A Building-Block Royal Road Where Crossover is Provably Essential

Richard A. Watson
Electronics and Computer Science
University of Southampton
Southampton, SO17 1BJ, UK
+44 23 8059 2690
raw@ecs.soton.ac.uk

Thomas Jansen
Universität Dortmund
FB Informatik, LS 2
44221 Dortmund, Germany
+49 231 755 4702
Thomas.Jansen@udo.edu

ABSTRACT

One of the most controversial yet enduring hypotheses about what genetic algorithms (GAs) are good for concerns the idea that GAs process building-blocks. More specifically, it has been suggested that crossover in GAs can assemble short low-order schemata of above average fitness (building blocks) to create higher-order higher-fitness schemata. However, there has been considerable difficulty in demonstrating this rigorously and intuitively. Here we provide a simple building-block function that a GA with two-point crossover can solve on average in polynomial time, whereas an asexual population or mutation hill-climber cannot.

Categories and Subject Descriptors

I.2.8 [**Artificial Inteligence**]: Problem Solving, Search – *heuristic methods*; F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Non-numerical Algorithms and Problems.

General Terms

Algorithms, Performance, Theory.

Keywords

Mutation, crossover, modularity, building block hypothesis, genetic algorithms theory, royal roads.

1. WHAT A GA IS GOOD FOR

It follows from the NFL theorem [12] that for both crossover and mutation there are classes of functions where one operator excels. Accordingly, there are many examples for evolutionary algorithms using only mutation performing provably well e.g. [5]. But, finding rigorous proofs of such results for evolutionary algorithms with crossover is much harder. The building block hypothesis, BBH, [6][8][10][11][17] suggests that a GA will perform well when it does because crossover can assemble short low-order schemata of above average fitness (building blocks) to create higher-order schemata of higher-fitness. Much effort has been directed at finding functions that discriminate the abilities of crossover-based and mutation-based algorithms [1][14][17][23][26][27], and recently some functions have been provided that show a principled distinction [13][22][26]. But interestingly, the first of these to prove rigorously that a GA with crossover can find the global optimum in expected time polynomial in the problem size, whereas a mutation-based algorithm takes on average exponential time [13], does not use explicit building-blocks. This function, like that in [21][22], incorporates a wide fitness valley that must be overcome by a crossover event and all other fitness improvements are provided by mutation. These functions seem quite contrived: There is no intuitive explanation for why it should be the case that the positions of the global optimum and local optima should be just as they are (such that the global optimum is located at a genotype produced by crossover of the locally optimal genotypes). Although such a function does satisfy the essential property of distinguishing polynomial versus exponential expected time complexities of crossover and mutation-based algorithms respectively, their contrived structure makes it difficult to see what they have to offer with respect to our understanding of what GAs are good for in general.

Some of the other functions that show a principled distinction in the ability of crossover [26][27] do use a building-block structure but are too complex to facilitate rigorous proofs for their time complexities [13]. In particular, the HIFF function [27] illustrates building-block structure directly inspired by the BBH but the ability of a GA to outperform mutation-based algorithms on this function is dependent on its multi-level building-block structure, and indeed the requirement for a true GA rather than a 'crossover hill-climber' [4][27] requires additional modifications to the function (specifically, the use of non-complementary global optima [27]). As yet, no simple single-level building-block function has been provided where crossover is provably essential.

In this paper we return to a simple separable building-block function to see if we can rescue some intuition and yet maintain rigorous proofs that distinguish polynomial and non-polynomial time complexities for a GA with and without crossover respectively. In this paper we address only the original conception of a building-block that includes the assumption of tight linkage [6][8][10][11][17] and ordinary one- or two-point crossover that could potentially exploit this structure as described in the BBH. Note that a large body of work on linkage learning [9] and model-building methods, e.g. [19], has developed in the GA community that addresses the exploitation of schemata (of bounded size) with above-average fitness *without* the assumption of tight linkage. But the original (i.e. tight-linkage) building-block idea, where the bits of a block are close together on the genome, has previously failed to demonstrate an advantage for ordinary crossover. We will not

GECCO'07, July 7–11, 2007, London, England, United Kingdom.

address genetic algorithms with linkage learning methods, model building methods or crossover methods that use learned linkage information since this is not relevant to the original BBH, or necessary to show a principled advantage to crossover.

The intuitive concept we wish to emphasise is simply that a GA with crossover can provide something that mutation cannot because it allows selection to operate on building-blocks. That is, we want a function where mutation finds good building blocks and then these good blocks can be brought together to create high-fitness genotypes by using the ability of crossover to separate blocks from the genetic backgrounds they arose in whilst keeping the bits of a block together (the latter requires tight linkage).This is a simple intuitive concept that has been sought after since the inception of the GA [10][11], notably with the use of the royal road functions [6][7][17] discussed in the next section. However, this apparently simple intuition has problems: 1) If there is any fitness-gradient information within a block that makes solving a block easy then it appears to make the whole problem easy without the need for crossover. Conversely, any intra-block gradient information that hinders mutation also hinders discovery of blocks in the GA and perhaps should be ignored. 2) Yet, if gradient information within blocks is ignored then blocks must be small enough for the best bit-configurations to be guessed (e.g. in the initialisation of the population), but if this is so then they are also small enough to be solved by mutation in reasonable time. (Note that taking time exponential in the block size for any constant block size, as opposed to the problem size, does not produce the principled distinction we seek, as we will discuss later.) We will discuss how this problem is overcome with the use of large blocks that are *semi-reliably* solvable by following internal fitness gradients.

It is notable that some very well-known functions, the concatenated trap functions [2][3] discussed in the next section, are almost suitable to illustrate the effect we want, but do not (in the form they are usually used) resolve the above problems wsince they assume small blocks with deceptive internal gradients. Much of the prior work with trap functions, although directed at understanding the abilities of GAs, was not directed at testing the BBH *per se* since it did not assume tight linkage (i.e. did not assume that schemata of above average fitness had short defining length). If schemata do not have short defining length then crossover cannot keep the bits of a schema together during recombination events. In such cases a more general crossover model such as uniform crossover, which does not depend on this assumption, is more appropriate – but uniform crossover does not address the capacity of a GA to select on building-blocks since uniform crossover provides no genetic linkage between any bit and any other bit.

Maintaining diversity has been a tricky aspect of previous work on the benefit of crossover, requiring methods like deterministic crowding in some cases [27] or the removal of duplicate genotypes in others [13]. In this paper we use and analyse a multi-deme population, or Island model, for the purposes of maintaining diversity.

In the next section we discuss some background on prior building-block functions. In section 3 we introduce our new function. In Section 4 we analyse the expected behaviour of mutation-based methods and a multi-deme GA with crossover. In Section 5 we illustrate the behaviour of a GA on this function with simulation experiments. Section 6 concludes.

2. BUILDING-BLOCK FUNCTIONS

The royal road functions [6][17] were specifically designed to test the building block hypothesis by providing an idealised problem structure presenting short low-order schemata of above average fitness that contain the global optimum. The royal roads are a simple separable building-block function. The fitness of a genotype *F(G)* in a separable building-block function (with tight linkage) can be defined using Eq.1.

$$F(G) = \sum_{i=1}^{B} f(g_i) \qquad \text{Eq.1.}$$

where $G = \langle x_1, x_2, \ldots x_N \rangle$ is a binary genotype of N bits, $B=N/k$ is the number of blocks in the function, $g_i = \langle x_{(i-1)k+1}, x_{(i-1)k+2}, \ldots x_{ik} \rangle$ is the i^{th} sub-block of G, and $\boldsymbol{f}$ is a sub-function defining the fitness of each block. In the royal roads $\boldsymbol{f}$ is defined as per Eq.2.

$$f(g) = \begin{cases} k, & if\ u(g) = k, \\ 0 & otherwise. \end{cases} \qquad \text{Eq.2.}$$

where g is a sub-block of k bits, and $u(g)$ is the unitation of g, i.e. the number of ones in g. Note that in the royal roads there is no selective gradient within each block to guide selection to find good blocks, and accordingly the expected time to find a good block is exponential in the block size, $k=N/B$. However, since in the canonical form of the function N=64 and $k=B$=8 it is not too difficult for a GA to find good blocks of all-ones.

However, when all schemata of above average fitness contain the global optimum, as is the case in the royal roads, there are no local optima in the resultant fitness landscape. Accordingly, although a GA with crossover is able to assemble building blocks as expected in this function, it is also possible to find the global optimum efficiently with a random mutation hill-climber (RMHC) [6]. The important point to note is that it is not too difficult for either a GA *or* a mutation-only algorithm to find good solutions to blocks. Hence it should not be surprising that a mutation-only algorithm finds the global optimum in this problem in time that is polynomial in N for small k.

Perhaps in large part because the GA failed to outperform a hill-climber on a function specifically designed to exemplify building block recombination, the GA community is somewhat divided into those that have discarded the building block idea altogether and those that moved their interest to blocks that do not depend on tight linkage. Nonetheless, it should be noted that the failure of the royal roads to demonstrate the advantage of a GA via building block assembly, despite their explicit design to do so, does not prove that this intuition has no value.

The concatenated trap functions [2] used a simple building-block structure like the royal roads but also included *deceptive* [2][3] schemata that did not contain the global optimum. A simple trap function is given by the sub-function defined in Eq.3.

$$f(g) = \begin{cases} k, & if\ u(g) = k, \\ (k - u(g))/2, & otherwise. \end{cases} \qquad \text{Eq.3.}$$

This sub-function, like Eq.2. of the royal roads, has a maximum at the all-ones configuration. However, it also contains a secondary lower-fitness local optimum at all-zeros, and importantly, the local fitness gradient at all points (other than those that neighbour the all-ones genotype) misleads a mutation hill-climber towards the sub-optimal blocks. This causes a mutation hill-climber to become trapped at locally-optimal all-zero blocks. Intuitively, this makes the good blocks harder to find and should make the problem difficult for a mutation-only algorithm. However, as noted, this structure also makes these

blocks more difficult to solve with the mutational processes within a GA. Often it is assumed that in the worst case the fitness gradients within a block should be ignored altogether, and the solutions to each block are therefore often assumed to be provided by random initialisation of the population.

At this stage it is worth pointing out that any problem that is linearly separable into blocks of order k, where k is a constant, can be solved by a mutation hill-climber in polynomial time (i.e. time to solution is a polynomial function of the problem size, N). This is quite obvious when the locations of blocks are assumed to be known since the configuration space of each block can be enumerated in 2^k steps, and there are only N/k blocks to solve, requiring $N2^k/k$ steps in total. Mutational methods that exploit the assumption of tight linkage without explicit knowledge of the module boundaries (e.g. MMHC discussed later) also perform efficiently [14].

But even if the linkage is assumed to be random such problems are still solvable in polynomial time. Specifically, the time to find the global optimum can be expressed as $T \leq tS$, where t is the maximum time for a mutation hill-climber to find a fitness improvement and S is the maximum number of fitness improvements necessary to find the global optimum [27]. For blocks of size k, for any genotype that is not optimal, there is always a fitness improvement available by changing at most k bits. The expected time to find this improvement by k-bit mutation[1] is $t < AB$ where $A=(N \text{ choose } k) \leq N^k/k!$ is the number of ways of choosing k bits from N, and $B=2^k$ is the number of different k-bit combinations. So, $t < N^k2^k/k!$. S, the number of fitness improvements necessary to reach the global optimum is at most $S=CD$, where $C \leq 2^k$ is the maximum number of fitness improvements in a block, and $D=N/k$ is the number of blocks. So, $S \leq N2^k/k$. Thus the total time to find the global optimum, $T \leq tS < (N^k2^k/k!)(N2^k/k)$ is therefore a polynomial function of N when k is a constant.

This is not a tight bound and naturally there are many ways to devise methods that can do better than this given appropriate assumptions about the nature of the sub-functions in the problem. But this observation is sufficient to show that problems with this kind of bounded difficulty (order-k separable functions with constant k, regardless of linkage) cannot be used to show a principled polynomial versus non-polynomial distinction between the time to solution for crossover and a mutation since they do not require more than polynomial time for a mutation-based method. In fact, for the same reason they cannot be used to show that any algorithm, any kind of GA (including linkage learning and model building methods) or anything else, can solve something in polynomial time that requires non-polynomial time for a mutation-only process.

Accordingly, showing a principled distinction between crossover and mutation using a building block function will require large blocks; a k which is a function of N. But in the royal roads and the concatenated deceptive trap functions a mutational process will take at least time exponential in the block size to solve a block since there is no useful fitness gradient within each block to guide a (mutation-based) search process. Yet if there were a useful fitness gradient within each block to guide a mutation-based search process to find the correct solution to a block in polynomial time, then it seems that the total time to solve the whole problem with mutation will also be polynomial. Given that blocks are separable and solving the whole problem is simply matter of solving each of the sub-problems, there appears to be no way to make the blocks difficult to solve, and yet make the whole problem easy for a GA to solve.

A very simple solution to these issues is simply to use blocks with size that is a function growing with N, and where the basin of attraction of both the optimal schema and non-optimal schemata remain some constant proportion of the total intra-block search space. This is provided by large trap functions, but they must not be fully-deceptive [2][3] else the basin of the preferred optimum becomes a vanishingly small proportion of the intra-block search space as k increases.

Specifically, the following sub-function serves to provide these simple properties.

$$f(g) = \begin{cases} k, & \text{if } u(g) = k \\ u(g)/2, & \text{if } k > u(g) > k/2, \\ (k - u(g))/2, & \text{otherwise.} \end{cases} \quad \text{Eq.4.}$$

This sub-function, like the previous two, has a maximum at all-ones. Like Eq.3. it also has a sub-optimum at all-zeros. However, unlike Eq.3. the basins of attraction for the optimum and the sub-optimum are of equal size (for even k) in this function. This means that a mutation hill-climber will find the optimal schema (in polynomial time) with probability 0.5 even in large blocks, whereas in Eq.3. this probability approaches 0 for large blocks.

We want k to be large so that mutation hill-climbers cannot escape the sub-optimal solutions in time polynomial in N. We cannot use $k=N$ because a single-block function would obviously make the probability of finding the optimal genotype 0.5. Although the basin of attraction of the optimal block should be a significant fraction of each intra-block search space, the basin of attraction of the global genotype should be an insignificant fraction of the total problem search space. The basin of attraction of the optimal genotype, given that each block is independent, is 0.5^B (see Fig.2). So we want B large and k large. Since $k=N/B$, the best compromise is to use $k=B=\sqrt{N}$.

The resolution to the apparent no-win trade-off between making blocks difficult for mutation hill-climbing or easy for mutation hill-climbing mentioned in the introduction is simply that although there *is* useful fitness gradient information within each block, the problem as a whole is not easily solvable by a mutation-only method because the basin of attraction for the global optimum is only $2^{-\sqrt{N}}$ of the search space, and every other optimum is at least $\sqrt{N}$ bits away from the global optimum. Nonetheless, utilisation of the mutational gradients within each block is crucial else a GA could not find the solution to even one block in polynomial time.

Eqs.1 and 4 thus define a problem that is sufficient for the aims of this paper. We clearly have not analysed this formally yet but we have built some intuition for the analysis that follows. However, this form of function is still more contrived than it needs to be, so before we do any analysis we generalise this function. In particular, there is no need for each sub-function to be a trap function where the sub-optimal configuration is the complement of the optimal configuration, nor is there any need to suppose that there are exactly two local-optima. These properties are a vestige of prior work that wanted to analyse fully-deceptive functions where the sub-optimum must be the complement of the optimal configuration. Being fully-deceptive actually defeats our aims, so the use of complementary optima is completely

[1] In principle, if k is not known the mutational process can be repeated for k=1, k=2 and so on, and yet still produce an overall time that is polynomial in N.

unnecessary. In the next section we describe a more general sub-function which shows that the ability of a GA to utilise building-block structure can be general and intuitive.

3. A GENERAL BUILDING-BLOCK FUNCTION

In this section we introduce a general and simple building-block function that we can use to prove a polynomial versus exponential distinction in the average time to find the global optimum for a crossover-based and mutation-only algorithm, respectively. The function simply concatenates sub-functions each of which has multiple fitness peaks of different heights, where each peak has an appreciable basin of attraction for a mutation hill-climber. As before, this is a separable building-block function:

$$F(G) = \sum_{i=1}^{B} f(g_i) \qquad \text{Eq.5.}$$

where $G = \langle x_1, x_2, \ldots x_N \rangle$ is a binary genotype of N bits, $B=N/k$ is the number of blocks in the function, $g_i = \langle x_{(i-1)k+1}, x_{(i-1)k+2}, \ldots x_{ik} \rangle$ is the i^{th} sub-block of G, and $\boldsymbol{f}$ is a sub-function defining the fitness of each block as per Eq.6.

$$f(g_i) = \sum_{j=1}^{T_i} c(g_i, t_{ij}) \text{, where}$$

$$c(g, t_{ij}) = \begin{cases} w_{ij}, & \text{if } |g - t_{ij}| = 0, \\ \left(1 + |g - t_{ij}|\right)^{-1}, & \text{otherwise.} \end{cases}$$

Eq.6.

where T_i is the number of different fitness peaks within the i^{th} block, t_{ij} is the j^{th} target string (peak) within the i^{th} block, $|g\text{-}t|$ is the Hamming distance between the genotype of the block and the target string, and $w_{ij}>1$ is the fitness weighting of t_{ij}.

Eq.6 defines the fitness of a block to be the sum of fitness contributions related to a set of T target strings. The fitness contribution, defined by $\boldsymbol{c}$, related to a particular target string is maximal when the block matches the target string and is otherwise inversely proportional to the distance from the target string. We see that Eq. 4 is qualitatively similar to a particular instance of Eq.6 where $T_i=2$, $t_{i1}=0^k$ and $t_{i2}=1^k$, and $w_{i1}=k/2$ and $w_{i2}=k$, for all i. Since the slopes approaching each peak are the same for all peaks, the fraction of the intra-block search space that is in the basin of attraction for each peak, including the fitness optimum of the block, will be on average $1/T$ for T randomly positioned peaks (regardless of k).

With some care in interpretation, we can visualise Eq.6 by substituting a one-dimensional metric space for the k-dimensional binary space of a block and using Euclidian distance instead of Hamming distance. Fig. 1 shows an example using T=4 with random t_j and random $1<w_j<2$.

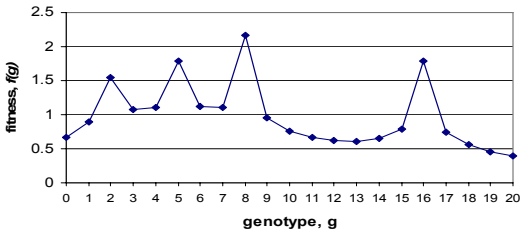

Figure 1. A one-dimensional, 4-peak sub-function (Eq.6).

Thus the sub-function used in this problem is nothing more than a random multi-peaked function (it is important though that the number of peaks is not exponential in k, so a purely random intra-block fitness landscape is not appropriate). However, the fact that the overall function, Eq. 5, sums several blocks of this form creates an overall fitness landscape that has significant structure. We can visualize the symmetries created by using a two-block example as in Fig. 2. - this uses the sub-function shown in Fig.1 for g_1 and a different random sub-function (with three peaks) for g_2. Any function created by the summation of fitness contributions from several independent sub-functions has this kind of symmetry, so this is quite a natural fitness landscape structure to expect for a problem that results from several sub-problems. Since the sub-functions are also simply random-peaked landscapes, the only special features of the overall function is that it is composed of several sub-functions, each containing multiple-peaks, and the optimum of each peak has a significant basin of attraction. This is, we argue, much more natural and less contrived than the functions in prior work, even if it is a bit more complicated to define.

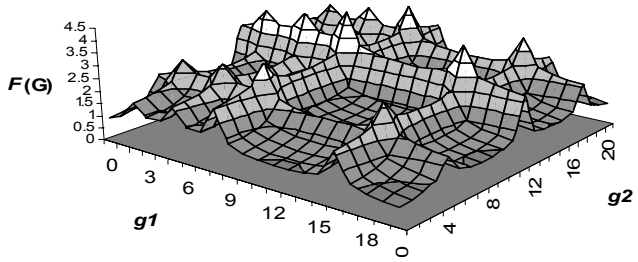

Figure 2. A two-block random function, Eq.5, T_1=4,T_2=3.

Note that the number of local optima in the overall function is the product of the number of peaks in each sub-function. Because the blocks are separable, high-fitness points in the overall function are created by combining high-fitness points in the sub-functions. Thus without any contrivance in the positioning of the global optimum with respect to the position of the local optima, it is nonetheless the case that the global optimum is both mutationally isolated from the local optima and yet exactly in the right position to be reached by crossover of locally optimal genotypes.

4. ANALYSIS

We address the behaviour of mutation-based and crossover-based algorithms on the function defined by Eq.5. (given small T and large k). Before providing formal analysis, we first sketch the expected behaviour which should now be quite intuitive. The crux of the result depends on the simple observation that since the optimal configuration for each block will not be found reliably by mutation, it is very unlikely that all optimal building-blocks will be found in any one individual in a population (thus defeating a purely mutation-based algorithm), nonetheless it is very likely that each optimal building-block will be found in at least one individual in the population (and this is sufficient for a crossover method to produce a genotype with all the optimal blocks). A mutation-based algorithm will become stuck at one of the local optima, requiring a specific multi-point mutation to escape to a higher-fitness optimum. The number of bits that must be mutated simultaneously in this case is determined by the width of fitness saddles between local optima, which will be $\boldsymbol{O}(k)$ on average

assuming T is constant. If we chose k to be a function of N, such as $k=\sqrt{N}$ as suggested in the previous section, then the number of bits that need to change to escape a local optimum will be $\boldsymbol{O}(\sqrt{N})$.

The basic idea for a GA with crossover is that different individuals are likely to find different local optima in time polynomial in k (and therefore N). With probability $1/T$, on average, a given individual will find the optimal configuration for a given block, but accordingly, it will find a suboptimal configuration with probability $1-1/T$. Nonetheless, so long as the good configuration for a block is found by some individual in the population, selection in a sexual population will be able to increase the frequency of this building-block (semi-) independently of the inferior configurations that may have been found at other blocks in that individual. If this happens independently for all blocks then the maximum fitness genotype will be discovered. Since the expected time to find the best configuration for each block (in at least some fraction of the population) is polynomial in N, and the expected time to find appropriate crossover points to put these blocks together is polynomial in N, the overall expected time for a crossover GA will also be polynomial in N.

To provide formal analysis of this intuition, we may assume without loss of generality, that the sub-functions are identical for each block. To keep analysis simple we assume the minimum number of peaks per block that will suffice, i.e. $T_i=2$ for all i. We also assume that each t_{ij} is chosen such that the Hamming distance between the peaks is exactly the average that would be expected with random peaks, i.e. $k/2$. For example, $t_{i1}=\{01\}^{k/2}$ and $t_{i2}=1^k$ would suffice. We also assume that each $w_{i1}=1$ and $w_{i2}=2$. The maximum fitness string is thus all-ones.

Mutation: First considering evolutionary algorithms without crossover, we want to show that such algorithms fail to be efficient. Statements about the performance of evolutionary algorithms cannot be made without describing the evolutionary algorithm considered in some detail. For a negative result, however, it is desirable to make statements about a large class of evolutionary algorithms. Here we consider evolutionary algorithms with the following properties. The initial population is generated uniformly at random, the only variation employed is standard bit mutations mutating each bit independently with some probability p_m to a new random bit. Moreover, the selection employed depends on fitness values only, does not favour smaller fitness values, and does not take any other properties of the individuals into account. For our fitness function the $B = \sqrt{N}$ blocks are independent and each block is symmetric. The class of mutation-based evolutionary algorithms considered here will find in each block only the local optimum with probability ½. Thus, with probability $1-2^{-\sqrt{N}}$, there is at least one block where $\Omega(\sqrt{N})$ bits must be flipped and there are no hints towards the global optimum that can be utilised. Note that for a direct mutation $p_m=1/\sqrt{N}$ is the optimal mutation probability for a direct mutation of $\sqrt{N}$ bits. For any mutation probability the time needed for such a mutation is therefore bounded below by $2^{\Omega(\sqrt{N})}$. The same lower bound holds for the time needed for a completely random walk towards the global optimum which is the best selection can do since fitness values lead back to the local optimum. This is due to the fact we rule out any selection mechanism preferring lower fitness. It follows that such algorithms will not find the unique global optimum within $2^{o(\sqrt{N})}$ steps with probability $1-2^{-\Omega(\sqrt{N})}$.

Crossover: To analyse a crossover-based method is more involved. The algorithm must be treatable such that guarantees for its performance on F can be proven in a rigorous way. Note that it is not too difficult to come up with some evolutionary algorithm that optimizes F quite efficiently in most runs (see simulation results), but here we desire a theoretical analysis advancing from experience to knowledge.

The diversity of the population will, as mentioned, be crucial to the success of the crossover method. Many different mechanisms are known and in use that are designed to maintain the diversity of a population. One way to achieve sufficient diversity is the use of population subdivision or multi-deme models. In fact, a multi-deme model allows us to not be concerned with intra-deme diversity, such that all useful diversity is inter-deme[2]. This enables a simple way for us to control how much genetic diversity is available, and assess how many independent demes are sufficient. In general we assume g independent evolutionary algorithms that exchange some members of their population with some migration rate. We call the number of generations, e, between two such exchanges an epoch, and the independence of the g EAs in the first epoch means that if any local optima are found within this epoch then the local optima found are independent. However, on a multi-epoch timescale where migrants are numerous we cannot guarantee continued independence of the local optima the different demes find. Thus although crossover of migrants provides the opportunity for getting good blocks together it also introduces the possibility that good blocks may be lost. For analysis we avoid this complication by having just one population that receives immigrants from the other populations. This way, the $g-1$ populations that are not receiving any immigrants remain independent throughout the run. Thus, independence of the local optima provided is given all the time.

The choice of EA used in each deme is not so important but must enable us to maintain formal treatment. One of the first choices one can make when designing an evolutionary algorithm is the choice of the overlap between two subsequent populations. A steady state approach has the advantage that the change from one generation to the next is easier to control facilitating analysis. Therefore, we choose to consider a $(\mu+1)$ evolutionary algorithm where a population of size μ produces just one offspring in each step that replaces one member of the population that is not superior with respect to fitness. We leave the choice of μ open for the moment. We use standard bit mutations with mutation probability $1/N$ as one of the variation operators. This is a very common operator, well known to optimize OneMax-like functions efficiently so it will suffice to find the local optima efficiently. Two-point crossover allows the algorithm to exchange single blocks between two parents. Note that uniform crossover would not be helpful since it would perform such an exchange with probability at most $2^{-\Omega(k)}$ which is way too small. We assume a probability of applying crossover in each reproduction, p_c.

We give a formal definition of the GA used in each deme for the sake of clarity. For $x_i \in \{0, 1\}^n$ and $j \in \{1, 2, \cdots, n\}$, $x_i[j]$ denotes the j^{th} bit in x_i.

Algorithm 1. EA for each deme.

Initialization

1. For i:= 1 To μ Do

[2] Clearly, this means that each deme may just as well be replaced with a mutation hill-climber, so long as some provision for crossover of migrants is enabled. Considering a GA as a combination of hill-climbing and crossover is familiar and intuitive and this multi-deme scenario formalises this. It also suggests that some forms of memetic algorithm with crossover will work well on this function.

Choose $x_i \in \{0, 1\}^N$ uniformly at random.

Selection for Reproduction

2. Select $i, j \in \{1, \cdots, \mu\}$ uniformly at random.

Variation

3. With probability p_c

2-Point Crossover

4. Select $c_1 \in \{1, \cdots, N\}$ uniformly at random.
5. Select $c_2 \in \{1, \cdots, N\}$ uniformly at random.
6. If $c_1 > c_2$, exchange c_1 and c_2.
7. $y_1 := x_i[1]x_i[2] \cdots x_i[c_1 - 1]x_j[c_1]x_j[c_1 + 1] \cdots x_j[c_2]x_i[c_2 + 1] \cdots x_i[N]$
8. $y_2 := x_j[1]x_j[2] \cdots x_j[c_1 - 1]x_i[c_1]x_i[c_1 + 1] \cdots x_i[c_2]x_j[c_2 + 1] \cdots x_j[N]$
9. If $F(y_1) \geq F(y_2)$, Then $y := y_1$, Else $y := y_2$.

10. Else

Mutation

11. $y := x_i$
12. Flip each bit in y independently with probability $1/N$.

Selection for Replacement

13. If $F(y) \geq \min\{F(x_1),\cdots,F(x_\mu)\}$ Then let y replace some randomly chosen x_i with minimal F-value.
14. Continue at line 2.

All g demes work this way in normal generations. Every e-th generation, the receiving deme uses an x_j in lines 7 and 8 of the algorithm from some other deme (selected uniformly at random) rather than from its own deme. We call the resulting algorithm a single-receiver multi-deme EA. When measuring its optimization time we use the number of function evaluations to measure time.

Theorem 1. The multi-deme EA with g demes, one receiving deme, with deme size μ, crossover probability p_c, and epoch length e, finds the unique global optimum of F within $\boldsymbol{O}(\mu egN^2 \log N)$ function evaluations with probability $1-\boldsymbol{O}(1/N)$ if $g \geq c \log\sqrt{N}$ and $\xi \leq p_c \leq 1-\xi$ hold for some constants $\xi > 0$ and $c > 1$ sufficiently large.

Proof. It is known that on average after $\boldsymbol{O}(\mu N \log N)$ generations, each deme consists only of local optima [13]. We consider the $B=\sqrt{N}$ blocks that comprise the bit string of length N. Considering one specific block, we see that the expected number of demes where the current best individual contains a global optimum in this block is bounded below by $\lfloor g/2 \rfloor$ (i.e. half the demes will find the better peak rather than the inferior peak). Application of Chernoff bounds [18] yields that we have at least $g/3$ optima with probability $1 - e^{-\Omega(g)}$. Since we have $g \geq c \log B$ sufficiently large, this probability is sufficiently close to 1. Now, we consider the receiving population, only. We consider the current best of this population and give an upper bound on the number of recombinations needed until the unique global optimum is constructed. The probability to select the current best of this population for crossover is $1/g$. The probability to select a second parent with an optimal piece where the already chosen parent is not optimal is $\boldsymbol{\Omega}(1)$ as we have just seen. With probability at least i/N^2, two good crossover points are chosen if i pieces are still not optimal in the current best individual. Summing up the expected waiting times for $i \in \{1, \ldots, B-1\}$ we obtain that $\boldsymbol{O}(\mu egN^2 \log N)$ function evaluations are sufficient on average. Choosing the multiplicative constant sufficiently large yields the claimed bound on the probability.▪

From these analyses we see that for large N a GA with two-point crossover using a reasonable number of subdivided demes and one receiving deme will find the global optimum with probability close to 1 in time polynomial in N. In contrast a mutation based-method, either a GA with mutation-only variation or a mutation hill climber, will, with probability close to 1, take time exponential in $\sqrt{N}$ to find the global optimum. Accordingly, this function is proven to show a principled distinction in the performance of crossover-based and mutation-based algorithms.

For the purposes of analysis, we supposed a $(\mu+1)$ algorithm for each deme with a replacement strategy that ensures that good genotypes are not lost. We also used a subdivided GA where only one deme received migrant individuals from the other demes (and the others only had outgoing migrants). This is assumed simply for the purposes of analytic assurance that the diversity of the initial demes is not lost before it can be used appropriately in the receiving deme. Similarly, we also assumed exactly two peaks in each sub-function separated by exactly $k/2$ bits, but this is merely for convenience. In practice, these restrictions are not necessary but showing this rigorously is beyond the proofs given. Accordingly, the next section provides simple simulation results on this function to illustrate that the result is not limited to these special cases.

5. SIMULATION RESULTS

In this section we use a normal multi-deme Island model GA, with rank-based selection in each deme, and low symmetric migration rates. We use a low per-bit mutation rate and one-point crossover. This serves to show that the few special details of the algorithm treated in theorem 1, in particular, the use of only one receiving deme and the special replacement scheme, are not necessary for solving this problem with a crossover GA.

We also show results of mutation-only methods for comparison. The ability of a GA with crossover to manipulate building blocks is, we would expect, dependent on the assumption of strong genetic linkage – we verify this by comparison with uniform crossover where no bits are linked with any other. And finally, the ability of a GA with crossover to manipulate building blocks is, we would expect, also dependent on the assumption that the genetic linkage corresponds with the epistasis that defines the building block structure, i.e. tight linkage – we verify this by comparison with the use of one-point crossover and shuffled genetic map. In this shuffled version of the problem the overall function is the sum of B sub-functions over B disjoint sets of k variables as before, but these subsets of variables are randomly located in the genotype rather than in contiguous loci.

Problem parameters: B=20 blocks; K=20 bits per block; N=400 bits per genotype; T=2 peaks per block; w_1=1 = value of 1st target string; w_2=10= value of 2nd target string;
t_1=10101010101010101010 = 1st target string;
t_2=11111111111111111111 = 2nd target string.

Algorithms: Multi-deme GA with one-point crossover. Multi-deme GA with uniform crossover. Multi-deme GA with one-point crossover and shuffled genetic map. Random Mutation Hill-Climber (RMHC). Macro Mutation Hill-Climber (MMHC).

The RMHC is a simple mutation hill-climber using a per-bit mutation rate [6]. Mutation rates of $1/L$, $5/L$, $10/L$, $15/L$, $20/L$, and $25/L$ were run. Results for $20/L$ (= k/L), which had the highest fitness on average, are shown. The MMHC is a mutation-based hill-climber that is able to utilise the problem specific knowledge of tight linkage, but not crossover. It picks two inter-local positions at random and randomises the bits in the section of genotype in-between these positions. The MMHC is of notable interest since it solves concatenated trap functions, or other separable functions with tight linkage and small k, efficiently [14].

GA parameters: 100 demes of 10 individuals each; Rank-based selection in each deme; Per-bit mutation rate 1/*N* (probability of assigning a new random bit at each site); Crossover is applied in all reproduction events; Migration rate 0.0004 (i.e. over the whole population, on average one individual in every 2.5 generations is a migrant from another deme).

Results: Fig. 3 shows the average fitness of individuals across the whole population averaged over 30 independent runs. The maximum fitness for these parameters is 20*(10+1/(1+10))) = 201.81. Error bars show +/- 1 standard deviation. All 30 runs of the multi-deme GA with one-point crossover (and tight linkage) find the globally optimal genotype in around 50 generations. None of the other algorithms finds the globally optimal genotype in any of the 30 runs. The results for other mutation rates of the RMHC are for the most part indistinguishable from those of the *k/L* mutation rate shown, but *k/L* has a slightly higher average fitness at the end of the runs than the other rates tested.

These results show that a GA with crossover is able to find the maximum fitness genotypes in this function quite easily, given sufficient population diversity as provided by population subdivision in this case. Given that this function has $2^{\sqrt{N}}=2^{20}$ local optima (only one of which is globally optimal), and each of these is at least $\sqrt{N}/2=10$ bits from the unique global optimum, it is not surprising that these results also show that mutation-based methods fail. The shuffled and uniform crossover GAs do better than the mutation hill-climber because they are able to utilise the diversity provided by the population and they only produce new combination of bits at loci where the parents' bits disagree. Nonetheless, even if useful diversity is maintained (as indicated by the success of the one-point GA) the inability of crossover to bring together good blocks when linkage is either absent (uniform crossover) or not tight (shuffled genetic map) is clearly illustrated.

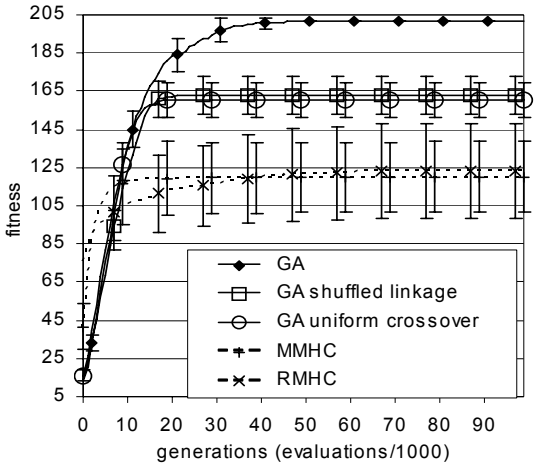

Figure 3. Simulation results using Eq.5, *T*=2.

The proofs in the previous section concern the expected number of evaluations to find the optimal genotype rather than the expected fitness found in a given number of evaluations. But the flat response of all methods after the initial generations makes this distinction quite apparent. The simulations show that the special details of the GA used in theorem 1 are not necessary for a GA to solve this problem efficiently. They show that crossover with tight linkage, as employed in the original intuition of the BBH is necessary for the success of the GA. And they show that mutation-based algorithms, even the MMHC utilising the knowledge of which bits depend on which others (implicit in the assumption of tight linkage), is not sufficient to solve this problem with these parameters. Of course, rigour is provided by the proof of the distinction between crossover and mutation-based methods in the previous section, so the central claims of this paper are independent of the necessarily non-exhaustive simulation study performed here.

The population structure of the GA used in these simulations is still somewhat non-standard however. It seems likely that appropriate fitness scaling and population sizing could provide appropriate population diversity to ensure reliable success in a single-population, generational GA with fitness-proportionate selection – but this has not been investigated thoroughly and the use of a multi-deme GA makes the parameter range where appropriate diversity is maintained easy to find.

6. DISCUSSION AND CONCLUSIONS

In the function we study here, search at two different scales [28] is required to find fit genotypes—local search at the bit scale provided by mutation is required to find the good solutions to blocks, and search in combinations of building blocks by crossover is required to put the blocks together. Although this idea is natural and intuitive it differs fundamentally from the approach of the fully-deceptive subfunctions or the original royal roads where bit-wise fitness gradients within each block are not useful. In our function, there is a significant likelihood that an individual will find the optimum of a given sub-function, and find it quickly, by exploiting bit-wise fitness gradients. But the fact that there is also a significant likelihood that the superior peak is *not* found means that the likelihood of solving all sub-functions by mutation alone becomes very small when the number of blocks is large. This point is important because although Walsh transform analysis [25] and schema disruption analysis [10][24], for example, will agree that our function, like other separable building-block functions, is GA-easy for moderate *k*, such analysis does not confirm that a problem is difficult for mutation-only algorithms.

The critical property of crossover that is exploited in our function is that it allows (semi-)independent search on different segments of the chromosome. Separating selection on schemata in one block-partition from selection on schemata in another block-partition via crossover allows solutions to different blocks to be found independently. Inevitably, we could devise a specialised algorithm that could also solve our problem in polynomial time using this same property of crossover but in another form. For example, an algorithm that uses the knowledge of tight linkage to apply a random-restart hill-climber to each block, one after the other, would suffice. Similarly, a cooperative coevolution approach [20] might also be made to work with some modification. But in fact, the suitability of such algorithms depends on multiple independent trials at solving each building-block, and this is just what crossover in the GA provides.

Explicit mention of two other functions that might be proposed as sub-functions for a discriminating problem is warranted. First, an entirely random subfunction will not provide the distinction we seek. In this case the expected number of local optima in a block is exponential in *k*, and the fraction of the configuration space from which a hill-climber may find the optimal configuration of the block approaches 0 for large *k*. Second, the *NK*-landscape [15] provides a tuneable amount of ruggedness in a landscape and it may be tempting to suggest concatenated *NK*-landscapes [16] (with low epistasis, *K*) might provide a suitable problem. However, since the gross-scale structure of an *NK* fitness

landscape is similar to a unimodal function with noise, the positioning of high-fitness peaks in *NK*-landscapes is not random but clustered. Our result depends on the property that several different runs of a mutation-selection process may reach local optima that are different in fitness and far apart in Hamming space and this is unlikely in *NK*-landscapes.

In general, it is not essential that the sub-functions in a discriminating problem of this type are separable, but it is important that inter-block epistasis does not change which block-configuration is optimal nor significantly change the ability of a mutational hill-climbing process to find that configuration. If this is maintained, inter-block dependencies are permissible. It is certainly not the case that F must be the *sum* of all sub-functions—any monotonic function, e.g. product, will suffice. This is significant since in population genetics 'no epistasis' is defined as multiplicative fitness not additive fitness (it is the *ratio* of a genotype's fitness to population mean fitness that controls its change in frequency under fitness proportionate selection, not the *difference* of fitnesses [29]). Moreover, it is notable that in natural genomes the genes themselves, each composed of thousands of nucleotides, constitute modules of genetic material that are both functionally and physically particulate—forming a very obvious and ubiquitous biological 'building-block' structure [29].

In summary, we provide the first separable building-block function that is easy for crossover and difficult for mutation in a provably rigorous sense (i.e. polynomial versus non-polynomial time to solution respectively). We describe a general function class that maintains the simple intuitions that were sought in the early work on the building block hypothesis, and we explain how prior work on building-block functions omitted crucial features. This helps us to better understand one of the most enduring hypotheses about what GAs are good for.

7. REFERENCES

[1] J. C. Culberson, "Mutation-Crossover Isomorphisms and the Construction of Discriminating Functions", *Evolutionary Computation 2*, 279 (1995).

[2] K. Deb, D. E. Goldberg. "Analyzing Deception in Trap Functions", in D. Whitley, Ed. *Foundations of Genetic Algorithms 2*, (Morgan Kaufmann, San Mateo, CA, 1992) pp. 93–108.

[3] K. Deb, D. E. Goldberg, "Sufficient Conditions for Deceptive and Easy Binary Functions", *Annals of Mathematics and Artificial Intelligence* 10, 385 (1992).

[4] M. Dietzfelbinger, B. Naudts, C. van Hoyweghen, I. Wegener, "The analysis of a recombinative hill-climber on H-IFF", *IEEE–Trans. on Evolutionary Computation* 7, 417 (2002).

[5] S. Droste, T. Jansen, I. Wegener, "On the analysis of the (1+1) evolutionary algorithm", *Theoretical Computer Science*, 276, 51. (2002)

[6] S. Forrest, M. Mitchell, "Relative Building-block fitness and the Building-block Hypothesis", in D. Whitley, Ed., *Foundations of Genetic Algorithms 2*, (Morgan Kaufmann, San Mateo, CA, 1993), pp. 109–126.

[7] S. Forrest, M. Mitchell, "What Makes a Problem Hard for a Genetic Algorithm? Some Anomalous Results and Their Explanation", *Machine Learning* 13(2), 285 (1993).

[8] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning* (Addison-Wesley, Reading, MA, 1989).

[9] G. Harik, D. E. Goldberg, "Learning Linkage", in R. K. Belew, M. D. Vose, Eds. *Foundations of Genetic Algorithms 4* (Morgan Kaufmann, San Francisco, 1997), pp. 247-262.

[10] J. H. Holland, *Adaptation in Natural and Artificial Systems (Ann Arbor*, MI: Univ. Michigan Press, 1975).

[11] J. H. Holland, "Building Blocks, Cohort Genetic Algorithms, and Hyperplane-Defined Functions", *Evolutionary Computation* 8, 373 (2000).

[12] C. Igel, M. Toussaint, "A no-free-lunch theorem for non-uniform distributions of target functions", *Journal of Mathematical Modeling and Algorithms*, 3(4), 313 (2004).

[13] T. Jansen, I. Wegener, "Real royal road functions - where crossover provably is essential", *Discrete Applied Mathematics* 149, 111 (2005).

[14] T. Jones, *Evolutionary Algorithms, Fitness Landscapes and Search*, PhD dissertation, 95-05-048, University of New Mexico, Albuquerque (1995).

[15] S. Kauffman, S. Levin, "Towards a general theory of adaptive walks on rugged landscapes", *J. Theor. Biol*. 128, 11 (1987).

[16] V. Kvasnicka, "An Evolutionary Model of Symbiosis", *Studies in Fuzziness and Soft Computing*, 54, 293 (2000).

[17] M. Mitchell, S. Forrest, J. H. Holland, "The royal road for genetic algorithms: Fitness landscapes and GA performance", in F. J. Varela, P. Bourgine Eds., *Procs. of first European Conference on Artificial Life,* (MIT Press, Cambridge, MA, 1992) pp. 245-254.

[18] R. Motwani, P. Raghavan. *Randomized Algorithms*. (Cambridge Univ. Press, 1995).

[19] M. Pelikan, *Bayesian Optimization Algorithm: From Single Level to Hierarchy*, Ph.D. Dissertation, Dept. of Computer Science at the University of Illinois at Urbana-Champaign (2002).

[20] M. A. Potter, K. A. De Jong, "Cooperative Co-evolution: An Architecture for Evolving Coadapted Subcomponents", *Evolutionary Computation*, 8(1), 1 (2000).

[21] A. Rogers, A. Prügel-Bennett, "A Solvable Model of a Hard Optimisation Problem", in L. Kallel, B.Naudts, A.Rogers, Eds. *Procs. of Theoretical Aspects of Evolutionary Computing* (Springer, Berlin, 2001), pp. 207–221.

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

[23] W. M. Spears, "Crossover or Mutation?", in D. Whitley, Ed. *Foundations of Genetic Algorithms 2*, (Morgan Kaufmann, San Mateo, CA, 1992), pp. 221–237.

[24] M. D. Vose, G. E. Liepins,. "Schema disruption", in *Procs. of the Fourth International Conference on Genetic Algorithms*, (Morgan Kaufmann: San Mateo, 1991), pp. 237–243.

[25] M. D. Vose, A. H. Wright, "The Simple Genetic Algorithm and the Walsh Transform: part I, Theory", *Evolutionary Computation*, 6(3), 253 (1998).

[26] R. A. Watson, "A Simple Two-Module Problem to Exemplify Building-Block Assembly Under Crossover", in X. Yao et al. Eds. *Parallel Problem Solving from Nature - PPSN VIII*, (Springer, Berlin, 2004), pp. 161-171.

[27] R. A. Watson, "Analysis of Recombinative Algorithms on a Non-Separable Building-Block Problem", in W.N. Martin, W.M. Spears, Eds. *Foundation of Genetic Algorithms 6*, (Morgan Kaufmann, San Francisco, 2001) pp. 69–89.

[28] R. A. Watson, "On the Unit of Selection in Sexual Populations", in *Advances in Artificial Life, Eighth European Conference (ECAL 2005)* (Springer, Berlin, 2005), pp. 895-905.

[29] R. A. Watson, D. Weinreich, J. Wakeley, "Sex avoids intragenic local optima that trap asexuals", in prep.

[30] J. B. Wolf, E. D. Brodie III, M. J. Wade, *Epistasis and the Evolutionary Process*. (Oxford University Press: New York, 2000).