Two Remarkable Computational Competencies of
The Simple Genetic Algorithm

Keki Burjorjee
Computer Science Department,
Brandeis University, Waltham 02454.
kekib@cs.brandeis.edu

Abstract

When applied to one-off instances of a wide range of combinatorial optimization problems the Simple Genetic Algorithm (SGA) frequently generates solutions of high fitness. Till date, however, the discovery of one or more seemingly hard problems (consisting of an infinite set of problem instances) that can be solved robustly and scalably by an SGA has eluded researchers. This is unfortunate because the identification of such problems is highly germane to the formulation and verification of explanations for the adaptive capacity of the SGA. The search for such problems has largely been conducted under one or both of the following assumptions: i) that the computational competencies of the SGA arise from its purported proficiency at identifying and composing building blocks, and ii) that the computational competencies of the SGA are to be found in its capacity for global optimization on classes of fitness functions yet to be pinpointed. In this paper we identify two seemingly hard problems that the SGA can solve scalably and robustly. In doing so we showcase computational competencies of the SGA that repudiate both of the assumptions mentioned above. Remarkably the two problems that we identify are closely related to a hard data-mining problem at the cutting edge of genetics having to do with the identification of epistatically interacting quantitative trait loci.

Keywords: genetic algorithms, epistasis, QTL, classification, active learning,
1 Introduction

“Research on the foundations of genetic algorithms aspires to answer two general questions: How do GAs work, and what are they good for?”

—Forrest and Mitchell (1993)

The two questions mentioned above are two sides of the same coin. If one has a sound explanation for the workings of the simple genetic algorithm (SGA) then one should be able to identify at least one seemingly hard computational problem such that one can prove—or at the very least, argue with some degree of rigor—that the problem can be solved robustly and scaleably by some SGA. Vice-versa, the identification of any such computational competencies of the SGA will probably further efforts to understand the remarkable adaptive capabilities of this algorithm.

It is important to clarify that by “problem”, do not mean a single problem instance, but rather an infinite class of problem instances. (Indeed any discussion of the scalability of a SGA is meaningless if were using the word “problem” in the former sense.) One can find numerous reports, within the GA literature, of good, and at times spectacular GA performance on one-off instances of hard problems. These reports are a large part of what make the field of genetic algorithmics interesting and exciting. However conspicuous by its absence is evidence that the SGA can solve some seemingly hard problem scalably and robustly. In other words, no computational competencies of the SGA have been identified so far.

Early efforts to map the computational abilities of the the SGA were strongly influenced by the building block hypothesis (Goldberg, 1989a; Holland, 1992; Mitchell, 1996). It is to some extent a measure of the faith in the explanatory and predictive power of this hypothesis, and to some extent a measure of the faith in the adaptive power of the abstract process described therein, that the earliest of such efforts were undertaken indirectly—by searching for problems that the SGA will not solve robustly and scalably (see for example, Bethke, 1980, Goldberg, 1987, 1989b, Liepins and Vose, 1990, Whitley, 1991, Das and Whitley, 1991). The first direct attempts to identify computational competencies of the SGA were made in the early ’90s by Mitchell, Forrest and Holland (Mitchell et al, 1992, Forrest and Mitchell, 1993, Mitchell et al, 1994), who described their research agenda as follows:

“Rather than studying hard problems on which the GA fails, our
initial approach has been to examine the GA’s behavior on landscapes for which it is likely to perform well. By understanding what features of those landscapes lead to good performance, we hope to better characterize the class of such landscapes.

“One major component of this endeavor is to define the simplest class of landscapes on which the GA performs “as expected”, thus confirming the broad claims of the building-block hypothesis.” (Forrest and Mitchell, 1993)

Forrest, Mitchell, and Holland, in other words, sought to begin by confirming a computational ability of the SGA, albeit an unremarkable one, predicted by the building block hypothesis, and intended to progress from there towards the identification of more impressive computational abilities of the simple genetic algorithm. In pursing this agenda, they espoused two widely held assumptions.

Assumption 1: The computational competencies of the SGA arise from its proficiency at identifying and composing building blocks.

Assumption 2: The computational competencies of the SGA lie in its capacity for efficient global optimization on classes of fitness functions that have yet to be pinpointed.

We will refer to these two assumptions as computational competency assumptions (CCAs).

The results of Forrest, Mitchell and Holland’s first foray into their stated research programme are well known. On a class of fitness functions called Royal Roads that were tailor-made to play to a much vaunted strength of the SGA—its capacity for building block identification and composition—a random mutation hillclimber seemed to need far fewer fitness evaluations to find the global optimum.

The Royal Roads experiments proved to be a watershed in the field of genetic algorithmics. One can identify three different views that emerged as a consequence. In some quarters these experiments sparked or deepened skepticism about the building block hypothesis. In other circles these experiments did not fundamentally undermine the BBH; they merely called

\[1\text{The phrase “global optimization” is often loosely used to refer to the generation of satisficing solutions. In this paper we use this phrase in its strict sense, to refer to the problem of finding a global optimum of an objective function.}\]
into question the efficiency with which the SGA implements the abstract process described therein; these circles are marked by the invention of several new algorithms geared towards the explicit implementation of this process—for example mGA and the fmGA (Goldberg et al., 1989, 1990, 1993; Kargupta, 1995), gemGA, (Kargupta, 1996; Goldberg, 2002); LLGA Harik and Goldberg 1997; Goldberg 2002, BOA (Pelikan et al., 1999; Goldberg 2002); ECGA Harik 1999; FDA Mühlbein and Mahnig 1999), LFDA (Mühlbein and Mahnig 2001), and hBOA (Pelikan and Goldberg 2001). Finally there was the view (see Watson 2006, p72-73) that the Royal Roads experiments do not call into question either the validity of the BBH, or the SGA’s efficiency 2.

These days, the search for computational competencies of the genetic algorithm is largely conducted outside the building block paradigm. In other words Assumption 1 is not espoused. The same however cannot be said about Assumption 2; the aim of many contemporary analyses is the derivation of bounds on the expected or worst case query complexity of some GA when it is applied to some global optimization problem. To the best of our knowledge such studies have not yielded any computational abilities of the SGA worth mentioning. To be sure, results that show the superiority of certain types of GAs (e.g. GAs with crossover) over others (e.g. GAs without crossover) on certain global optimization problems have been derived (e.g. Jansen and Wegener 2002). In general however, given the simplicity of the problems involved, the derived bounds have not been remarkable (one can easily think up other algorithms that solve the involved global optimization problems far more efficiently).

Despite the failure to identify a computational competency of the SGA in the thirty plus years since its introduction, the SGA continues to be widely used, because when applied to combinatorial optimization problems that are poorly understood or known to be NP-hard, the SGA frequently finds

---

2Because the Royal Roads functions are “linear” at the level of the basic building blocks, and therefore do not fully exploit the SGA’s purported capacity for hierarchical composition. Following experiments with the Hierarchical if-and-only-if (HIFF) fitness function, Watson voiced skepticism about the SGA’s ability to efficiently implement the process described in the BBH (Watson 2006, p184), and joined the circle of researchers inventing new algorithms explicitly geared towards the implementation of this process (Watson and Pollack 2000; Watson 2002, 2006). We have voiced our skepticism of the building block hypothesis in another publication currently under review (see www.cs.brandeis.edu/~kekib/jmlrpresumptions.pdf). We will not discuss our critique here, except to say that it strongly undermines the rationale under which these new algorithms are being invented.
solutions of high quality. The identification of one or more computational competencies of the SGA remains likely to help us understand why.

In this paper we identify two problems that seem infeasible given the “mainstream” computational technique for solving them. We then show that both problems can be solved robustly and scalably by a simple genetic algorithm. In other words, we identify two computational competencies of the SGA. Remarkably, the two problems we identify are closely related to a thorny problem at the cutting edge of genetics having to do with the identification of epistatically interacting quantitative trait loci (QTLs).

1.1 On the Identification of Epistatically Interacting QTLs

Consider a phenotypic trait for which there exists a single polymorphic locus\(^1\) such that allele substitutions at this locus result in large changes in the phenotypic trait. Many such traits have been identified (e.g. seed color in pea-plants, eye color in fruit flies, presence of sickle cell anemia, presence of cystic fibrosis). In most cases however changes in a phenotypic trait are more fine-grained, and are influenced by allele substitutions at several polymorphic loci. Such traits are called complex or quantitative, and the loci that influence them are called quantitative trait loci. An important goal of modern genetics is the identification of quantitative trait loci for traits of interest, e.g. the oil content of corn seeds (Hartl 2000, p164), grain weight in rice plants (Xing et al. 2002), and of course, susceptibility to common diseases with complex genetic underpinnings (cancer, diabetes, schizophrenia etc.)

A popular technique for identifying loci that affect quantitative traits is called genome scanning. Given the genomes of a set of individuals and the corresponding values of a particular quantitative trait, genomic loci are visited one by one to determine which loci have a statistically significant effect on the trait when averaged over all other loci. Geneticists distinguish between the main effect of a locus and its interaction effect with other loci. Frankel and Shork (1996) distinguish between the two as follows:

“A main effect is the average effect of a [locus] taken over all other [loci]. Main effects ultimately emerge when one is studying, or mapping, a [locus] either in isolation or without regard to other [loci]. Interaction effects are those attributable to the

\(^1\)A locus with multiple alleles
simultaneous influence of two or more loci. Most contemporary data analysis and statistical modeling strategies for genome scan investigations assess the significance of only the main effects of potential trait loci.”.

Frankel and Shork then eloquently explain why interaction effects have not received much attention, and point out the peril of concentrating solely on main effects:

“There are, of course, many scientific reasons which in part account for this main effect ‘bias’ and these reasons all derive from difficulties surrounding the statistical treatment of epistatic effects . . . Given these difficulties, it is easy to see why epistatic effects have been neglected in favor of main effects in complex trait analysis investigations. Unfortunately, however, there exists the possibility that a locus’s effect might only be detected within a framework that accommodates epistasis. Thus, for example, a locus’s true main effect might be too small to detect with any reasonable statistical power and sample size, and yet it might enter into a critical epistatic effect with a second locus”.

It is easy to see how groups of loci can have interaction effects without exhibiting any main effects. One possible way is shown in Tables 1(a) and 1(b) for two and three interacting loci respectively (for the sake of simplicity we have assumed bi-allelic haploid genomes). In fact for any non-empty set of loci \{A_1, \ldots, A_n\}, and any set of bits \{x_1, \ldots, x_n\}, one can construct a similar table by letting the interaction effect of the two genotypes \(x_1 \ldots x_n\), and \(\bar{x_1} \ldots \bar{x_n}\) be some value \(\delta\), and letting the interaction effect of all other genotypes be \(-2\delta\) (This observation will come in handy in our definition of pivotal functions in section 3).

If loci with interaction effects also have statistically significant main effects then they will be detected by genome wide scans for main effects. Once detected the interaction effects between these loci can be mapped. If, however, loci that interactively influence a quantitative trait have no main effects (or if their main effects are statistically insignificant) then, as Frankel and Shork explain, one will not detect such loci unless one explicitly uses an investigative technique that “accommodates epistasis”.

Main effects are detected by visiting the genomic loci one locus at a time and testing for differentiated marginal effects. Let us call this strategy differentiated marginal effects testing (DMET). To the best of our knowledge,
the only known way to accommodate for epistasis between loci when main effects are absent, is to visit *multi-locus combinations*, and to test each such combination for differentiated (multivariable) marginal effects. We shall call this approach *combinatorial differentiated marginal effects testing*, or combinatorial DMET for short. The computational intractability of combinatorial DMET, even for small combination sizes, is discussed in a recent article by Moore (2008). Moore remarks:

“Identifying the optimal combination of [loci] from an astronomical number of possible combinations is computationally infeasible, especially when the [loci] do not have independent [i.e. main] effects. The following example illustrates the computational magnitude of the problem. Let’s assume that $10^6$ [loci] have been measured. Let’s also assume that 1,000 computational evaluations can be completed in one second on a single processor and that 1,000 processors are available for use. Exhaustively evaluating all of the approximately $4.9 \times 10^{11}$ two-way combinations of [loci] would require approximately 5.7 days. Exhaustively evaluating all of the approximately $1.6 \times 10^{17}$ three-way combinations of [loci] would require 1,929,007 years. This of course assumes a best-case scenario in which the genetic model of interest consists of only two or three important attributes or genetic variations.” [Moore (2008)]

The problem described above (see also Moore, 2003, and Moore and Ritchie, 2004) is a specific instance of the more general problem of identifying interacting attributes in data-mining [Freitas, 2001].

In this paper we will focus on “generative” versions of two specific interacting attribute identification problems in which main effects are entirely absent. By “generative” we mean that the value of any synthesized data point is ascertainable (like in active learning). The first problem, introduced in section 3, can only be solved by a DMET strategy that tests attributes in combinations of two or more. The running time of such a strategy is therefore quadratic in the number of attributes. The second problem, introduced in section 6, can only be solved by a DMET strategy that tests attributes in combinations of four or more; the time required by this strategy is therefore $O(\ell^4)$, where $\ell$ is the number of attributes of an instance of the problem. We will show that both the first and the second problem can be robustly solved by an SGA in time that is linear with respect to the number of attributes $\ell$. Moreover, in both cases, the query complexity of the SGA will be shown
Table 1: The interaction effects and marginal effects of two bi-allelic loci A and B on a quantitative trait are shown in table (a). Note how neither of these loci have a main effect (the marginal effect of each allele of each locus is zero) even though they clearly influence the trait in question. Table (b) shows how three bi-allelic loci A, B, and C can interact epistatically on a trait, yet have no main effect. The reader can check that the the marginal effect of each allele of each locus is zero.

to be constant with respect to \( \ell \).

1.2 Overview

The rest of this paper is organized as follows: in Section 2 we explain the role played by symmetry and symmetry arguments in this paper. In Sections 3–6 we present the two problems mentioned above and show that the SGA can solve both these problems robustly and scalably. We conclude in Section 7 with a discussion of the ramifications of our results for the two computational
competency assumptions mentioned earlier. We also how our findings are relevant to the formulation a theory that explains the adaptive capacity of the SGA.

2 Our Mode of Analysis

This paper is somewhat unusual as foundational studies of genetic algorithms go in that experiments play a primary role in our mode analysis. Experiments are typically used in foundational GA research either to confirm behavior predicted by formal models, or to draw attention to phenomena not predicted by prevailing theories (e.g. Syswerda [1989], Forrest and Mitchell [1993]). The use of experiments as a primary tool of analysis is, however, typically avoided because of the problem of specificity.

One can identify two kinds of specificity. First, as GAs are stochastic processes, any observations about the behavior of a GA during some run are, strictly speaking, only valid for the integer used to seed the random number generator. Of course, one can easily get around this problem by running the GA several times with different seeds. Doing so allows one to build confidence that observed effects are not artifacts of some random seed. In most cases it is straightforward to quantify this confidence using statistics.

The second kind of specificity is more problematic. Strictly speaking, an experimental result is only valid for the parameter value used in the experiment. In practice it may be possible, by changing a parameter while holding all others constant, to glean the relationship between that parameter and some aspect of GA behavior. However if we aim to be rigorous, then the extrapolation involved in this approach is less than ideal. In this paper we circumvent the problem with the second kind of specificity by exploiting the symmetries of the SGAs we construct. Doing so makes it possible to obtain hard quantitative results from a single experiment for an infinite set of parameter values.

Symmetry arguments (Van Fraassen [2003], Jaynes [2007]), while not new to GA research (for a previous instance see Altenberg [1997]), are not common either. Such arguments are more frequently used in physics and chemistry. Indeed according to the theoretical physicist E. T. Jaynes “almost the only known exact results in atomic and nuclear structure are those which we can deduce by symmetry arguments, using the methods of group theory” (Jaynes [2007], p331-332).
One does not, however, need to venture so far afield in order to find an example of a symmetry argument. Consider the following two problems having to do with a fair coin, and a blind coin flipping process.

**Problem 1:** What is the probability of the coin coming up heads when flipped tails-up?

**Problem 2:** What is the probability of the coin coming up tails when flipped heads-up?

Note that an operation that interchanges heads and tails transforms problem 1 into problem 2 and vice versa. If $x$ is the answer to problem 1, then because of the symmetry of these two problems, $x$ is also the answer to problem 2. It does not matter whether the coin is flipped at the bottom of the ocean, or on the surface of the moon.

“The great power of symmetry arguments lies just in the fact that they are not deterred by any amount of complication in the details”, writes Jaynes (2007, p331). Symmetry arguments, in other words, allow one to cut through complications that might hobble other modes of argument (for instance, modes based entirely upon the construction and analysis of formal models). Imagine, for example, having to deduce the conclusion we reached above without the use of symmetry. One would have to model the physics of coin tossing and show, environment by environment, that the answers to both problems will be the same—a daunting prospect.

Jaynes stresses, as do we, that symmetry arguments rely not on ‘equal ignorance’, but on ‘positive knowledge of symmetry’. For instance, going back to the example above, if we are told that the coin is not fair, we would, in a sense, be ‘equally ignorant’ of the answers to problems 1 and 2. Our ‘equal ignorance’ does not of course entail that that the answer to both problems is the same.

2.1 Uniform Crossover

Uniform crossover (Ackley, 1987) was popularized by Syswerda (1989) who showed that this form of crossover can outperform one-point and two-point crossover on problems ranging from simple (e.g. one max) to complex (the travelling salesperson problem). A large amount of evidence for the practical utility of uniform crossover has since accumulated (see for e.g. Rudnick et al., 1994). Syswerda (1989) also showed how any homologous crossover
operation can be represented by a probability distribution over the set of all binary masks. Only in the case of uniform crossover however can the mask of a crossover operation be given by a string of independent identically distributed random binary variables. This feature of uniform crossover is a form of symmetry which we will exploit in our analysis; we call it the i.i.d. property of uniform crossover.

3 Pivotal SGAs

We begin by defining a class of SGAs whose fitness functions such that none of the genomic loci have main effects even though some loci may have epistatic interactions with others. For reasons that will soon become clear we call the members of this class pivotal SGAs.

We will make use of the following notation: for any positive integer \( n \), let \([n]\) denote the set of positive integers \( \{1, \ldots, n\} \). For any \( n \)-tuple \( x \) and any \( i \in [n] \) let \( x_i \) denote the \( i \)th element of \( x \). For any bitstring \( s \) let \( s_i \) denote the \( i \)th symbol of \( s \). For any bit \( x \) let \( \overline{x} \) denote the complement of \( x \).

Let \( N(\mu, \sigma^2) \) denote the normal distribution with mean \( \mu \) and variance \( \sigma^2 \).

A pivotal function, defined below, specifies the length of the genomes of a pivotal SGA and its fitness function.

**Definition 1** Let \( \gamma = (o, \sigma, \delta, \ell, L, V) \) be a 6-tuple such that \( o \) is a positive integer, \( \sigma \) and \( \delta \) are positive real numbers, \( \ell \) is a positive integer greater than \( o \), \( V \) is a \( o \)-tuple of binary values and \( L \) is an \( o \)-tuple of positive integers in \([\ell]\) with strictly increasing values. A pivotal function with descriptor \( \gamma \) is a stochastic function \( f \) which behaves as follows: for any input bitstring \( g \), if \((g_{L_1} = V_1) \land \ldots \land (g_{L_o} = V_o) \) or \((g_{L_1} = \overline{V_1}) \land \ldots \land (g_{L_o} = \overline{V_o}) \) then \( f \) returns a value drawn from \( N(\delta, \sigma^2) \), otherwise \( f \) returns a value drawn from \( N(-\frac{2\delta^2}{o-2}, \sigma^2) \).

We say that an SGA \( G \) is pivotal, with descriptor \( \gamma = (o, \sigma, \delta, \ell, L, V) \), if its fitness function is pivotal, with descriptor \( \gamma \). For any \( i \in [o] \), the locus at index \( L_i \) is said to be pivotal, and \( V_i \) is said to be the pivotal value of that locus. Loci of \( G \) that are not pivotal are said to be non-pivotal. We call \( o \) the order, \( \delta \) the increment, and \( \sigma \) the standard deviation of the pivotal function \( f \).

When a pivotal function is queried with some bitstring \( g \), the distribution from which the answer is drawn pivots upon the values of \( g \) at the
pivotal loci—hence the name *pivotal* function. If we assume that the pivotal function is queried with samples drawn from the uniform distribution over the function’s domain, then, discounting sampling error, i.e. assuming an infinite number of samples, none of the loci will have main effects. The increment parameter $\delta$ determines the strength of the interaction effects between the pivotal loci.

Let $f$ be a pivotal function with descriptor $(\sigma = 3, \delta = 0.18, \ell, L, V)$. The pdfs of $\mathcal{N}(-\frac{2\delta}{\sigma^2}, \sigma^2)$ and $\mathcal{N}(\delta, \sigma^2)$ are shown in Figure 1. Now consider the task of robustly recovering the indices of the pivotal loci using a scanning strategy given only the values of $\sigma, \delta$ and $\ell$, and query access to the function $f$. Because of the stochastic nature of $f$, as long as there is any overlap between the two distributions there will always be some probability of error. The large overlap between the two distributions shown in Figure 1 make the minimization of this error computationally expensive when $\ell$ is large (say $10^6$). But it is the absence of main effects that really makes this problem seem thorny. A DMET strategy that tests loci one-by-one clearly will not work because none of the loci have main effects. Such a strategy only begins to hold promise if loci are tested in combinations of two or more. The number of such combinations however scales at least quadratically with $\ell$.

In the next two sections we will show that a simple GA with uniform crossover and sigma scaling (Mitchell, 1996, p167) will identify the pivotal loci of $f$ relatively robustly (with less than a 0.005 chance of misclassification per locus) in time that is linear in $\ell$, and with some number of queries that is constant with respect to $\ell$.

4 Symmetry Analysis

In this section we use the symmetry of pivotal functions, and the i.i.d. property of uniform crossover to arrive at several useful theoretical results.

For any positive integer $n$ let $\Upsilon_n$ denote the set \{0, $\frac{1}{n}$, $\frac{2}{n}$, $\ldots$, $\frac{n-1}{n}$, 1\}. Let us call the frequency of 1’s and 0’s in a population at some locus $k$ in some generation $t$ the one-frequency and zero-frequency respectively of locus $k$ in generation $t$. For some SGA $G$ with population size $N$, let $1_{(G,i)}^{(t)} : \Upsilon_N \rightarrow [0,1]$ be the probability mass function such that, for any $x \in \Upsilon_N$, $1_{(G,i)}^{(t)}(x)$ is the probability that the one-frequency of locus $i$ after $t$ generations of running $G$ is $x$. Likewise let $0_{(G,i)}^{(t)} : \Upsilon_N \rightarrow [0,1]$ be the probability mass
Probability Density Functions of Normal Distributions

$\mu = 0.18, \sigma = 1$
$\mu = -0.06, \sigma = 1$

Figure 1: The solid black and grey lines shows the pdfs of two normal distributions with standard deviation 1, and means -0.06 and 0.18 respectively.

function such that, for any $x \in \Upsilon_N$, $0^{(t)}_{(G,i)}(x)$ is the probability that the zero-frequency of locus $i$ after $t$ generations of running $G$ is $x$. We call such distributions one and zero frequency distributions. Clearly, for any $x \in \Upsilon_N$, $0^{(t)}_{(G,i)}(x) = 1^{(t)}_{(G,i)}(1 - x)$, and $1^{(t)}_{(G,i)}(x) = 0^{(t)}_{(G,i)}(1 - x)$.

**Proposition 1** Let $G$ be a pivotal SGA. Then for any locus $k$ of $G$, and for any generation $t$

(a) $1^{(t)}_{(G,k)} = o^{(t)}_{(G,k)}$

(b) $E[1^{(t)}_{(G,k)}] = E[0^{(t)}_{(G,k)}] = \frac{1}{2}$

**Argument:** Part (a) follows by consideration of the symmetry between $1^{(t)}_{(G,k)}$ and $0^{(t)}_{(G,k)}$ induced by the fitness function, and the fact that the initial population of any SGA is always drawn from the uniform distribution over the set of bitstrings. Part (b) follows from part (a) and the observation that for any SGA $A$, for any locus $j$ of $A$, and for any generation $t$, $E[1^{(t)}_{(A,i)}] + E[0^{(t)}_{(A,i)}] = 1$. $\Box$
Note that this result applies to an SGA with any pivotal descriptor, any population size, any genome length, any commonly used selection operator (e.g. rank, tournament, fitness proportional etc.), any homologous crossover operator, and any mutation and crossover rates. Imagine having to show all of this without the use of symmetry!

Like homologous crossover, various forms of mutation can be identified with probability distributions over the set of binary masks. We say that a mutation operator is uniform if the mask used in a mutation operation can be represented by a string of independent identically distributed binary random variables. Uniform mutation thus shares the i.i.d. property of uniform crossover. In the interest of brevity, let us call an SGA with uniform crossover and uniform mutation a UGA. The next definition shows how one can derive what we call the standard form of any pivotal UGA.

**Definition 2** Let $U$ be some pivotal UGA with descriptor $(o,\delta,\sigma,\ell,L,V)$, and let $U^*$ be a pivotal UGA with descriptor $(o,\delta,\sigma,o+1,(1,\ldots,o),(1,\ldots,1))$ that is identical to $U$ in all other respects. We call $U^*$ the standard form of $U$.

In words, if the pivotal UGA $U^*$ is the standard form of some pivotal UGA $U$ with order $o$ then the genome length of $U^*$ is $o+1$, the first $o$ of loci of $U^*$ are pivotal, the last locus of $U^*$ is non-pivotal, the pivotal values of $U^*$ are all one, and in all other respects, $U^*$ is identical to $U$. The standard form of any pivotal UGA is clearly unique. Let $U$ be a pivotal UGA with descriptor $(o,\delta,\sigma,\ell,L,V)$. We say that $U$ is basic if the standard form of $U$ is $U^*$. And since the last three elements of the descriptor of $U$ are then derivable from the first three, we write this descriptor as $(o,\delta,\sigma)$.

**Proposition 2** Let $U$ be a pivotal UGA with descriptor $\gamma = (o,\delta,\sigma,\ell,L,V)$, and let $U^*$ be the standard form of $U$. Then, for any generation $t$,

(a) For any pivotal locus $k$, $1_{(U,k)}^{(t)} = 1_{(U^*,1)}^{(t)}$

(b) For any non-pivotal locus $k$, $1_{(U,k)}^{(t)} = 1_{(U^*,o+1)}^{(t)}$

In other words the one frequency distribution of any pivotal locus of $U$ in some generation $t$ is same as the one frequency distribution of the first locus of $U^*$ in generation $t$, and the one frequency distribution of any non-pivotal
locus of $U$ in generation $t$ is the same as the one frequency distribution of
the last locus of $U^*$ in generation $t$.

ARGUMENT: Let $U'$ be a pivotal UGA that is identical to $U$ in all respects
except that its descriptor is $\gamma' = (o, \delta, \sigma, \ell, L, (1, \ldots, 1))$. We shortly present
four claims. Part (a) of proposition 2 follows from claim 1a, claim 2, claim 4 and proposition 1a. Part (b) of the above proposition follows from claim 1b and claim 3.

Claim 1 For any generation $t$, we have the following:

(a) For any $i \in [o]$, if $V_i = 1$, then $1_{(U', L_i)}^{(t)} = 1_{(U, L_i)}^{(t)}$, otherwise $1_{(U', L_i)}^{(t)} = 0_{(U, L_i)}^{(t)}$.

(b) For any generation $t$, and any non-pivotal locus $k$ of $U$ and $U'$,

$1_{(U', k)}^{(t)} = 1_{(U, k)}^{(t)}$

Claim 2 For any generation $t$, and any $i \in [o]$, $1_{(U^*, i)}^{(t)} = 1_{(U', L_i)}^{(t)}$

Claim 3 For any generation $t$, and any non-pivotal locus $k$ of $U$,

$1_{(U^*, o+1)}^{(t)} = 1_{(U', k)}^{(t)}$

Claim 4 For any generation $t$, $1_{(U^*, 1)}^{(t)} = 1_{(U^*, 2)}^{(t)} = \ldots = 1_{(U^*, o)}^{(t)}$

Claim 1 follows from the observation that in any generation the population of $U$ can be “changed into” the population of $U'$ and vice versa by a simple $0 \leftrightarrow 1$ relabeling of all genomic bits at the pivotal loci of $\gamma$ whose corresponding pivotal values are 0.

Claims 2 follows by consideration of the symmetry between loci $L_1, \ldots, L_o$ of $U'$ and loci $1, \ldots, o$ of $U^*$ respectively. Claim 3 follows by consideration of the symmetry between any non-pivotal locus of $U'$ and locus $o + 1$ of $U^*$. These symmetries follow from the the definition of the pivotal functions of $U'$ and $U^*$, and the i.i.d. property of uniform crossover and uniform mutation.

We offer the following observations in order to make these symmetries manifest. Figure 2(a) shows a “vertical view” of a hypothetical population of $U'$ with three pivotal loci (shown in gray). Given the definition of the fitness function of $U'$, it is easy to see that the fitness of any genome depends only
Figure 2: A “vertical view” of a hypothetical population is shown in subfigure (a). Each row is a genome. The shaded columns show the positions of three hypothetical pivotal loci. Given the definition of the fitness function (see text) only the bits in the shaded column matter during selection. Subfigure (b) shows a “vertical view” of a hypothetical crossover operation. Two parents, \( x \) and \( y \) are about to undergo uniform crossover which will yield a child \( z \). The bits of \( z \) will be determined by the values of the random binary variables that comprise the mask \( m \). As crossover is uniform these variables are independent and identically distributed upon the value of that genome’s bits at the pivotal loci. Thus only the bits in the shaded columns of Figure 2(a) matter in determining a genome’s fitness, and by extension its chance of being selected (note that this is true regardless of the selection scheme used). Figure 2(b) shows a “vertical view” of a hypothetical uniform crossover operation in \( U' \). Two genomes, \( x \) and \( y \), have been selected for uniform crossover. The crossover mask \( m \) is represented as a string of random binary variables. The values of these variables determines the bits of the child \( z \). Because crossover is uniform, the random variables in \( m \) are independent and identically distributed.
Claim 4 “follows” from the symmetry that exists between the first \( o \) loci of \( U^* \). Like the symmetries used in claims 2 and 3, this symmetry follows from the i.i.d. property of uniform crossover and uniform mutation, and the definitions of the stochastic fitness function used by \( U' \) and \( U^* \).

By proposition 2(a), for any pivotal locus \( k \) of \( U \), drawing monte-carlo samples from \( 1^{(t)}_{(U^*,1)} \) is equivalent to drawing monte-carlo samples from \( 1^{(t)}_{(U,k)} \). And by proposition 2(b), for any non-pivotal locus \( k \) of \( U \), drawing monte-carlo samples from \( 1^{(t)}_{(U^*,o+1)} \) is equivalent to drawing monte-carlo samples from \( 1^{(t)}_{(U,k)} \).

5 Experiment 1

In the experiment described in this section we used a fitness scaling technique called sigma scaling (Mitchell, 1996, p167), wherein, for any generation, the fitness of each genome \( x \) is adjusted (before selection operates upon the population) as follows:

\[
f'_{\alpha}(x) = \begin{cases} 
\max \left( 1 + \frac{f(x) - \bar{f}}{\alpha \sigma} , 0 \right) & \text{if } \sigma \neq 0 \\
1,0 & \text{if } \sigma = 0
\end{cases}
\]

where \( f'(x) \) is the adjusted fitness of \( x \), \( f(x) \) is the unadjusted fitness of \( x \), \( \bar{f} \) is the unadjusted fitness of the population and \( \sigma \) is the standard deviation of the unadjusted fitness values of the population. We call the parameter \( \alpha \in (0, \infty) \) the sigma scaling coefficient. The reader can check that selection pressure will vary inversely with this coefficient. Because we scale fitness as described here we are free to use fitness functions that may return negative values. The adjusted fitness will always be non-negative. The crossover rate was 1, and selection was fitness proportional. We denote the population size by \( N \), and the per bit mutation probability (also called the mutation rate) by \( \rho \).

We executed 3000 runs of a basic pivotal UGA (\( N = 2000, \rho = 0.003, \alpha = 1 \)) with descriptor \( (o = 3, \delta = 0.24, \sigma = 1) \). Let us call this UGA \( U^* \). Each run of \( U^* \) consisted of 200 generations. Figure 3 shows the one-frequency dynamics of the first and last loci in each run. In all 3000 runs the first locus went to fixation by the 160th generation, whereas the

\footnote{We use the term ‘fixation’ loosely. Clearly, as long as the mutation rate is non zero, no locus can ever be said to go to fixation in the strict sense of the word.}
Algorithm 1: ClassifyLoci

```
pivotalLoci={}
Run U for 160 generations
for i = 1 to ℓ do
    x = one-frequency of the i\(^{th}\) locus of U after 160 generations
    if x < 0.07 \text{ or } x > 0.93 then
        pivotalLoci = pivotalLoci \cup \{i\}
    end
end
return pivotalLoci
```

one-frequency of the last locus in the 160th generation was always between 0.93 and 0.07.

5.1 A Computational Strength of the UGA

Given the conclusions of our symmetry analysis of pivotal SGAs, the result shown in Figure 3 provides us with a window into the frequency dynamics of all pivotal and non-pivotal loci of any pivotal UGA with standard form \(U^*\). We infer that the pivotal loci of such a UGA will tend to go to fixation by the 160th generation. We also infer that drift in the non-pivotal loci, while significant will tend not to be not extreme.

Let \(U\) be some pivotal UGA with bitstrings of length \(ℓ\) such that the standard form of \(U\) is \(U^*\). Now, consider Algorithm 1. The results of our experiment suggest that ClassifyLoci will classify each locus of \(U\) fairly accurately. Let us quantify this accuracy. Note that in all 3000 runs of \(U^*\), by the end of the 160th generation the one-frequency of the first locus was outside the interval \([0.07, 0.93]\), and the one-frequency of the last locus was inside this interval. Let \(a\) be the probability that the one-frequency of the first locus of \(U^*\) will be inside \([0.07, 0.93]\) at the end of the 160th generation of \(U^*\). Let \(H_0\) be the hypothesis that \(a \geq 0.005\). If \(H_0\) is true then the probability that the one-frequency of the first locus will be outside \([0.07, 0.93]\) at the end of the 160th generation in each of 3000 runs (as observed in the above experiment) is less than \((1 - 0.005)^{3000} < 3 \times 10^{-7}\). Therefore we reject \(H_0\) at the \(3 \times 10^{-7}\) level of significance. Now consider the hypothesis that with probability greater than or equal to 0.005 the one-frequency of the last locus of \(U^*\) will be outside the interval \([0.07, 0.93]\) at the end of 160 generations. Using very similar reasoning we reject this hypothesis at
Figure 3: The top and bottom plots show the one-frequency dynamics of
the first and last locus respectively in each of 3000 runs of a basic pivotal
UGA ($N = 2000, \rho = 0.003$) with descriptor ($\alpha = 3, \delta = 0.18, \sigma = 1$). The
crossover rate of was 1, and sigma scaling was used ($\alpha = 1$). The horizontal
black lines mark the frequencies 0.93 and 0.07.
the $3 \times 10^{-7}$ level of significance. Thus, with probability of error less than three in ten million, the following statement is true: for any positive integer $k \in [\ell]$, there is less than a 0.005 probability that CLASSIFYLOCIs makes an error in its classification of locus $k$ as pivotal or non-pivotal.

Note that $\ell$, may be any positive integer greater than 3. There are $\binom{\ell}{3} \in O(\ell^3)$ possible configurations of the three pivotal indices. Remarkably, CLASSIFYLOCIs achieves the level of robustness mentioned above ($p < 0.005$ per locus) in time that is linear in $\ell$, after making some number of fitness evaluations that is constant with respect to $\ell$.

5.2 Discussion

In order to explain the the behavior of $U$ we reveal an experimental finding that we did not previously mention because it is not relevant to the computational strength of $U$ that we have showcased. This finding is however relevant when it comes to understanding the behavior of $U$.

In order to clearly describe the finding we develop the following terminology. A site array is a tuple of the indices of distinct loci, and is denoted using angle brackets (e.g. $\langle 2, 15, 3 \rangle$). The number of elements in some site-array $\Gamma$ is called its order, and is denoted by $o(\Gamma)$. Given a site array $\Gamma$, a genotype of $\Gamma$ is a binary string of length $o(\Gamma)$. Given some site array $\Gamma$, some genotype $\gamma$ of $\gamma$, and some bi-allelic (bitstring) genome $x$, we say that $x$ is of type $\gamma$, or that it belongs to $\gamma$, if $x_{\Gamma_1} = \gamma_1 \land \ldots \land x_{\Gamma_{o(\Gamma)}} = \gamma_{o(\Gamma)}$. When convenient we will treat $\gamma$ as the set of all the genomes that belong to it. The sense in which we refer to a genotype (as a string, or as a set) will be clear from the context.

We found that 160 generations into each run of $U^*$, either the genotype 000, or the genotype 111, of the site-array $\langle 1, 2, 3 \rangle$, dominated the population. The average fraction of the population that belonged to the dominant genotype at the end of 160 generations was 0.947 (with standard deviation 0.013).

We now explain the behavior of $U$ that we inferred above. The frequency dynamics of the non-pivotal loci of $U$ is explained by the notion of drift. To understand the frequency dynamics of these loci it helps to go back to the “vertical views” presented in Figure 2. Observe that discounting the effect of sampling error, only selection and mutation have an effect on the composition of the bit-pool of each locus. Crucially, crossover does not

---

5Changes in the one and zero frequencies of a locus can be visualized as changes in the
change the composition of these bit-pools. Now, let $x_1, x_2,$ and $x_3$ denote the indices of the pivotal loci of $U$, and, without loss of generality, suppose that the pivotal values of the first, second, and third pivotal loci are 0, 1, and 0 respectively. Consider the frequency dynamics of the genotype 010 of the site-array $\langle x_1, x_2, x_3 \rangle$. The probability of generating genomes of type 010 in some generation is highly (though not completely) dependent upon the composition of the bit-pools of the pivotal loci in the previous generation. A genome of type 010, once generated, will tend to be preferentially selected over all other “sibling” genotypes except 101. Thus, regardless of what happens during crossover, once generated, a genome of type 010 will tend to increase the frequency of 0’s, 1’s, and 0’s in the bit-pools of the first, second, and third pivotal loci respectively. This makes conditions more favorable for the generation of genomes of type 010 in future generations. Of course, the same argument applies to genomes of type 101. Now, given that the alleles 1 and 0 are, in a sense, “rivals” of each other at each locus, the genotypes 010 and 101 “compete” for dominance of the bit-pools of each of the pivotal loci. One of these genotypes eventually manages to gain an edge over the other, “pulling” the composition of the bit-pools of all three pivotal loci far enough in its favor that a self-reinforcing loop that heavily favors the future generation of the victorious genotype then ensues.

In light of this analysis, one can conclude that the building block hypothesis takes an overly-grim view of the disruption of fit low-order schemata with high defining-lengths. This view misses the fact that the “debris” from the disruption of such a schema changes the composition of the bit-pools of the loci of the defined bits of that schema in a way that favors the future generation of the genomes that belong to the schema.

It also merits mentioning that the computational strength of $U$ showcased here would be invisible to analytic approaches in which an infinite population is assumed (e.g. Vose and Liepins 1991; Burjorjee 2007). This is because without some kind of symmetry breaking, the one and zero frequencies of the pivotal loci will not depart from 1/2. The role of symmetry breaking is performed here by sampling error which is absent from infinite population models of genetic algorithms.

composition of a pool of bits. This metaphor is especially useful in conjunction with the “vertical view” of a population presented in figure 2(a) each column is a bit-pool
6 Type II Pivotal SGAs

If the order \( o \) of a pivotal function is greater than one, then no individual locus will have differentiated marginal effects. Some combinations of two loci, however, will have differentiated (multilocus) marginal effects—specifically those combinations with two pivotal loci. Thus for any \( o > 1 \), the pivotal loci can be robustly identified using a combinatorial DMET strategy in time that is quadratic in the length of the input bitstrings.\(^6\)

From here onwards let us refer to previously defined pivotal SGAs and pivotal functions as type I pivotal SGAs, and type I pivotal functions respectively. In this section we define a new class of SGAs—type II pivotal SGAs—which use a new class of stochastic functions—type II pivotal functions—to calculate fitness. Type II pivotal functions are expressly defined so that for any even order \( o \), and any positive integer \( m < o \), no combination of \( m \) loci will have differentiated marginal effects!

Let \( \oplus \) denote the exclusive-or operator (also the addition modulo 2 operator). Type II pivotal functions are defined as follows:

**Definition 3** Let \( \gamma = (o, \sigma, \delta, \ell, L) \) be a 5-tuple such that \( o \) is a positive even integer, \( \sigma \) and \( \delta \) are positive real numbers, \( \ell \) is a positive integer greater than \( o \), and \( L \) is an \( o \)-tuple of positive integers in \([\ell]\) with strictly increasing values. A type II pivotal function with descriptor \( \gamma \) is a stochastic function \( f \) which behaves as follows: for any input bitstring \( g \), if \( g_{L_1} \oplus \ldots \oplus g_{L_o} = 1 \) then \( f \) returns a value drawn from \( \mathcal{N}(\delta, \sigma^2) \), otherwise \( f \) returns a value drawn from \( \mathcal{N}(-\delta, \sigma^2) \).

We say that an SGA \( G \) is type II pivotal, with descriptor \( \gamma = (o, \sigma, \delta, \ell, L) \), if its fitness function \( f \) is type II pivotal, with descriptor \( \gamma \). For any \( i \in [o] \), the locus at index \( L_i \) is said to be pivotal. Loci of \( G \) that are not pivotal are said to be non-pivotal. We call \( o \) the order, \( \delta \) the increment, and \( \sigma \) the standard deviation of the pivotal function \( f \). Note how the descriptor of type II pivotal functions differs from the descriptor of type I pivotal functions: \( o \) is even, and no pivotal values associated with the pivotal loci.

Consider a type II pivotal function \( f \) with order 4. If we assume that \( f \) is queried with an infinite number of uniformly drawn samples, then Table

\(^6\)Of course, as \( o \) increases, the constant of the time complexity of this approach will increase very quickly. Nevertheless, the time complexity is quadratic in \( \ell \) for any fixed value of \( o \).
shows the interaction effects between the four pivotal loci. The reader can check that interchanging the alleles 1 and 0 leaves the interaction effect of each four-allele combination unchanged. This is a form of symmetry that we will exploit. Note also that any two columns can be interchanged without affecting the interaction effect of each four-allele combination. This is another form of symmetry that we will make use of; it arises from the associativity and commutativity of the $\oplus$ operator. It is easily seen that these two symmetries will be present regardless of the order and increment of $f$. It should also be clear that the marginal effects of any combination of $m$ loci, will not be differentiated as long as $m$ is less than the order $o$. Thus the time complexity of a combinatorial DMET strategy that robustly identifies the pivotal loci must of needs be $O(\ell^o)$.

Let $f$ be a type II pivotal function with descriptor $(o = 4, \delta = 0.15, \sigma = 1, \ell, L)$. In the next section we will show that a simple GA with uniform crossover and sigma scaling can identify the pivotal loci of $f$ relatively robustly (with less than a 0.005 chance of misclassification per locus) in time that is linear in $\ell$, and with some number of queries that is constant with respect to $\ell$. Our approach is almost identical to the approach we took in Sections 3–5, where we showed a similar result for a class of pivotal functions of type I.

6.1 Symmetry Analysis

We define the standard form of a type II pivotal UGA as follows:

Definition 4 Let $W$ be some type II pivotal UGA with descriptor $(o, \delta, \sigma, \ell, L)$, and let $W^*$ be a type II pivotal UGA with descriptor $(o, \delta, \sigma, o+1, (1, \ldots, o))$ that is identical to $W$ in all other respects. We call $W^*$ the standard form of $W$.

Let $W$ be a type II pivotal UGA with descriptor $(o, \delta, \sigma, \ell, L, V)$. We say that $W$ is basic if the standard form of $W$ is $W$. And since the last two elements of the descriptor of $W$ are then derivable from the first three, we write this descriptor as $(o, \delta, \sigma)$.

The next proposition is almost identical to Proposition 2.

Proposition 3 Let $W$ be a type II pivotal UGA with descriptor $\gamma = (o, \delta, \sigma, \ell, L)$, and let $W^*$ be the standard form of $W$. Then, for any generation $t$,
Table 2: Let $f$ be a type II pivotal function with descriptor $(o = 4, \delta, \sigma, \ell, L)$. The table above shows the interaction effects of the alleles of the pivotal loci when $f$ is queried with an infinite number of samples drawn uniformly from its domain.

|   |   |   |   | Interaction Effect |
|---|---|---|---|---------------------|
| 0 | 0 | 0 | 0 | $-\delta$ |
| 0 | 0 | 0 | 1 | $+\delta$ |
| 0 | 0 | 1 | 0 | $+\delta$ |
| 0 | 0 | 1 | 1 | $-\delta$ |
| 0 | 1 | 0 | 0 | $+\delta$ |
| 0 | 1 | 0 | 1 | $-\delta$ |
| 0 | 1 | 1 | 0 | $-\delta$ |
| 0 | 1 | 1 | 1 | $+\delta$ |
| 1 | 0 | 0 | 0 | $+\delta$ |
| 1 | 0 | 0 | 1 | $-\delta$ |
| 1 | 0 | 1 | 0 | $-\delta$ |
| 1 | 0 | 1 | 1 | $+\delta$ |
| 1 | 1 | 0 | 0 | $-\delta$ |
| 1 | 1 | 0 | 1 | $+\delta$ |
| 1 | 1 | 1 | 0 | $+\delta$ |
| 1 | 1 | 1 | 1 | $-\delta^2$ |

(a) For any pivotal locus $k$, $1_{(W,k)}^{(t)} = 1_{(W^*,1)}^{(t)}$

(b) For any non-pivotal locus $k$, $1_{(W,k)}^{(t)} = 1_{(W^*,o+1)}^{(t)}$

We omit the argument for this proposition, on the assumption that it will be clear to readers who have digested the argument for Proposition 2.

By proposition (a), for any pivotal locus $k$ of $W$, drawing monte-carlo samples from $1_{(W^*,1)}^{(t)}$ is equivalent to drawing monte-carlo samples from $1_{(W,k)}^{(t)}$. And by proposition (b), for any non-pivotal locus $k$ of $W$, drawing monte-carlo samples from $1_{(W^*,o+1)}^{(t)}$ is equivalent to drawing monte-carlo samples from $1_{(W,k)}^{(t)}$. 
Algorithm 2: \textsc{ClassifyLoci2}

\begin{algorithm}
\begin{algorithmic}
\State pivotalLoci=\{
\EndState
\State Run \textit{W} for 1600 generations
\For{$i = 1 \text{ to } \ell$}
\State $x$ = one-frequency of the $i^{th}$ locus of \textit{W} after 1600 generations
\If{$x < 0.06 \text{ or } x > 0.94$}
\State pivotalLoci = pivotalLoci $\cup \{i\}$
\EndIf
\EndFor
\State return pivotalLoci
\end{algorithmic}
\end{algorithm}

6.2 Experiment 2

We executed 3000 runs of a basic type II pivotal UGA ($N = 2500$, $\rho = 0.003$, $\alpha = 1$) with descriptor ($\sigma = 4$, $\delta = 0.16$, $\sigma = 1$). Let us call this UGA \textit{W*}. Each run of \textit{W*} consisted of 2000 generations. Figure 4 shows the one-frequency dynamics of the first and last loci in each run. In all 3000 runs the first locus went to fixation by generation 1600, whereas the one-frequency of the last locus in generation 1600 was always between 0.94 and 0.06.

Furthermore, we found that 1600 generations into each run of \textit{W*}, some genotype \textit{wxyz}, such that $w \oplus x \oplus y \oplus z = 1$, of the site-array \textit{(1,2,3,4)}, dominated the population. On average the fraction of the population that belonged to the dominant genotype in generation 1600 was 0.943 (with a standard deviation of 0.012).

Let \textit{W} be a UGA with genomes of length $\ell$ such that the standard form of \textit{W} is \textit{W*}. The conclusions of our symmetry analysis of type II pivotal SGAs, and the result shown in Figure 4 provides us with a window into the frequency dynamics of all pivotal and non-pivotal loci of \textit{W}.

Consider Algorithm 2. Based on arguments that are almost identical to the ones in section 5.1 we can conclude that with probability of error less than three in ten million the following statement is true: for any positive integer $k \in [\ell]$, there is less than a 0.005 probability that \textsc{ClassifyLoci2} makes an error in its classification of locus $k$ as pivotal or non-pivotal.

There are $\binom{\ell}{4} \in O(\ell^4)$ possible configurations of the pivotal indices. Remarkably, like \textsc{ClassifyLoci}, \textsc{ClassifyLoci2} achieves the level of robustness mentioned above ($p < 0.005$ per locus) in time that is linear in $\ell$, after
making some number of fitness evaluations that is constant with respect to \( \ell \).

Note that although the interaction effects of the pivotal loci in Experiment 1 were less dispersed than the interaction effects of the pivotal loci in Experiment 2, Algorithm 2 requires ten times as many fitness evaluations to achieve a similar level of robustness. This, in our opinion, reflect the greater difficulty of type II pivotal functions.

7 Conclusion

The identification of computational competencies of the SGA is inextricably linked with the formulation and verification of theories that explain the SGA’s remarkable capacity for adaptation. Till date the search for such competencies has been conducted under one or both of the computational competency assumptions mentioned in the introduction. If nothing else, the computational competencies of the SGA presented in this paper—the first two to be discovered—serve as counterexamples to these assumptions; the identification and composition of building blocks plays no part in either computational competency, and neither computational competency has anything to do with global optimization.

The utility of the experiments and analysis presented however extends beyond the repudiation of the aforementioned assumptions. What makes the computational abilities showcased in this paper particularly noteworthy is the domain in which they lie. A central explanandum of the field of genetic algorithmics, and indeed one might say, of the field of evolutionary biology, is the persistence of adaptation in sexually evolving populations despite the ubiquity of epistatic interactions between genomic loci. That the SGA can, in particular cases, robustly and scalably “identify” small numbers of unlinked epistatically interacting loci with no main effects, and moreover that the SGA does so by sending a genotype with above average fitness to fixation, is almost certain to be material to theories about the remarkable adaptive capacity of evolution. In our forthcoming dissertation we present an explanation for the adaptive capacity of the UGA. This explanation is based largely upon the findings presented here.
Figure 4: The top and bottom plots show the one-frequency dynamics of
the first and last locus respectively in each of 3000 runs of the basic type II
pivotal UGA $W^\ast$ $(N = 2500, \rho = 0.003)$ with descriptor $(\phi = 3, \delta = 0.16, \sigma = 1)$. The crossover rate of was 1, and sigma scaling was used $(\alpha = 1)$. The
horizontal black lines mark the frequencies 0.94 and 0.06.
References

D.H. Ackley. *A connectionist machine for genetic hillclimbing*. Kluwer Academic Publishers, 1987.

L. Altenberg. Fitness distance correlation analysis: An instructive counterexample. *Proceedings of the Seventh International Conference on Genetic Algorithms*, pages 57–64, 1997.

A. D. Bethke. *Genetic Algorithms as Function Optimizers*. PhD thesis, University of Michigan, Ann Arbor, 1980.

Keki Burjorjee. Sufficient conditions for coarse-graining evolutionary dynamics. In *Foundations of Genetic Algorithms 9 (FOGA IX)*, 2007.

R. Das and D. Whitley. The only challenging problems are deceptive: Global search by solving order-1 hyperplanes. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 166–173. Morgan Kaufmann Publishers, 1991.

Stephanie Forrest and Melanie Mitchell. Relative building-block fitness and the building-block hypothesis. In L. Darrell Whitley, editor, *Foundations of Genetic Algorithms 2*, pages 109–126, San Mateo, CA, 1993. Morgan Kaufmann.

W. N. Frankel and N. J. Schork. Who’s afraid of epistasis. *Nature Genetics*, 14:371–373, 1996.

A.A. Freitas. Understanding the Crucial Role of Attribute Interaction in Data Mining. *Artificial Intelligence Review*, 16(3):177–199, 2001.

D. E. Goldberg, B. Korb, and K. Deb. Messy genetic algorithms: Motivation, analysis, and first results. *Complex Systems*, 3:493–530, 1989.

David E. Goldberg, *Genetic Algorithms in Search, Optimization & Machine Learning*. Addison-Wesley, Reading, MA, 1989a.

David E. Goldberg. *The Design Of Innovation*. Kluwer Academic Publishers, 2002.

David E. Goldberg, Kalyanmoy Deb, and Bradley Korb. Messy genetic algorithms revisited: Studies in mixed size and scale. *Complex Systems*, 4(4):415–444, 1990.
David E. Goldberg, Kalyanmoy Deb, Hillol Kargupta, and Georges Harik. Rapid, accurate optimization of difficult problems using fast messy genetic algorithms. pages 56–64, 1993.

D.E. Goldberg. Simple genetic algorithms and the minimal, deceptive problem. Genetic Algorithms and Simulated Annealing, 74, 1987.

D.E. Goldberg. Genetic algorithms and Walsh functions: Part II, deception and its analysis. Complex Systems, 3(2):153–171, 1989b.

G. Harik. Linkage learning via probabilistic modeling in the ecga, 1999.

Georges R. Harik and David E. Goldberg. Learning linkage. In Richard K. Belew and Michael D. Vose, editors, Foundations of Genetic Algorithms 4, pages 247–262, San Francisco, CA, 1997. Morgan Kaufmann.

Daniel L. Hartl. A Primer Of Population Genetics. Harvard University Press, 2000.

John H. Holland. Genetic algorithms. Scientific American, 267, 1992.

Jansen and Wegener. The analysis of evolutionary algorithms – A proof that crossover really can help. ALGRTHMICA: Algorithmica, 34, 2002.

E.T. Jaynes. Probability Theory: The Logic of Science. Cambridge University Press, 2007.

Hillol Kargupta. The gene expression messy genetic algorithm. pages 814–819, 1996.

Hillol Kargupta. SEARCH, Polynomial Complexity, and the Fast Messy Genetic Algorithm. PhD thesis, University of Illinois at Urbana-Champaign, 1995.

GE Liepins and MD Vose. Deceptiveness and genetic algorithm dynamics. CONF-9007175-1, Oak Ridge National Lab., TN (USA); Tennessee Univ., Knoxville, TN (USA), 1990.

M. Mitchell, J.H. Holland, and S. Forrest. When Will a Genetic Algorithm Outperform Hill Climbing? Advances in Neural Information Processing Systems, pages 51–51, 1994.

Melanie Mitchell. An Introduction to Genetic Algorithms. The MIT Press, Cambridge, MA, 1996.
Melanie Mitchell, Stephanie Forrest, and John H. Holland. The royal road for genetic algorithms: Fitness landscapes and GA performance. In F. J. Varela and P. Bourgine, editors, Proc. of the First European Conference on Artificial Life, pages 245–254, Cambridge, MA, 1992. MIT Press.

Jason Moore. The critical need for computation in human genetics. http://www.reviews.com/hottopic/hottopic_essay_07.cfm, 2008.

Jason H. Moore and M.D. Ritchie. The challenges of whole-genome approaches to common diseases. Journal of the American Medical Association, 291:1642–1643, 2004.

J.H. Moore. The Ubiquitous Nature of Epistasis in Determining Susceptibility to Common Human Diseases. Hum Hered, 56:73–82, 2003.

H. Mühlenbein and T. Mahnig. Evolutionary algorithms: From recombination to search distributions. In Leila Kallel, Bart Naudts, and Alex Rogers, editors, Theoretical Aspects of Evolutionary Computing, pages 135–173, Berlin, 2001. Springer.

Heinz Mühlenbein and Thilo Mahnig. The factorized distribution algorithm for additively decompressed functions. In 1999 Congress on Evolutionary Computation, pages 752–759, Piscataway, NJ, 1999. IEEE Service Center.

Martin Pelikan and David E. Goldberg. Hierarchical bayesian optimization algorithm = bayesian optimization algorithm + niching + local structures. In Optimization by Building and Using Probabilistic Models (OBUPM) 2001, pages 217–221, San Francisco, California, USA, 7 July 2001.

Martin Pelikan, David E. Goldberg, and Erick Cantú-Paz. BOA: The Bayesian optimization algorithm. In Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99, volume I, pages 525–532. Morgan Kaufmann Publishers, San Fransisco, CA, 13-17 July 1999.

EM Rudnick, JG Holm, DG Saab, and JH Patel. Application of simple genetic algorithms to sequential circuit test generation. Proceedings of the European Design and Test Conference, pages 40–45, 1994.

G. Syswerda. Uniform crossover in genetic algorithms. In J. D. Schaffer, editor, Proceeding of the Third International Conference on Genetic Algorithms. Morgan Kaufmann, 1989.

Bas C. Van Fraassen. Laws and Symmetry. Clarendon Press, 2003.
Michael D. Vose and Gunar E. Liepins. Punctuated equilibria in genetic search. *Complex Systems*, 5(1):31–44, February 1991.

R.A. Watson and J.B. Pollack. Symbiotic Combination as an Alternative to Sexual Recombination in Genetic Algorithms. In *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, pages 425–434. Springer-Verlag London, UK, 2000.

Richard A. Watson. *Compositional Evolution: Interdisciplinary Investigations in Evolvability, Modularity, and Symbiosis*. PhD thesis, April 01 2002.

Richard A. Watson. *Compositional Evolution: The Impact of Sex, Symbiosis and Modularity on the Gradualist Framework of Evolution*. The MIT Press, 2006.

L.D. Whitley. Fundamental principles of deception in genetic search. *Foundations of Genetic Algorithms*, 1:221–241, 1991.

Y. Xing, Y. Tan, J. Hua, X. Sun, C. Xu, and Q. Zhang. Characterization of the main effects, epistatic effects and their environmental interactions of QTLs on the genetic basis of yield traits in rice. *TAG Theoretical and Applied Genetics*, 105(2):248–257, 2002.