

Two Remarkable Computational Competencies of The Simple Genetic Algorithm

Keki M. Burjorjee
 Computer Science Department
 Brandeis University
 Waltham, MA 02454
 kekib@cs.brandeis.edu
 May 28, 2009

Abstract—Since the inception of genetic algorithmics the identification of computational efficiencies of the simple genetic algorithm (SGA) has been an important goal. In this paper we distinguish between a *computational competency* of the SGA—an efficient, but narrow computational ability—and a *computational proficiency* of the SGA—a computational ability that is both efficient and broad. Till date, attempts to deduce a computational proficiency of the SGA have been unsuccessful. It may, however, be possible to infer a computational proficiency of the SGA from a set of related computational *competencies* that have been deduced. With this in mind we deduce two computational competencies of the SGA. These competencies, when considered together, point toward a remarkable computational proficiency of the SGA. This proficiency is pertinent to a general problem that is closely related to a well-known data-mining problem at the cutting edge of computational genetics.

Index Terms—genetic algorithms, theory, symmetry, epistasis, data-mining, active learning, QTL

I. INTRODUCTION

When applied to combinatorial optimization problems that are poorly understood, or known to be NP-Hard, simple genetic algorithms (SGAs) frequently evolve usable solutions after evaluating a relatively small number of samples. At a general level it seems reasonable to presume that SGAs owe their adaptive prowess to a capacity for performing one or more kinds of computation relatively efficiently, i.e. robustly and scalably relative to other known algorithms. The identification of such *computational efficiencies*—the source of the remarkable adaptive capacity of SGAs—has been a goal since the inception of the field of genetic algorithmics.

The early genetic algorithmics literature (pre 1990s) is marked by spirited debates about the purported efficiency with which SGAs “process” schemata (for a review see [18, p119-127], [7, p114-119],[27, p74-78]).

Claims about the computational efficiency of SGAs made during this period were supported almost entirely by theoretical arguments, or to be more precise, *quasi-theoretical* arguments—though these arguments made use of mathematics, i.e. they were deductive in parts, the amount of *a priori* and *a posteriori* speculation involved was substantial. These claims were rarely, if ever, backed up by empirical support. To the best of our knowledge no bold claim about the computational efficiency of the SGA made during this period has been empirically supported in a convincing fashion.

Instead, when empirical tests *were* conducted, a large gap was discovered between the claimed efficiency of the SGA, and its actual performance. The Royal Roads experiments [19], [8] in particular proved to be a watershed in the history of genetic algorithmics. On a class of fitness functions called *Royal Roads* that were tailor-made to play to a much vaunted computational efficiency of the genetic algorithm—its capacity for building block identification and composition—a random mutation hill-climber seemed to need far fewer fitness evaluations to find the global optimum [8], [17]. Following the publication of these results, many researchers declared the SGA *inefficient*, and began inventing algorithms geared towards the explicit implementation of the abstract process described in the building block hypothesis [12, p 55] [14], [25], [26].

The publication of the Royal Roads experiments was the first of two major developments that seem to have cooled the search for computational efficiencies of the SGA. The second development was the publication of the no free lunch (NFL) theorems for optimization [31]. After deriving one of the the most general NFL theorems obtained till date, Igel and Toussaint [15] conclude that in all likelihood NFL does *not* apply to the *practice* of black-box optimization. This conclusion certainly seems to be borne out by the empirical record; in the *practice* of

black-box optimization, free lunches seem to be aplenty. Unfortunately, for much too long, the NFL theorem has subliminally, if not overtly, been regarded as “proof” that the opposite is true. This belief entails that when an SGA outperforms random search (or for that matter, pick-the-worst-neighbor search) on some black-box optimization problem in practice, it does so because of a fortuitous pairing between problem and algorithm. If this is so, then any advantage over random search that accrues when a genetic algorithm is used for black-box optimization in practice can be ascribed to fortune rather than to some innate computational efficiency of the genetic algorithm.

Through both of the aforementioned developments, the simple genetic algorithm has continued to be frequently, and successfully used to perform black-box optimization in fields ranging from finance to operations research to electrical engineering. The identification of one or more computational efficiencies of the SGA still promises to help us understand the reason for its success.

A. Computational Competencies and Proficiencies

To demonstrate an efficiency of some computational system one typically shows that the system can scalably and robustly solve some problem—typically the problem that the system was *explicitly* designed to solve. But what about computational systems that aren’t explicitly designed to solve some well-defined problem (e.g. brains, genetic algorithms)? While we may conjecture that some of these systems are capable of efficient computation, expressing this sentiment rigorously is far from simple. A fundamental difficulty lies in identifying a *general* problem such that one can derive impressive bounds on the computational complexity of the system when it is applied to the problem.

Suppose, instead, one succeeds in identifying a set of *specific* problems for which impressive computational bounds can be derived. Even if the problems in this set are highly specific, it may be possible, by noting similarities between the problems, to intuitively infer the outlines of a more general problem that the system can tackle efficiently. To distinguish between a computational system’s ability to efficiently tackle some vaguely defined general problem, and its proven ability to efficiently solve some well-defined specific problem, we refer to the former ability as a *computational proficiency*, and the latter ability as a *computational competency*.

In this paper we identify two computational competencies of the SGA. That is, we identify two specific problems, and show that the SGA can solve each prob-

lem very efficiently—certainly more efficiently than the “mainstream” computational technique for solving these problems. When these two competencies are considered together, they point unmistakably towards a powerful computational proficiency of the SGA. Remarkably, the general problem that this proficiency is concerned with is closely related to a well-known statistical problem at the cutting edge of computational genetics having to do with the identification of epistatically interacting quantitative trait loci (QTLs).

B. Epistatically Interacting QTLs

Consider a phenotypic trait for which there exists a single polymorphic locus¹, 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 [13, p164], grain weight in rice plants [32], 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 [9] 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 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.”

¹A locus with multiple alleles

		(a)		
		Locus A		Marginal Values
		0	1	
Locus B	0	+0.2	-0.2	0.0
	1	-0.2	+0.2	0.0
Marginal Values		0.0	0.0	

				(b)
A	B	C		Marginal Values
0	0	0		+0.18
0	0	1		-0.06
0	1	0		-0.06
0	1	1		-0.06
1	0	0		-0.06
1	0	1		-0.06
1	1	0		-0.06
1	1	1		+0.18

TABLE I
MARGINAL VALUES OF TWO (TOP TABLE) AND THREE (BOTTOM TABLE) BI-ALLELIC INTERACTING LOCI. NONE OF THE LOCI HAVE MAIN EFFECTS

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].” [9]

It is easy to see how a group of loci can interact even though no locus in this group has a main effect. Table I(a) shows how this might happen when two loci A and B interact (for the sake of simplicity we have assumed bi-allelic haploid genomes). Note how neither of these

loci have main effects (the marginal value of each allele of each locus is zero) even though they clearly influence the trait in question. Table I(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 value of each allele of each locus is zero.) In fact for any non-empty set of loci $\{A_1, \dots, A_n\}$, and any set of bits $\{b_1, \dots, b_n\}$, one can construct a similar table by letting the marginal values of two genotypes $b_1 \dots b_n$, and $\bar{b}_1 \dots \bar{b}_n$ be some value δ and letting the marginal values of all other genotypes be $-\frac{2\delta}{2^n-2}$ (This observation will come in handy in our definition of type 1 pivotal functions in section III).

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

Main effects can be detected by visiting loci one at a time and testing for differentiated marginals (marginals with non-zero marginal values). Let us call this strategy *differentiated marginal testing* (DMT). To the best of our knowledge, the only sure way to accommodate for epistasis between loci when main effects are absent, is to visit *multi-locus combinations*, and to test the (multivariable) marginal of each such combination for differentiation. We shall call this approach *combinatorial differentiated marginal testing*, or combinatorial DMT for short. The computational intractability of combinatorial DMT, even for small combination sizes, is discussed in a recent article by Moore [20]. 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 4.9×10^{11} two-way combinations of [loci] would require approximately 5.7 days. Exhaustively evaluating all of the approximately 1.6×10^{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.”

The problem described above (see also [22], and [21]) is a specific instance of the general problem of identifying interacting attributes in data-mining [10].

In this paper we focus on generative versions of two specific problems having to do with the identification of interacting-attributes. Crucially, main effects are entirely absent in both problems. By “generative” we mean that the value of any synthesized data point can be queried—like in active learning. The first problem can only be solved by a combinatorial DMT 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 can only be solved by a combinatorial DMT strategy that tests attributes in combinations of four or more; the time required by this strategy is therefore $\Omega(\ell^4)$, where ℓ is the number of attributes of an instance of the problem. We will show that both the first and the second problem can be solved robustly by an SGA in time that is *linear* with respect to the number of attributes. Moreover, we will show that in both cases, the query complexity² of the SGA is *constant* with respect to the number of attributes.

II. 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. [28], [19], [8], [17]). 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

²Because fitness evaluation is by far the most time consuming part of a typical GA run, genetic algorithmicists often use the term *time complexity* to refer to the relationship between a parameter of some problem and the number of fitness evaluations required by a GA to solve the problem. In this paper we use the term *query complexity* instead. Our usage of this term is in line it’s usage in theoretical computer science (the fitness function of a GA can be thought of as an oracle that gets “queried” by the GA). We use the term *time complexity* as it is typically used in computer science—to refer to the relationship between a problem parameter and the number of “basic steps” required to solve the problem.

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 circumvent 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 only pertains to the parameter values 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 our aim is 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. In doing so we obtain *quantitative* results from a single experiment for an infinite set of problem instances.

While the use of symmetry arguments is new to GA research, symmetry arguments have long been exploited in other fields of science, such as 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”[16, p331-332].

One does not, however, need to venture so far afield in order to find an example of a symmetry argument. Let \mathfrak{B}_n denote the set of bitstrings of length n . For any bitstring g , let \bar{g} denote the bitwise complement of g (for example, $\bar{1011} = 0100$). Let ℓ be some positive integer, and let f be some fitness function over \mathfrak{B}_ℓ such that for any bitstring $g \in \mathfrak{B}_\ell$, $f(g) = f(\bar{g})$ (for example, if $\ell = 4$ and $f(1011) = 2.75$, then $f(0100) = 2.75$). Let G be some finite population SGA with fitness function f , such that the initial generation of G is drawn from the uniform distribution over the set \mathfrak{B}_ℓ . For any generation t , let $p^{(t)}(g)$ denote the probability that some bitstring $g \in \mathfrak{B}_\ell$ will be in the population of G in generation t . Then by appreciating the symmetry of the situation, we can deduce that for any generation t , and any bitstring $g \in \mathfrak{B}_\ell$, $p^{(t)}(g) = p^{(t)}(\bar{g})$.

This result holds regardless of the size of the population, the mutation and crossover rates, the mutation and crossover operators used, and the way in which the SGA G scales the fitness values of individuals (if it does) and performs selection. In fact, this result holds even if

f is stochastic (i.e. if f returns different values when applied to the same genome) as long as for any genome g , the values of $f(g)$ and $f(\bar{g})$ are drawn from the same distribution. Since mathematical models of genetic algorithms with finite populations (e.g. [24]) tend to be unwieldy, a formal proof of the above, i.e. a proof within some formal axiomatic system, would be quite involved and would be relatively inaccessible. “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 [16, p331]. Symmetry arguments, in other words, allow one to cut through complications that might hobble other forms of reasoning.

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 cannot be sure that the initial population of G is drawn from the uniform distribution over \mathfrak{B}_ℓ , then for any generation t , and any bitstring g , we would, in a sense, be ‘equally ignorant’ of the values of $p^{(t)}(g)$ and $p^{(t)}(\bar{g})$. Our ‘equal ignorance’ does not, of course, entail that these two values are the same.

But what constitutes ‘positive knowledge of symmetry’? This question, like the question “what constitutes beauty?”, has no direct answer. Historically, the appreciation of a symmetry by a community of theorists well versed in “the art” was enough to constitute ‘positive knowledge’ of that symmetry. Interestingly, over the last two centuries, mathematicians have largely agreed to eschew all non-sentential symmetries in their *formal* communication; these days, only certain types of symmetry between sentential forms are acknowledged. At the beginning of this deep shift in the communication of mathematics, Euclidian geometry was one of the only fields of mathematics that had an axiomatic foundation [23]. By the end of the shift, “new as well old branches of mathematics ... were supplied with what appeared to be adequate sets of axioms” [23]. An obvious benefit of this shift is a reduction in the number of mistakes communicated. The rarely acknowledged cost is the impedance of timely and accessible communication of results derived through the insightful exploitation of non-sentential forms of symmetry.

III. TYPE 1 PIVOTAL FUNCTIONS

We begin by defining a class of fitness functions with bitstring inputs such that when any of these functions is queried with an infinite set of samples drawn from a uniform distribution over its domain, no locus has a main effect, even though some loci may interact epistatically

with others. For reasons that will soon become clear we call the members of this class *pivotal functions*.

For any positive integer n , let $[n]$ denote the set of positive integers $\{1, \dots, n\}$. 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 b let \bar{b} denote the complement of b . Let $\mathcal{N}(\mu, \sigma^2)$ denote the normal distribution with mean μ and variance σ^2 .

Definition 1. Let $\psi = (o, \sigma, \delta, \ell, L, V)$ be a 6-tuple such that o is a positive integer, σ and δ are non-negative real numbers, ℓ is a positive integer greater than o , V is an o -tuple of binary values, and L is an o -tuple of distinct positive integers in $[\ell]$ sorted in ascending order. A type 1 pivotal function with descriptor ψ is a stochastic function f over the set of bitstrings of length ℓ which behaves as follows: for any input bitstring g , if $(g_{L_1} = V_1) \wedge \dots \wedge (g_{L_o} = V_o)$ or $(g_{L_1} = \bar{V}_1) \wedge \dots \wedge (g_{L_o} = \bar{V}_o)$ then f returns a value drawn from $\mathcal{N}(\delta, \sigma^2)$, otherwise f returns a value drawn from $\mathcal{N}(-\frac{2\delta}{2^o-2}, \sigma^2)$.

We call o, δ, σ, ℓ and V the *order, increment, noisiness, and span* of a pivotal function respectively. When a pivotal function is queried with some bitstring, the distribution from which the result is drawn pivots upon the values of the bitstring at the *pivotal loci* given by L , and the *pivotal values* given by V —hence the name *pivotal function*. If we assume that a type 1 pivotal function is queried with samples drawn from the uniform distribution over the function’s domain, then the *expected* marginal value of each allele of any individual locus is zero. This is what we mean when we say that no locus has a main effect. The increment parameter δ determines the strength of the expected multilocus marginal values of the pivotal loci.

Let f be a type 1 pivotal function with descriptor $(o = 3, \sigma = 1, \delta = 0.18, \ell, L, V)$. The pdfs of $\mathcal{N}(-\frac{2\delta}{2^o-2}, \sigma^2)$ and $\mathcal{N}(\delta, \sigma^2)$ are shown in Figure 1. Consider the task of robustly recovering the indices of the pivotal loci (i.e. the values of L) given only the values of o, σ, δ and ℓ , and query access to the function f . Because of the stochastic nature of f , as long as there is any overlap between the two pdfs there will always be some probability of error. The *large* overlap between the two pdfs shown in Figure 1 make the minimization of this error expensive as ℓ gets large (say 10^6). But it is the absence of main effects that *really* makes this problem thorny. A DMT strategy that visits 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 visited in combinations of two or more. The number of such combinations however scales

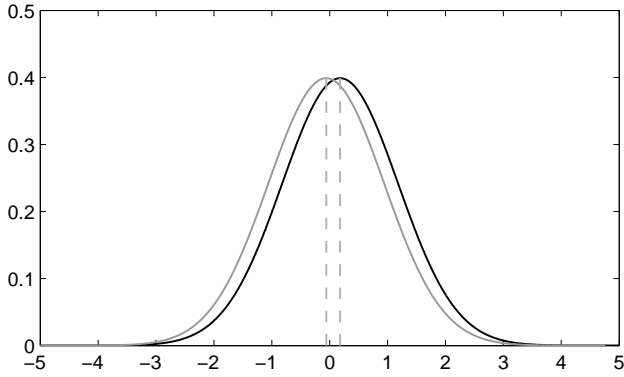


Fig. 1. The pdfs of two normal distributions with standard deviation 1, and means -0.06 (grey) and 0.18 (black)

at least quadratically with ℓ .

We will show that an SGA with uniform crossover 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 ℓ , and with some number of queries that is *constant* with respect to ℓ .

A. Symmetry Analysis

For our purposes a *semi-parameterized* SGA is an SGA with just two “parameters”: a positive integer ℓ which specifies the length of the genomes, and a fitness function over the set of bitstrings \mathfrak{B}_ℓ . Two semi-parameterized SGAs are considered to be distinct if they differ in their crossover rates, say, or in the selection schemes that they use. It is important to clarify that the per-bit mutation rate of a semi-parameterized SGA is not dependent on the genome length. For any positive integer ℓ , any semi-parameterized SGA G and any fitness function f over \mathfrak{B}_ℓ , we use the “oracle notation” of theoretical computer science to denote the (unparameterized) SGA that results when the length of the bitstrings of G is fixed at ℓ , and the fitness function that G queries is set to f ; specifically, we denote this SGA by G^f .

For any positive integer n let Υ_n denote the set $\{0, \frac{1}{n}, \frac{2}{n}, \dots, \frac{n-1}{n}, 1\}$. 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 any unparameterized SGA A with population size N , let $\mathbf{1}_{(A,i)}^{(t)} : \Upsilon_N \rightarrow [0, 1]$ be a probability mass function such that, for any $x \in \Upsilon_N$, $\mathbf{1}_{(A,i)}^{(t)}(x)$ is the probability that the one-frequency of locus i after t generations of running A is x . Likewise let $\mathbf{0}_{(A,i)}^{(t)} : \Upsilon_N \rightarrow [0, 1]$ be a probability mass function such that, for any $x \in \Upsilon_N$, $\mathbf{0}_{(A,i)}^{(t)}(x)$ is the probability

that the zero-frequency of locus i after t generations of running A is x . We call such distributions one- and zero-frequency distributions. Finally, let $\uparrow_{(A,i)}^{(t)}$ and $\downarrow_{(A,i)}^{(t)}$ be random variables that give the one- and zero-frequencies, respectively, of locus i in generation t . Clearly then, the probability mass functions of $\uparrow_{(A,i)}^{(t)}$, and $\downarrow_{(A,i)}^{(t)}$ are $\mathbf{1}_{(A,i)}^{(t)}$ and $\mathbf{0}_{(A,i)}^{(t)}$ respectively. Note that for any $x \in \Upsilon_N$, $\mathbf{0}_{(A,i)}^{(t)}(x) = \mathbf{1}_{(A,i)}^{(t)}(1-x)$, and $\mathbf{1}_{(A,i)}^{(t)}(x) = \mathbf{0}_{(A,i)}^{(t)}(1-x)$.

Proposition 1. *Let G be a semi-parameterized SGA, and let f be a type 1 pivotal function. Then for any locus k of G^f , and for any generation t*

- (a) $\mathbf{1}_{(G^f,k)}^{(t)} = \mathbf{0}_{(G^f,k)}^{(t)}$
- (b) $\mathbf{E}[\uparrow_{(G^f,k)}^{(t)}] = \mathbf{E}[\downarrow_{(G^f,k)}^{(t)}] = \frac{1}{2}$

ARGUMENT: For any generation t , and any locus k , Part (a) follows by consideration of the symmetry between $\mathbf{1}_{(G^f,k)}^{(t)}$, and $\mathbf{0}_{(G^f,k)}^{(t)}$ induced by the type 1 pivotal function f . Part (b) follows from part (a) and the claim that for any unparameterized SGA A , any locus i of A , and for any generation t , $\mathbf{E}[\uparrow_{(A,i)}^{(t)}] + \mathbf{E}[\downarrow_{(A,i)}^{(t)}] = 1$. For a proof of the claim note that if N is the size of the population of G , then for any generation t , and for any locus i ,

$$\sum_{z \in \Upsilon_N} \mathbf{0}_{(A,i)}^{(t)}(z)z = \sum_{z \in \Upsilon_N} \mathbf{0}_{(A,i)}^{(t)}(1-z)(1-z)$$

So,

$$\begin{aligned} & \sum_{x \in \Upsilon_N} \mathbf{1}_{(A,i)}^{(t)}(x)x + \sum_{y \in \Upsilon_N} \mathbf{0}_{(A,i)}^{(t)}(y)y \\ &= \sum_{x \in \Upsilon_N} \mathbf{1}_{(A,i)}^{(t)}(x)x + \mathbf{0}_{(A,i)}^{(t)}(1-x)(1-x) \\ &= \sum_{x \in \Upsilon_N} \mathbf{1}_{(A,i)}^{(t)}(x)x + \mathbf{1}_{(A,i)}^{(t)}(x)(1-x) \\ &= \sum_{x \in \Upsilon_N} \mathbf{1}_{(A,i)}^{(t)}(x) \\ &= 1 \end{aligned}$$

Note that proposition 1 holds for any type 1 pivotal function, and a semi-parameterized SGA with any population size, any commonly used selection operator (e.g. rank, tournament, fitness proportional etc.), any of the typical crossover and mutation operators, any mutation and crossover rates, and any fitness scaling scheme. Imagine having to prove all of this without appealing to the symmetries of SGAs with type 1 pivotal functions.

Uniform crossover [1], if present, adds yet another exploitable symmetry. This form of crossover was popularized by Syswerda [28], who showed that uniform 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. Syswerda also observed that *any* homologous crossover operation can be represented by a probability distribution over the set of 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 *absence of positional bias* [6] is a crucial property of uniform crossover that we will exploit forthwith. For the sake of brevity we call an SGA with uniform crossover a UGA.

Definition 2. Let f be a type 1 pivotal function with descriptor $(o, \delta, \sigma, \ell, L, V)$. Then the basic form of f is a type 1 pivotal function with descriptor $(o, \delta, \sigma, o + 1, (1, \dots, o), (1, \dots, 1))$.

According to this definition if some function f^* is the basic form of some type 1 pivotal function f with order o then the span of f^* is $o + 1$. The first o loci of any input to f^* will be pivotal, the last locus of any input to f^* will be non-pivotal, and the pivotal values used by f^* are all ones. We say that a type 1 pivotal function f with descriptor $(o, \delta, \sigma, L, V)$ is *basic* if the basic form of f is f . Since the last three elements of the descriptor of f are then derivable from the first three, we write this descriptor as (o, δ, σ) .

Proposition 2. Let A be a semi-parameterized UGA, let f be a type 1 pivotal function with descriptor $(o, \delta, \sigma, \ell, L, V)$, and let f^* be the basic form of f . Then, for any generation t ,

- (a) For any pivotal locus k , $\mathbf{1}_{(A^f, k)}^{(t)} = \mathbf{1}_{(A^{f^*}, 1)}^{(t)}$
- (b) For any non-pivotal locus k , $\mathbf{1}_{(A^f, k)}^{(t)} = \mathbf{1}_{(A^{f^*}, o+1)}^{(t)}$

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

ARGUMENT: Let f' be a type 1 pivotal function with descriptor $(o, \delta, \sigma, \ell, L, (1, \dots, 1))$. We shortly present four claims. Part (a) of proposition 2 follows from claim 1(a), claim 2, claim 4 and proposition 1(a). Part (b) of the above proposition follows from claim 1(b) and claim 3.

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

- (a) For any $i \in [o]$, if $V_i = 1$, then $\mathbf{1}_{(A^{f'}, L_i)}^{(t)} = \mathbf{1}_{(A^f, L_i)}^{(t)}$, otherwise $\mathbf{1}_{(A^{f'}, L_i)}^{(t)} = \mathbf{0}_{(A^f, L_i)}^{(t)}$.
- (b) For any generation t , and any non-pivotal locus k of $A^{f'}$, $\mathbf{1}_{(A^{f'}, k)}^{(t)} = \mathbf{1}_{(A^f, k)}^{(t)}$

Claim 2. For any generation t , and any $i \in [o]$,

$$\mathbf{1}_{(A^{f^*}, i)}^{(t)} = \mathbf{1}_{(A^{f'}, L_i)}^{(t)}$$

Claim 3. For any generation t , and any non-pivotal locus k of A^f , $\mathbf{1}_{(A^{f^*}, o+1)}^{(t)} = \mathbf{1}_{(A^{f'}, k)}^{(t)}$

Claim 4. For any generation t ,

$$\mathbf{1}_{(A^{f^*}, 1)}^{(t)} = \mathbf{1}_{(A^{f^*}, 2)}^{(t)} = \dots = \mathbf{1}_{(A^{f^*}, o)}^{(t)}$$

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

Claims 2 follows by consideration of the symmetry between loci L_1, \dots, L_o of $A^{f'}$ and loci $1, \dots, o$ of A^{f^*} respectively. Claim 3 follows by consideration of the symmetry between any non-pivotal locus of $A^{f'}$ and locus $o + 1$ of A^{f^*} . These symmetries follows from the absence of positional bias in uniform crossover and from the definition of the ladder functions f^* and f' .

To make these symmetries manifest, we offer the two “vertical views” shown in Figure 2. Figure 2(a) shows a hypothetical population of $A^{f'}$ with three pivotal loci (whose locations are marked by shaded columns). Given the definition of the fitness function of $A^{f'}$, it is easy to see that the fitness of any genome depends only 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 $A^{f'}$. 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 each of the first o loci of A^{f^*} . This symmetry follows from the absence of positional bias in uniform crossover, and from the definition of the fitness function used by A^{f^*} . \square

1	0	1	1	0	1	0	0	1	0	1	0	1	0	0	0	1	0
0	0	1	0	1	0	0	1	1	0	1	0	0	1	0	1	0	0
1	0	0	1	0	0	1	1	1	0	1	0	1	1	0	0	1	0
1	1	0	0	1	0	1	0	0	0	1	1	0	1	0	1	0	1
1	0	0	0	0	1	0	1	0	0	1	0	0	1	0	1	1	0
0	0	0	1	0	1	0	0	1	1	0	1	0	0	0	1	0	0
1	1	0	1	0	0	1	1	1	0	1	1	0	1	0	0	1	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

(a) “Vertical view” of a hypothetical population. Each row is a genome

$x = 1$	0	0	1	0	0	1	1	1	0	1	0	1	1	0	0	1	0
$m = X_1$	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	X_{16}	X_{17}	X_{18}
$y = 1$	1	0	0	1	0	1	0	0	0	1	1	0	1	0	1	0	1
$z =$?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

(b) “Vertical view” of a hypothetical crossover operation

Fig. 2. Subfigure (a) shows a “vertical view” of a hypothetical population. Each row is a genome. The shaded columns show the positions of three hypothetical pivotal loci. By the definition of a type 1 pivotal function (see text), only the bits in the shaded columns 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 independent identically distributed random binary variables that comprise the mask m .

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

B. Experiment 1

Let W denote the semi-parameterized UGA described in the materials and methods section in the appendix, and let f_1^* denote a basic type 1 pivotal function with descriptor ($o = 3$, $\delta = 0.18$, $\sigma = 1$). Figure 3 shows the one-frequency dynamics of the first and last locus of $W^{f_1^*}$ in each of 3000 runs. In all 3000 runs the first locus went to fixation³ by the 200th generation, whereas the one-frequency of the last locus in the 200th generation was always between 0.9 and 0.1.

In order to clearly describe the rest of our findings we develop the following notation. We denote a *schema*

³We 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.

partition [18] by a tuple consisting of the indices of the defining positions of that schema partition—e.g. (2, 15, 3). The *order* of a schema partition Γ , denoted by $o(\Gamma)$, is the number of elements in some tuple that denotes Γ . The denotation of a *schema* is dependent on the denotation of the schema-partition that the schema belongs to. For any genome g , let g_i denote the i^{th} bit of g . Given a schema partition denoted by some tuple Γ , the schemata in this partition are denoted by binary strings of length $o(\Gamma)$. Let $b_1, \dots, b_{o(\Gamma)}$ be some bits. Then, $b_1 \dots b_{o(\Gamma)}$ denotes the schema consisting of the genomes $\{g | g_{\Gamma_1} = b_1 \wedge \dots \wedge g_{\Gamma_{o(\Gamma)}} = b_{o(\Gamma)}\}$. The denotation of the relevant schema partition must always be borne in mind when interpreting a denoted schema.

In addition to the findings reported above, we found that 200 generations into each run of $W^{f_1^*}$, either the schema 000, or the schema 111, of the schema partition (1, 2, 3), dominated the population. The average fraction of the population that belonged to the dominant schema at the end of 200 generations was 0.9563 (with standard error 1.56×10^{-4}).

Given the conclusions of our symmetry analysis, the result shown in Figure 3 provides us with a window into

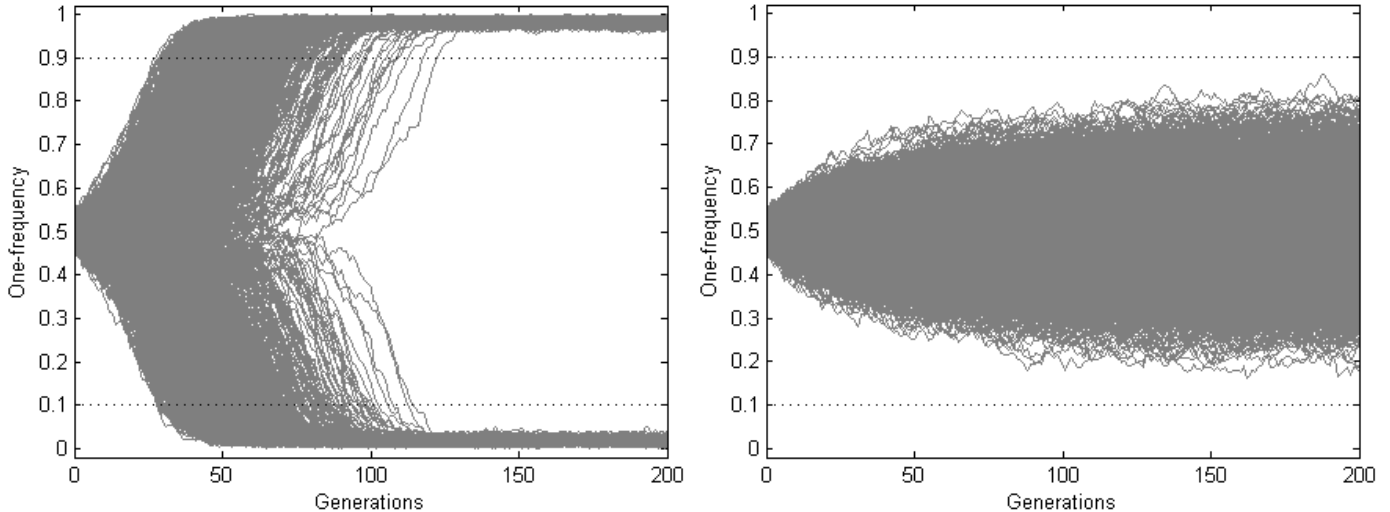


Fig. 3. The one-frequency dynamics of the first (*left*) and fourth (*right*) loci of the UGA $W^{f_1^*}$

the frequency dynamics of any UGA W^{f_1} , where f_1 is a type 1 pivotal function whose basic form is f_1^* . We infer that the pivotal loci of W^{f_1} will tend to go to fixation by the 200th generation. We also infer that the divergence from 0.5 of the one-frequencies of the non-pivotal loci of W^{f_1} will tend not to be not extreme.

We now explain the behavior of W^{f_1} that we have just deduced. Note that while this discussion is speculative and imprecise, it is entirely tangential to our aim of identifying computational competencies of the SGA. The one-frequency dynamics of the non-pivotal loci of W^{f_1} is easily explained by the notion of *drift*. To understand the frequency dynamics of the pivotal 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 locus⁴. Crucially, crossover does not change the composition of these bit-pools. Now, let x_1, x_2 , and x_3 denote the indices of the pivotal loci of W^{f_1} , 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 schema 010 of the schema partition $\langle x_1, x_2, x_3 \rangle$. 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 genomes except those that belong to the “sibling” schema 101. Thus, regardless of what happens during crossover, once generated, a genome of type 010 will tend to increase the frequency of 0, 1, and 0 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 schemata 010 and 101 “compete” for dominance of the bit-pools of each of the pivotal loci. One of these schemata eventually manages to gain an edge in “pulling” the composition of the bit-pools of all three pivotal loci far enough in it’s favor that a self-reinforcing loop that heavily favors the future generation of the victorious schema then ensues.

In light of this analysis, one can conclude that the building block hypothesis [11], [18], [14] 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 at the defining positions of that schema in a way that favors the future generation of genomes belonging to the schema.

C. A Computational Competency of the SGA

Consider Algorithm 1. The results of our experiment with W^{f^*} suggest that when CLASSIFYLOCI_200 is applied to f_1 , it will classify each locus of f_1 fairly

⁴Changes in the one and zero frequencies of a locus can be visualized as changes in the composition of a *pool of bits*. The bit-pool metaphor is especially useful in conjunction with the “vertical view” of a population presented in figure 2(a); each column can be thought of as a pool of bits

Algorithm 1: CLASSIFYLOCI_n

Input: a type 1 or type 2 pivotal function f
 $\ell = \text{span of } f$
 $\text{pivotalLoci} = \{\}$
 $\text{nonPivotalLoci} = \{\}$
 $P = \text{population of } W^f \text{ after } n \text{ generations}$
for $i = 1$ **to** ℓ **do**
 $x = \text{one-frequency of locus } i \text{ in population } P$
if $0.1 \leq x$ **and** $x \leq 0.9$ **then**
 $\text{nonPivotalLoci} = \text{nonPivotalLoci} \cup \{i\}$
else
 $\text{pivotalLoci} = \text{pivotalLoci} \cup \{i\}$
end
end
return $\text{pivotalLoci}, \text{nonPivotalLoci}$

accurately. Let us quantify this accuracy. Note that in *all* 3000 runs of $W^{f_1^*}$, by the end of the 200th generation, the one-frequency of the first locus was outside the interval $[0.1, 0.9]$, 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 $W^{f_1^*}$ will be *inside* $[0.1, 0.9]$ at the end of the 200th generation. 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 $[0.1, 0.9]$ at the end of the 200th 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 3×10^{-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 $W^{f_1^*}$ will be *outside* the interval $[0.1, 0.9]$ at the end of 200 generations. Using very similar reasoning we reject this hypothesis at the 3×10^{-7} level of significance. Thus, with probability of error less than three in ten million, the following statement is true: for any locus k of f_1 , there is less than a 0.005 probability that CLASSIFYLOCI₂₀₀ will misclassify locus k .

Note that ℓ , may be *any* positive integer greater than 3. There are $\binom{\ell}{3} \in \Omega(\ell^3)$ possible combinations of the three pivotal indices⁵. Remarkably, CLASSIFYLOCI₂₀₀ achieves the level of robustness mentioned above ($p < 0.005$ per locus) in time that is linear in ℓ , after making some number of fitness evaluations that is *constant* with respect to ℓ

IV. TYPE 2 PIVOTAL FUNCTIONS

If the order o of a type 1 pivotal function is greater than one, then even though no individual locus will have

⁵To obtain this bound we have used the inequality $(n/k)^k \leq \binom{n}{k}$. See [29].

differentiated marginal effects, certain combinations of *two* loci, will have differentiated (multilocus) marginal effects (specifically, those combinations in which both loci are pivotal). Thus for any $o > 1$, loci can be classified as pivotal or non-pivotal with a fixed level of robustness using a combinatorial DMT strategy that visits some number of two-locus combinations that is quadratic in the span of the pivotal function⁶. Type 2 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 binary addition modulo 2 operator). Type 2 pivotal functions are defined as follows:

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

The order of a type 2 pivotal function is always even; furthermore, no pivotal values are associated with the pivotal loci.

Let f be some type 2 pivotal function with span ℓ . Observe that for any bitstring g of length ℓ , $f(g) = f(\bar{g})$. Observe also that as \oplus is associative and commutative, the order in which it is applied to the pivotal bits of some bitstring is immaterial. Both of these observations reveal symmetries of f that we will exploit forthwith. These symmetries can also be seen in Table II, which shows the marginal of the pivotal loci of some type 2 pivotal function with increment δ , and order four.

Finally observe that for any type 2 pivotal function with order o , the multilocus marginal of any combination of $m < o$ alleles of distinct loci will not be differentiated. Thus the time complexity of a combinatorial DMT strategy that robustly identifies the pivotal loci is $\Omega(\ell^o)$.

Let f be a type 2 pivotal function with descriptor ($o = 4, \delta = 0.25, \sigma = 1, \ell, L$). We now show that a UGA can identify the pivotal loci of f relatively robustly (with less than a 0.005 chance of misclassification per locus)

⁶As o increases, the constant associated with this scaling relationship will increase very quickly. Nevertheless for any fixed value of o , the number of combinations that must be visited scales quadratically with the span of a type 1 pivotal function.

A	B	C	D	Expected Marginal Values
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	$-\delta$

TABLE II
THE EXPECTED MARGINAL OF THE PIVOTAL LOCI OF A TYPE 2 PIVOTAL FUNCTION

in time that is *linear* in ℓ , and with some number of queries that is *constant* with respect to ℓ . Our approach is almost identical to the approach we took in Section III, where we showed a similar result for a class of pivotal functions of type 1.

A. Symmetry Analysis

We define the *basic form* of a type 2 pivotal function as follows:

Definition 4. Let f be some type 2 pivotal function with descriptor $(o, \delta, \sigma, \ell, L)$. We define the basic form of f to be a type 2 pivotal function with descriptor $(o, \delta, \sigma, o + 1, (1, \dots, o))$

Let f be a type II pivotal function with descriptor $(o, \delta, \sigma, \ell, L, V)$. We say that f is *basic* if the basic form of f is f . Since the last two elements of the descriptor of f are derivable from the first three, we write this descriptor as (o, δ, σ) .

Proposition 3. Let A be a semi-parameterized UGA, let f be a type 2 pivotal function with descriptor $(o, \delta, \sigma, \ell, L, V)$, and let f^* be the basic form of f . Then, for any generation t ,

- (a) For any pivotal locus k , $\mathbf{1}_{(A^f, k)}^{(t)} = \mathbf{1}_{(A^{f^*}, 1)}^{(t)}$
- (b) For any non-pivotal locus k , $\mathbf{1}_{(A^f, k)}^{(t)} = \mathbf{1}_{(A^{f^*}, o+1)}^{(t)}$

This proposition is almost word-for-word identical to Proposition 2. Likewise the argument for this proposition, is very similar to the argument for Proposition 2.

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

By proposition 3(a), for any pivotal locus k of A^f , drawing monte-carlo samples from $\mathbf{1}_{(A^f, 1)}^{(t)}$ is equivalent to drawing monte-carlo samples from $\mathbf{1}_{(A^f, k)}^{(t)}$. And by proposition 3(b), for any non-pivotal locus k of A^f , drawing monte-carlo samples from $\mathbf{1}_{(A^{f^*}, o+1)}^{(t)}$ is equivalent to drawing monte-carlo samples from $\mathbf{1}_{(A^f, k)}^{(t)}$.

B. Experiment 2

Recall that W denotes the semi-parameterized UGA described in the materials and methods section in the appendix. Let f_2^* denote a basic type 2 pivotal function with descriptor $(o = 4, \delta = 0.25, \sigma = 1)$. We executed 3000 runs of the UGA $W^{f_2^*}$. The one-frequency dynamics of the first and last loci in each run is plotted in Figure 4. In all 3000 runs the first locus went to fixation by generation 1000, whereas the one-frequency of the last locus in generation 1000 was always between 0.9 and 0.1. We found that 1000 generations into each run the population was dominated by some schema $b_1 b_2 b_3 b_4$ of the schema partition $(1, 2, 3, 4)$ with $b_1 \oplus b_2 \oplus b_3 \oplus b_4 = 1$. On average the fraction of the population that belonged to the dominant schema in generation 1000 was 0.9634 (with standard error 1.22×10^{-4}).

Let f_2 be a type 2 pivotal function with span ℓ such that the basic form of f_2 is f_2^* . The conclusions of our symmetry analysis of type 2 pivotal fitness functions, and the result shown in Figure 4 provide us with a window into the frequency dynamics of all pivotal and non-pivotal loci of W^{f_2} .

C. A Second Computational Competency of the SGA

Based on arguments that are almost identical to the ones in section III-C we conclude, with probability of error less than three in ten million, that the following statement is true: For any locus of f_2 , the probability that the locus will be misclassified by CLASSIFYLOCI_1000 is less than a 0.005. There are $\binom{\ell}{4} \in \Omega(\ell^4)$ possible configurations of the pivotal indices. Remarkably, CLASSIFYLOCI_1000 achieves the level of robustness mentioned above ($p < 0.005$ per locus) in time that is linear in ℓ , after making some number of fitness evaluations that is *constant* with respect to ℓ .

It merits mentioning that both the computational competencies showcased in this paper are essentially invisible to analytic approaches in which an infinite population

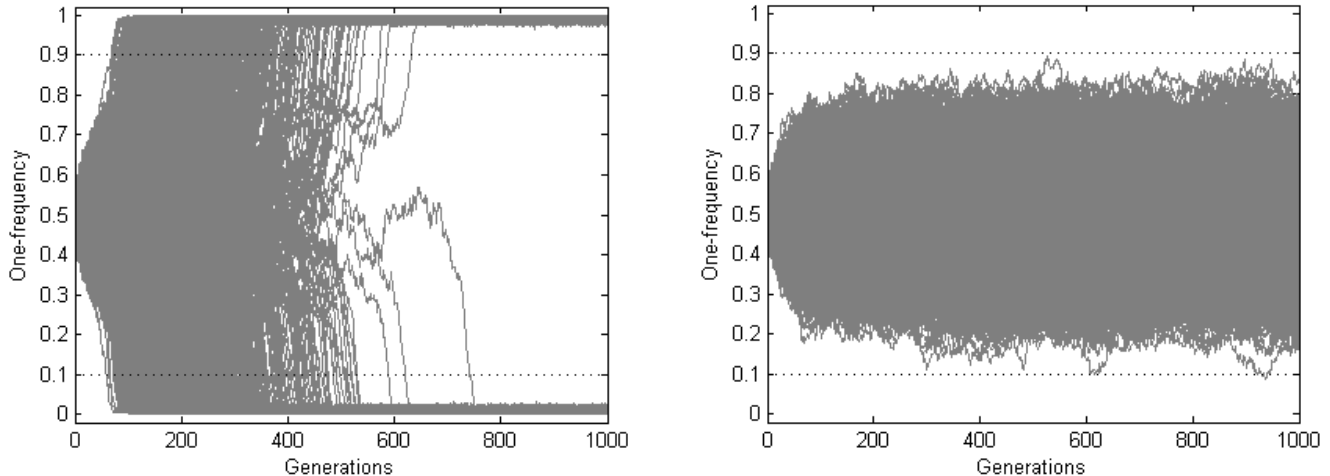


Fig. 4. The one-frequency dynamics of the first (*left*) and fifth (*right*) loci of the UGA $W^{f_2^*}$

is assumed (e.g. [30], [4], [3]). 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 in infinite population models of genetic algorithms.

V. CONCLUSION

This paper can be viewed as a response of sorts to the nihilism that the no free lunch theorems [31] have inspired. As mentioned in the introduction, no free lunch theorems are frequently, if subliminally, regarded as “proof” that in the practice of *black-box* optimization, one search algorithm is as good as another. If this is indeed the case, then genetic algorithms have no special advantage over other search algorithms; the fortuitous pairing of optimization problem with search algorithm is all. Effort toward the identification of computational efficiencies underlying the genetic algorithm’s capacity for practical blackbox optimization then seems misplaced.

Nihilism of this sort is puzzling given the frequency with which genetic algorithms are successfully applied as black-box optimization algorithms in practice. The importance accorded to Wolpert and Mcready’s results seems to be a reaction, or rather, an overreaction, to the empirical discovery of serious flaws [19], [8], [17] in certain widely accepted statements about the computational efficiency of the simple genetic algorithm; these statements came to be regarded as fact because of an unclear demarcation between mathematical deduction and speculation in the early genetic algorithmics literature [5].

It is important to emphasize that the problem was not speculation itself but the *blurring of the boundary* between deduction and speculation. If the mistakes of the past are not to be revisited, this boundary must be clearly maintained at all times. Accordingly, we have distinguished between a computational *competency* of the SGA—a *specific* computational efficiency that can be derived deductively⁷—and a computational *proficiency* of the SGA—a *general* computational efficiency that is inferred inductively from a set of related computational competencies.

In this paper we have derived two closely related computational competencies of the SGA. The general domain in which these competencies lie is noteworthy. A central explanandum of the field of genetic algorithmics, and indeed, of the field of evolutionary biology, is the persistence of adaptation in sexually evolving populations *despite* the ubiquity of epistatic interactions between unlinked 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 specific genotypes with above average fitness to fixation, is likely to be material to theories about the adaptive prowess of both the genetic algorithm and natural evolution.

⁷We clarify here that we offer the symmetry arguments presented in this paper as *deductive* arguments. These results have, of course, not been derived within some formal axiomatic system. However, to argue that this automatically makes our arguments non-deductive is to argue that deduction played a very limited role in mathematics before the nineteenth century.

REFERENCES

- [1] D.H. Ackley. *A connectionist machine for genetic hillclimbing*. Kluwer Academic Publishers, 1987.
- [2] James E. Baker. Adaptive selection methods for genetic algorithms. In John J. Grefenstette, editor, *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*. Lawrence Erlbaum Associates, Publishers, 1985.
- [3] Keki Burjorjee. Sufficient conditions for coarse-graining evolutionary dynamics. In *Foundations of Genetic Algorithms 9 (FOGA IX)*, 2007.
- [4] Keki Burjorjee and Jordan B. Pollack. A general coarse-graining framework for studying simultaneous inter-population constraints induced by evolutionary operations. In *GECCO 2006: Proceedings of the 8th annual conference on Genetic and evolutionary computation*. ACM Press, 2006.
- [5] Keki M. Burjorjee. The fundamental problem with the building block hypothesis. *CoRR*, abs/0810.3356, 2008.
- [6] L.J. Eshelman, R.A. Caruana, and J.D. Schaffer. Biases in the crossover landscape. *Proceedings of the third international conference on Genetic algorithms table of contents*, pages 10–19, 1989.
- [7] David B. Fogel. *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. IEEE Press, NY, 2000.
- [8] 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.
- [9] W. N. Frankel and N. J. Schork. Who’s afraid of epistasis. *Nature Genetics*, 14:371–373, 1996.
- [10] A.A. Freitas. Understanding the Crucial Role of Attribute Interaction in Data Mining. *Artificial Intelligence Review*, 16(3):177–199, 2001.
- [11] David E. Goldberg. *Genetic Algorithms in Search, Optimization & Machine Learning*. Addison-Wesley, Reading, MA, 1989.
- [12] David E. Goldberg. *The Design Of Innovation*. Kluwer Academic Publishers, 2002.
- [13] Daniel L. Hartl. *A Primer Of Population Genetics*. Harvard University Press, 2000.
- [14] John H. Holland. Building blocks, cohort genetic algorithms, and hyperplane-defined functions. *Evolutionary Computation*, 8(4):373–391, 2000.
- [15] Christian Igel and Marc Toussaint. Recent results on no-free-lunch theorems for optimization, March 31 2003. Comment: 10 pages, LaTeX, see <http://www.neuroinformatik.rub.de/PROJECTS/SONN/>.
- [16] E.T. Jaynes. *Probability Theory: The Logic of Science*. Cambridge University Press, 2007.
- [17] 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.
- [18] Melanie Mitchell. *An Introduction to Genetic Algorithms*. The MIT Press, Cambridge, MA, 1996.
- [19] 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. Bourguine, editors, *Proc. of the First European Conference on Artificial Life*, pages 245–254, Cambridge, MA, 1992. MIT Press.
- [20] Jason Moore. The critical need for computation in human genetics. http://www.reviews.com/hottopic/hottopic_essay_07.cfm, 2008.
- [21] 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.
- [22] J.H. Moore. The Ubiquitous Nature of Epistasis in Determining Susceptibility to Common Human Diseases. *Hum Hered*, 56:73–82, 2003.
- [23] Nagel, E. and Newman, J. R. *Gödel’s Proof*. New York Univ. Press, 2001.
- [24] A.E. Nix and M.D. Vose. Modeling genetic algorithms with Markov chains. *Annals of Mathematics and Artificial Intelligence*, 5(1):79–88, 1992.
- [25] M. Pelikan. *Hierarchical Bayesian Optimization Algorithms*. Springer Verlag, 2005.
- [26] Martin Pelikan, Kumara Sastry, and Erick Cantú-Paz, editors. *Scalable Optimization via Probabilistic Modeling*, volume 33 of *Studies in Computational Intelligence*. Springer, 2006.
- [27] C.R. Reeves and J.E. Rowe. *Genetic Algorithms: Principles and Perspectives: a Guide to GA Theory*. Kluwer Academic Publishers, 2003.
- [28] G. Syswerda. Uniform crossover in genetic algorithms. In J. D. Schaffer, editor, *Proceeding of the Third International Conference on Genetic Algorithms*. Morgan Kaufmann, 1989.
- [29] R. L. Rivest T. H. Cormen, C. H. Leiserson. *Introduction to Algorithms*. McGraw-Hill, 1990.
- [30] Michael D. Vose and Gunar E. Liepins. Punctuated equilibria in genetic search. *Complex Systems*, 5(1):31–44, February 1991.
- [31] David H. Wolpert and William G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, April 1997.
- [32] 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.

APPENDIX

MATERIALS AND METHODS

The arguments in this paper rest not just on the qualitative behavior of the semi-parameterized UGA we used, but on aspects of its *quantitative* behavior. It is therefore necessary to describe this UGA in sufficient detail that these aspects can be reproduced. While we believe that the description given below is sufficient to reproduce all quantitative aspects of the behavior of our UGA that are relevant to our proofs, small differences in implementation may interfere with reproducibility. We therefore release the Matlab code for the semi-parameterized UGA that we used. This code, along with the code for the pivotal fitness functions used is available for download⁸.

The semi-parameterized UGA we used (denoted by *W* in this paper) implements the specification for a simple genetic algorithm given by Mitchell [18, p 10], with two exceptions:

- 1) In each generation, right after evaluating the fitness of all individuals, our UGA used sigma scaling [18, p 167] to adjust the fitness of each individual,

⁸<http://cs.brandeis.edu/~kekib/competenciesMatlab.zip>

and used this adjusted fitness when selecting the parents of that generation. Suppose $f_x^{(t)}$ is the fitness of some individual x in some generation t , and suppose the average fitness and standard deviation of the fitness of the individuals in generation t are given by $\overline{f^{(t)}}$ and $\sigma^{(t)}$ respectively, then the adjusted fitness of x in generation t is given by $h_x^{(t)}$ where, if $\sigma^{(t)} = 0$ then $h_x^{(t)} = 1$, otherwise,

$$h_x^{(t)} = \min\left(0, 1 + \frac{f_x^{(t)} - \overline{f^{(t)}}}{\sigma^{(t)}}\right)$$

- 2) The SGA used universal stochastic sampling [2] [18, p 166] to select parents.

Selection is fitness-proportionate. the population size is 1500, the probability of crossover is one, and bit-flip mutation with a mutation rate of 0.003 per bit is used.