennett, "When a genetic algorithm outperforms hill-climbing," *Journal of Theoretical Computer Science*, 2001. Invited contribution for special issue on soft computing, in press.

[6] T. Jansen and I. Wegener, "Real royal road functions—where crossover provably is essential," in *GECCO*, (San Francisco, California, USA), pp. 1034–1041, Morgan Kaufmann, 2001.

[7] A. Prügel-Bennett and J. L. Shapiro, "The dynamics of a genetic algorithm for simple random Ising systems," *Physica D*, vol. 104, pp. 75–114, 1997.

[8] P. Galinier and J. K. Hao, "Hybrid evolutionary algorithms for graph coloring," *Journal of Combinatorial Optimization*, vol. 3, no. 4, pp. 379–397, 1999.

[9] C. A. Glass and A. Prügel-Bennett, "Genetic algorithms for graph colouring: Exploration of Galinier and Hao's algorithm," *Journal of Combinatorial Optimization*, 2001. Accepted for publication.

[10] N. J. Radcliffe, "Genetic set recombination and its application to neural network topology optimisation," *Neural Computing and Applications*, vol. 1, no. 1, 1992.

[11] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, and L. Troyansky, "Determining computational complexity from characteristic 'phase transisition'," *Nature*, vol. 400, pp. 133–137, 1999.

[12] L. Pitt and L. G. Valient, "Computational limitations on learning from examples," *Comput. Machinary*, vol. 35, p. 965, 1988.

[13] H. Horner, "Dynamics of learning for the binary perceptron problem," *Zeitschrift für Physik: B*, vol. 86, pp. 291–308, 1992.

[14] E. Gardner, "The space of interations in neural network models," *J. Phys. A*, vol. 21, p. 257, 1988.

[15] E. Gardner and B. Derrida, "Optimal storage properties of neural network models," *J. Phys. A*, vol. 21, p. 271, 1988.

[16] W. Krauth and M. Mézard, "Storage capacity of memory networks with binary couplings," *Journal de Physique*, vol. 50, pp. 3057–3066, 1989.

[17] O. C. Martin, R. Monasson, and R. Zecchina, "Statistical mechanics methods and phase transitions in optimization problems," *Theoretical Computer Science*, vol. 265, no. 1-2, pp. 3–67, 2001.

[18] M. Mézard, G. Parisi, and R. Zecchina, "Analytic and algorithmic solution of random satisfiability problems," *Science*, vol. 297, pp. 812–815, 2002.

[19] M. Mézard and R. Zecchina, "The random K-satisfiability problem: from an analytic solution to an efficient algorithm," *Phys. Rev. E*, vol. 66, p. 056126, 2002.

[20] L. L. A. Crisanti and G. Parisi, "The 3-Sat problem with large number of cleases in the ∞-replica symmetry breaking scheme," *Journal of Physics:A*, vol. 35, pp. 481–497, 2002.

[21] M. Mézard, G. Parisi, and M. A. Virasoro, *Spin-Glass Theory and Beyond*, vol. 9 of *Lecture Notes in Physics*. Singapore: World Scientific, 1987.

[22] P. Stadler, "Spectral landscape theory," in *Evolutionary Dynamics — Exploring the Interplay of Selection, Neutrality, Accident and Function* (J. Crutchfield and P. Schuster, eds.), (Oxford), Oxford University Press, In Press.

[23] C. Flamm, I. L. Hofacker, P. F. Stadler, and M. T. Wolfinger, "Barrier trees of degenerate landscapes," *Z. Phys. Chem.*, vol. 216, pp. 155–173, 2002.

[24] C. M. Reidys and P. F. Stadler, "Combinatorial landscapes," *SIAM Review*, vol. 44, pp. 3–54, 2002.

[25] J. Hallam and A. Prügel-Bennett, "Barrier trees for search analysis," in *GECCO 2003*, 2003. In Press.

[26] A. Prügel-Bennett, "Finite population effects for ranking and tournament selection," *Complex Systems*, vol. 12, no. 2, pp. 183–205, 2000.

[27] R. A. Fisher, "On the dominance ratio," *Proc. Roy. Soc. Edin.*, vol. 42, pp. 321–341, 1922.

[28] R. A. Fisher, "The distribution of gene ratios for rare mutations," *Proc. Roy. Soc. Edin.*, vol. 50, pp. 205–220, 1930.

[29] S. Wright, "Evolution in Mendelian populations," *Genetics*, no. 24, pp. 97–159, 1931.

[30] M. Kimura and N. Takahata, *Population Genetics, Molecular Evolution, and Nutral Theory: Selected Papers of Motoo Kimura*. Chicago: The University of Chicago Press, 1st ed., 1994.

[31] J. F. Crow and M. Kimura, *An Introduction to Population Genetics Theory*. New York: Harper and Row, 1970.

[32] W. J. Ewens, *Mathematical Population Genetics*. Berlin, Heidelberg, New York: Springer, 1979.

[33] B. Øksendal, *Stochastic Differential Equations*. Berlin, Heidelberg, New York: Springer, 1985.

[34] M. Kimura, “Diffusion models in population genetics,” *Journal of Applied Probability*, vol. 1, pp. 177–232, 1964.

[35] R. Durrett, *Probability Models for DNA Sequence Evolution*. Probability and its applications, New York: Springer, 1st ed., 2002.

[36] M. Kimura and T. Ohta, “The average number of generations until fixation of a mutant gene in a finite population,” *Genetics*, vol. 61, pp. 763–771, 1968.

[37] W. B. Provine, *Sewall Wright and Evolutionary Biology*. Chicargo: The University of Chicargo Press, 1986.

[38] J. A. Coyne, N. H. Barton, and M. Turelli, “Perspective: A critique of Sewall Wright’s shifting balance theory of evolution,” *Evolution*, vol. 71, pp. 643–671, 1997.

[39] E. Cantú-Paz, *Efficient and Accurate Parallel Genetic Algorithms*, vol. 1 of *Genetic Algorithms and Evolutionary Computation*. Boston: Kluwer Academic Publishers, 1st ed., 2000.