

EA Models and Population Fixed-Points Versus Mutation Rates for Functions of Unitation

J Neal Richter
Computer Science Department,
Montana State University
Bozeman, Montana
richter@cs.montana.edu

John Paxton
Computer Science Department,
Montana State University
Bozeman, Montana
paxton@cs.montana.edu

Alden Wright
Department of Computer Science
University of Montana-Missoula
Missoula, Montana
wright@cs.umt.edu

ABSTRACT

Using a dynamic systems model for the Simple Genetic Algorithm due to Vose[1], we analyze the fixed point behavior of the model without crossover applied to functions of unitation. Unitation functions are simplified fitness functions that reduce the search space into a smaller number of equivalence classes. This reduction allows easier computation of fixed points. We also create a dynamic systems model from a simple nondecreasing EA like the (1+1) EA and variants, then analyze this models on unitation classes.

Categories and Subject Descriptors

I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, and Search – Genetic Algorithms.

General Terms

Algorithms, Artificial Intelligence

Keywords

theory of evolutionary algorithms, infinite population models, unitation functions, fixed points, genetic algorithms.

1. INTRODUCTION

The Vose infinite population model [1] of simple genetic algorithms is a dynamic systems model that represents populations as a vector of proportions. This vector has dimension s , where s is the size of the search space. Each entry in the vector is the proportion of members in the global population that are represented by a given chromosome in the search space. This representation allows utilization of techniques and theorems from the mathematical theory of dynamic systems to analyze the Genetic Algorithm (GA).

This paper is restricted to examining a class of fitness functions called 'functions of unitation.' These functions establish equivalence classes, allowing a reduction in the dimensionality of the corresponding Vose model. The analysis will also be restricted to mutation-selection GAs with no crossover, elitism or

other advanced techniques.

Using the eigensystem of the mutation-only Vose model, fixed points of the mutation-selection GA can be found. Fixed points are population vectors such that applying the GA to them results in the same population vector. Fixed points are not the optimal points in the fitness landscape, they represent the expected long-run distribution of a population for a GA given a large enough population. A fixed point may or may not contain a significant proportion of the global optima of the fitness landscape.

This paper is largely an extension of Rowe [4] that further explores the effect of mutation rates on fixed points. We also reintroduce two simple variants of the (1+1) EA and model them using the same tools. The intent of this exploration is to lay the groundwork for studying how adaptive mutation schemes can be understood and designed.

2. THE VOSE DYNAMIC SYSTEMS MODEL

The Vose infinite population model is largely the work of Michael Vose[1]. The intent of the model is to allow a mathematical analysis of the Simple Genetic Algorithm. For our purposes, we define the Simple Genetic Algorithm as a GA with proportionate selection, bitwise mutation, and several standard crossover operators.

2.1 Background

The Vose model is a discrete dynamical system and is a kind of 'map.' Maps are discrete dynamic systems that translate an input to an output. An example is the well-known Logistic Map [2].

$$f(x) = \lambda x(1-x) \quad \{\text{where } \lambda \in \mathbb{R}; \lambda > 0; x \in \mathbb{R}\} \quad (1)$$

The Vose model translates the current population vector to the next population vector. Iterating the map simulates the trajectory of the GA (in the limit of infinite population size), where the next population vector becomes the input to the next generation of the GA. This forms a sequence of population vectors $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \dots, \mathbf{p}_k$. This sequence is the trajectory of the GA model through the population space.

The Vose model is a deterministic dynamical system. For a sufficiently large population in a real GA, this model allows us to make fairly accurate predictions of the *expected* next population and the long term behavior of the population [3]. If the population is small, then the actual populations produced by the GA will have wide variability compared to the predicted population. If the population is very large, then the actual

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'05, June 25-29, 2005, Washington, D.C., USA
Copyright 2005 ACM 1-59593-010-8/05/0006...\$5.00.

population distributions produced by the GA should be close to the predicted model for large number of generations [1]. Thus the Vose model is called an *infinite population model*.

2.2 Theory

The following section summarizes Rowe [3] as it applies to selection-mutation models of GAs. Given a binary genome of length d , the search space of the GA is of size $s=2^d$. To use the Vose model, we represent the population as a vector of proportions of length s , $\mathbf{p} = \{p_0, \dots, p_{s-1}\}$. Each p_i is the proportion of membership in the population by the binary string i .

For a 2-bit genome, a possible population vector is $\mathbf{p} = \{0.1, 0.2, 0.5, 0.2\}$. This could represent a population of 10 individuals, 1 copy of 00, 2 copies of 01, 5 copies of 10, and 2 copies of 11. Note that the population vectors have the simplex property. Each component of the vector is in the range $[0,1]$, and the sum of the components equals 1.

$$\Lambda = \left\{ (x_0, \dots, x_{s-1}) : \sum_{i=0}^{s-1} x_i = 1, \quad x_j \in [0,1] \quad x \in \mathbb{R} \right\} \quad (2)$$

Next, the properties of the model for mutation-selection GAs are given. The mutation-selection infinite population model is created as follows. Operator G is defined by:

$$p_{t+1} = G(p(t)) = \frac{1}{\mu(p_t)} U S p_t \quad (3)$$

where $\mu(p_t)$ is the average fitness of the population p_t . The $s \times s$ mutation matrix U is composed of the probabilities that a chromosome string j will mutate into string i . Note that this U matrix is symmetric. The probability a given bit in the chromosome string mutates to its complement state is q . The $s \times s$ selection matrix S is a diagonal matrix consisting of fitness values along the diagonal and zeros elsewhere. Dividing by $\mu(p_t)$ implements proportionate selection.

$$U_{i,j} = q^h (1-q)^{d-h} \quad h = \text{Hamming Distance } (i, j) \quad (4)$$

$$S_{k,k} = f(x_k) \quad k \in [0, s-1]$$

From the theory we know these five properties of the US matrix [3].

1. US is an irreducible, positive matrix, all entries are non-negative.
2. Fixed-points of the model are the normalized (so that all elements sum to 1) eigenvectors of US .
3. Only one normalized eigenvector is in the simplex (via Perron-Frobenius theorem [3][7]).
4. Eigenvalues of US are the average fitness of the population given by the corresponding eigenvector.
5. The largest eigenvector corresponds with the lone eigenvector inside the simplex.

These properties allow the computation of the fixed points of the infinite population model for a given fitness function and mutation rate. By normalized vector we mean that all elements have been scaled such that they all sum to 1, creating a vector that obeys the simplex property. The lone normalized fixed point is the global attractor of the dynamical system modeling the GA. This

fixed point is the *expected population distribution* of the GA for a sufficiently large number of generations and population size.

There can exist fixed points outside, but very near the simplex. Such fixed points contain at least one negative component and thus cannot represent a real population. Finite populations p near such fixed points may be *metastable states* [3] in that the distance from p to $G(p)$ is small. Thus, the region of the simplex near such a fixed point may be called a *metastable region*. If the GA population enters such a region, it will typically spend a long time in the region before escaping.

Note that fixed size populations form a subset of the simplex called a *lattice*. Some simplex population vectors (an example is any vector with an element like $2/3$ where the cardinality of the vector is not a multiple of 3) are not representable with finite populations. Finite populations move from lattice point to lattice point in the simplex. The smaller the population size, the sparser the lattice points are in the simplex.

3. FUNCTIONS OF UNITATION

Unitation functions are fitness functions where fitness is defined only by the number of ones in a chromosome $x: \{0,1\}^d$. All fitness values are non-negative:

$$u(x): \{0,1\}^d \rightarrow \mathbb{R}^+ \quad (5)$$

An example function for 3 bits is

$$u(0) = 3 \quad u(1) = 2 \quad u(2) = 1 \quad u(3) = 4$$

This definition allows us to reduce the dimensionality of the infinite proportionality population vector from $2^d \times 1$ to $(d+1) \times 1$. This vector is represented as $\mathbf{p} = \{p_0, \dots, p_d\}$, where p_k is the proportion of the population having exactly k ones. Note that this vector is of $d+1$ dimension as it must have an entry for the all zeros case. Using the above fitness function, a selection matrix S is defined as the $(d+1) \times (d+1)$ diagonal matrix $S_{k,k} = u(k)$.

The mutation matrix is \hat{U} , whose entries are the probabilities that a string with j ones mutates to a string with i ones. \hat{U} is defined as an $(d+1) \times (d+1)$ matrix with each entry representing the probability that a bitstring with j ones mutates to a string with i ones. Equation 6 is the formula from [4] where $\delta_{x,y}$ is the Kronecker delta function and q is the mutation probability. Note that like U , \hat{U} is a column stochastic matrix. Each column sums to 1. Equation 7 contains a three bit example of \hat{U} where $q=1/3$.

$$\hat{U}_{i,j} = \sum_{k=0}^{d-j} \sum_{l=0}^j \delta_{j+k-l,i} \binom{d-j}{k} \binom{j}{l} q^{k+l} (1-q)^{d-k-l} \quad (6)$$

$$\text{where } \delta_{j+k-l,i} = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}$$

$$\hat{U} = \begin{pmatrix} .296 & .148 & .074 & .037 \\ .444 & .444 & .333 & .222 \\ .222 & .333 & .444 & .444 \\ .037 & .074 & .148 & .296 \end{pmatrix} \quad \text{where } q = \frac{1}{3} \quad (7)$$

3.1 Example Functions of Unitation

The three fitness functions pictured in Figure 1 are called NEEDLE, BINEEDLE and ONEMAX, and have been theoretically studied for parameter static simple GAs by Rowe [3] and Wright [4]. For Figure 1, values of $d=10$ and $\alpha=10$ are used for NEEDLE and BINEEDLE.

$$\begin{aligned} \text{NEEDLE} &= \begin{cases} 1 + \alpha & \text{all ones string} \\ 1 & \text{otherwise} \end{cases} \\ \text{BINEEDLE} &= \begin{cases} 1 + \alpha & \text{all ones string} \\ 1 & \text{otherwise} \\ 1 + \alpha & \text{all zeros string} \end{cases} \\ \text{ONEMAX} &= \text{number of ones in string} \end{aligned} \quad (8)$$

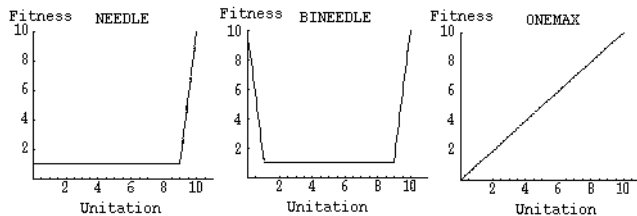


Figure 1. NEEDLE, BINEEDLE and ONEMAX fitness functions.

4. FIXED POINTS AS A FUNCTION OF MUTATION RATE

At this point, fixed points of the infinite population model can be calculated for various functions of unitation. Rather than simply choosing a few values of q and finding the fixed point in the simplex, these distributions will be calculated over a range of q . This gives the reader a visual picture of how the fixed point distributions change as the mutation rate changes. Note that all fitness functions used were changed slightly to contain no zero entries (a small ϵ is used) to avoid not meeting the conditions for Perron-Frobenius.

4.1 Fixed-point Surfaces for ONEMAX

Figure 2 shows two fixed points for the ONEMAX fitness function. These are the normalized leading eigenvector of G with mutation rate $q=0.005$ and $q=0.03$ both with $d=10$, providing 11 unitation classes. For example, they show that for $q=0.005$ approximately 62% of the population should contain strings with 10 bits of value 1 after a sufficiently large number of generations of the mutation-selection GA have been computed using a very large population. These values of q were chosen to highlight the movement of the distribution off of the global maxima. At $q=0.03$ only 10% of the population contains strings of 10 bits of 1.

Next, a sequence of fixed points were computed for mutation rates starting at $q=0.01$ and the population distributions were plotted as a 3-dimensional surface. Figure 3 shows the ONEMAX fixed points plotted from $q=0.01$ to $q=0.20$.

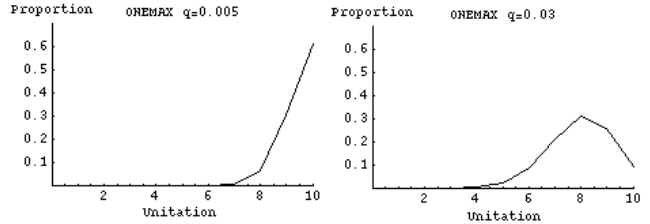


Figure 2. ONEMAX fixed point distributions for $q=0.005$ and $q=0.03$.

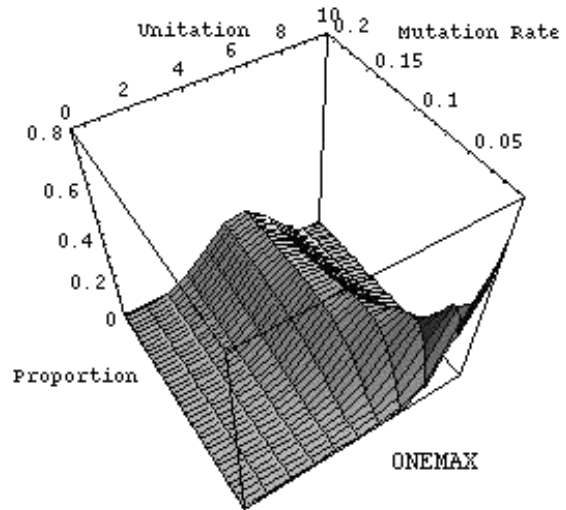


Figure 3. ONEMAX fixed point surface.

Note that at mutation rates near 0.01, the population contains a significant proportion of the $f(10)=10$ optimal mutation class. By $q=0.10$ there is near zero membership.

4.2 Fixed-point Surfaces for NEEDLE and BINEEDLE

Figures 4 and 5 contain the fixed point surfaces of NEEDLE and BINEEDLE for both $\alpha=5$ and $\alpha=15$. Figure 4 tells us that for low mutation rates (i.e. mutation rates below $q=0.10$ and $q=0.15$ respectively), NEEDLE has a significant proportion of the population at the maximum.

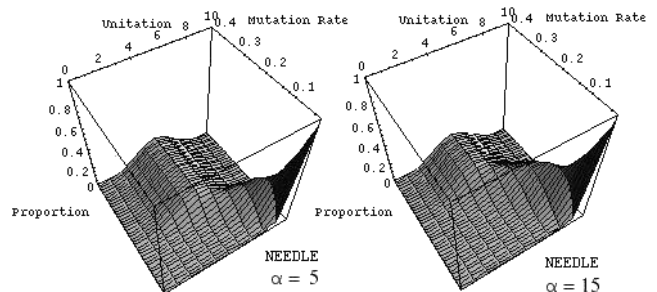


Figure 4. NEEDLE fixed point surfaces with $\alpha=5$ and $\alpha=15$.

In Figure 5 note the dramatic change in population distribution for BINEEDLE that occurs near $q=0.10$ (for $\alpha=5$). Above this mutation rate the population contains near zero proportional membership in either global optimum. This tells us that mutation

rates below this value are likely very important for good GA performance. For $\alpha=15$ this phase change occurs near $q=0.20$, indicating that this version of the fitness function is more tolerant of higher mutation rates. NEEDLE has similar properties.

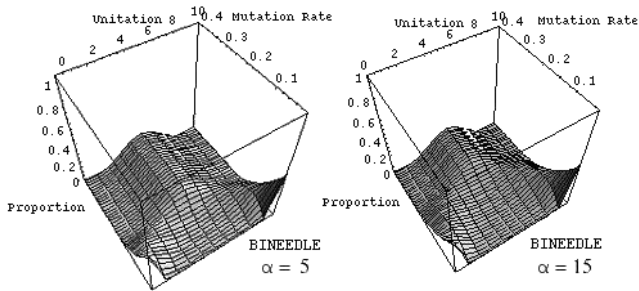


Figure 5. BINEEDLE fixed point surfaces with $\alpha=5$ and $\alpha=15$.

Notice that for mutation rates greater than the critical q values for both functions, the population is centered around the unitation midpoint, a string of 5 ones and 5 zeros.

4.3 Fixed-point Surface for a Fully Deceptive Trap Function

Trap functions are piecewise linear functions that divide the search space into two Hamming space basins [6]. Each basin has an optimal point, one of which is the global optimum. In Deb and Goldberg [6], a set of conditions for calling a fitness function 'fully deceptive' is given. A fully deceptive function from [6] is adopted here and is referred to as DECTRAP.

$$f(x) = 10 * \begin{cases} 1 & \text{if } u(x) = d \\ 1 - \frac{1+u(x)}{d} & \text{otherwise} \end{cases} \quad (9)$$

Figure 6 illustrates DECTRAP and its fixed-point surface. The trap function is at fitness of 9 for the all zeros string, and is fitness 10 for the all ones string. The all zeros basin takes up the majority of the function space.

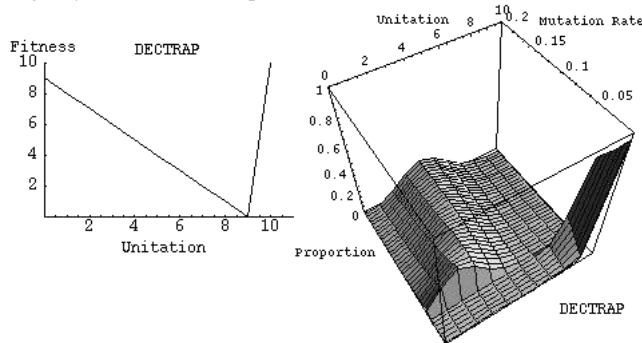


Figure 6. Fully deceptive trap functions DECTRAP and fixed point surface.

The fixed-point surface has a drastic phase change at approximately $q=0.02$. Below this mutation rate a high proportion of the globally optimal string exists. Above this mutation rate the fixed point contains nearly zero proportion of the global optimal. Notice again that as the mutation rate

increases, the fixed point moves towards a population centered around the unitation midpoint.

4.4 Fixed-point Surface for Functions with Two Traps

Figure 7 illustrates a trap function containing two traps, referred to as 2TRAP. The fixed point surface is very similar to the BINEEDLE surface with a critical phase change at $q=0.10$. Note that mutation rates below 0.05 are slightly superior since the fixed points are still centered much closer to the two local optima. Between 0.05 and 0.10 the population clusters are moving away from the maximums, as the mutation rates are too high for the population to maintain high membership.

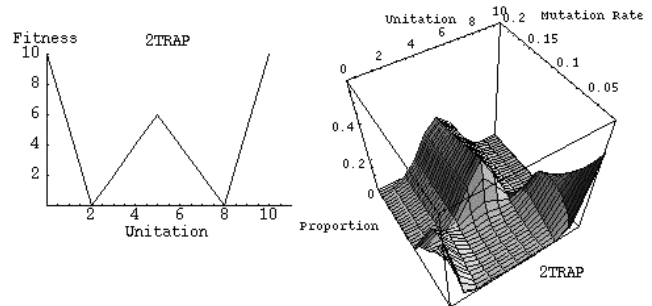


Figure 7. Double trap function 2TRAP and fixed point surface.

Figure 8 shows a deceptive double trap function, or DEC2TRAP. This function is modeled after the fully deceptive function given in [6]. The formula is given here:

$$f(x) = 10 * \begin{cases} 1 & \text{if } u(x) = d/2 \\ 1 - \frac{1+u(x)}{d/2} & \text{if } u(x) < d/2 \\ \frac{u(x) - d/2 - 1}{d/2} & \text{if } u(x) > d/2 \end{cases} \quad (10)$$

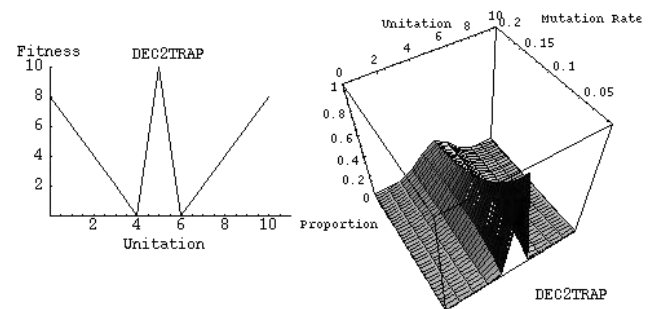


Figure 8. Deceptive double trap function DEC2TRAP and fixed point surface.

DEC2TRAP's fixed point landscape is very interesting in that it has virtually no membership of the $u(x)=0$ and $u(x)=10$ high fitness points. It varies between 10^{-13} at $q=.001$ to 10^{-2} at $q=.020$. This result is counter-intuitive to the expectations of a practitioner. The regions on either side of the center optimal needle have smooth hills to climb that lead to the local maximums. A practitioner might expect empirical GAs to retain membership in the local optima, given that low mutation rates make it harder for a population to move outside the basin of either

local maxima. Lack of elitism and other advanced features in the model partially explains the result, as well as the fact that after a high enough number of generations has passed, the Simple GA will converge totally into the basins of global maximums. Another contributing factor is that there are many strings of fitness $u(5)=1$ and only one string in the two local maxima classes.

5. (1+1) EAs AND VARIANTS

Next, we briefly review three simple variants of the genetic algorithm applied to functions of unitation and their associated Markov models. These algorithms and models have been studied extensively. Examples are [8][9][11].

5.1 Algorithms

Algorithm 1 is the (1+1) EA and its properties are summarized extensively in [8]. Again, the fitness function is $f(x):\{0,1\}^d \rightarrow \mathbb{R}$.

Algorithm 1: The (1+1) EA

1. Choose $p(n) \in (0,1/2]$
2. Choose $x \in \{0,1\}^d$ uniformly at random.
3. Create y by flipping each bit of x independently with $p(n)$
4. If $f(y) \geq f(x)$, set $x := y$
5. Continue at line 3

If we remove the restriction in line 4 that the child bitstring y must have a better fitness than the parent bitstring x , we create the (1,1) EA. This algorithm is more precisely denoted a ‘random walk’ and is commonly covered in introductory stochastic modeling texts like [12]. The U and \hat{U} matrices are the Markov transition matrices of the (1,1) EA for standard and unitation (respectively) fitness functions.

Equation 11 details how Equation 6 is used to construct the (1+1) EA transition matrix. Equation 12 is an example of the \hat{U}^+ transition matrix for a 3-bit bitstring with $q=1/3$ with a 3-bit ONEMAX fitness function. Note that these matrices are lower-triangular since they are built in a column stochastic manner.

$$\hat{U}^+_{i,j} = \frac{\mu_{i,j}}{\sum_{k=0}^i \mu_{i,k}} \ni \mu_{i,j} = \begin{cases} U_{i,j} & f(j) \leq f(i) \\ 0 & \text{else} \end{cases} \quad (11)$$

$$\hat{U}^+ = \begin{pmatrix} .296 & 0 & 0 & 0 \\ .444 & .522 & 0 & 0 \\ .222 & .391 & .750 & 0 \\ .037 & .087 & .250 & 1 \end{pmatrix} \quad \text{where } q = \frac{1}{3} \quad (12)$$

The matrix in Eq. 12 is an absorbing Markov model. Once the transition is made from some bitstring to the optimal string of all ones, all further transitions stay at this string. This is in contrast to Eq. 7, where transition from any unitation class to any other class is possible. Figure 9 is the state diagram for Eq. 12.

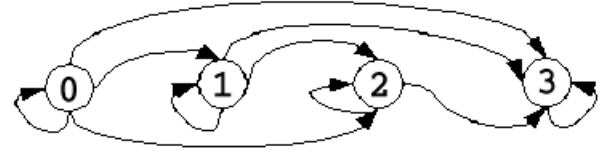


Figure 9. (1+1) EA example state diagram

The following algorithm is based upon one detailed and analyzed in [9].

Algorithm 2: The (1+1) Metropolis EA

1. Choose $p(n) \in (0,1/2]$
2. Choose $\alpha \in (1,\infty)$
3. Choose $x \in \{0,1\}^d$ uniformly at random.
4. Create y by flipping each bit of x independently with $p(n)$
5. If $f(y) \geq f(x)$, set $x := y$
6. Else set $x := y$ with probability $1/\alpha^{f(y)-f(x)}$
7. Continue at line 3

This algorithm allows selection of child bitstrings with lower fitness values. The probability of selection is in inverse exponential proportion to the drop in fitness. Algorithm 2 is an example of the Metropolis algorithm of [10]. The classic Metropolis algorithm restricts line 4 by mutating only one randomly chosen bit of x . The constant α is always greater than 1, usually much greater. Equation 13 details the construction of the (1+1) MEA transition matrix and Equation 14 is an example matrix for a 3 bit bitstring with $q=1/3$ and $\alpha=1024$ for a 3-bit ONEMAX fitness function.

$$\hat{U}^+_{i,j} = \frac{\mu_{i,j}}{\sum_{k=0}^i \mu_{i,k}} \ni \mu_{i,j} = \begin{cases} U_{i,j} & f(j) \leq f(i) \\ \frac{1}{\alpha^{f(j)-f(i)}} & \text{else} \end{cases} \quad (13)$$

$$\hat{U}^+ = \begin{pmatrix} .296 & .001 & \epsilon & \epsilon \\ .444 & .522 & .001 & \epsilon \\ .222 & .390 & .750 & .003 \\ .037 & .087 & .250 & .996 \end{pmatrix} \quad \text{where } q = \frac{1}{3} \quad (14)$$

and $\epsilon \approx 10^{-6}$

The matrix \hat{U}^+ contains what can be called a probabilistically absorbing state at state 3. The algorithm has a small probability of leaving the state, and if it does will likely rapidly return and stay there for long periods of time.

Algorithm 2 can also be converted into a simulated annealing algorithm by having α be a function of time $\alpha(t)$ that decreases as time increases [9][13].

Both Algorithm 1 and 2 are modeled by applying two operators together. The Markov transition generated by each has both a mutation component that is exactly the same as \hat{U} . The matrices in Equations 12 and 14 are really a composition of $P \hat{U}$ where P is a matrix that implements an elitist selection strategy. Here there are three different versions of P , one for each of Equations 12 and 14. One implements line 4 of Algorithm 1 and the other implements Algorithm 2's line 5.

5.2 Modeling a $(\mu+\mu)$ Metropolis EA

We now have a matrix associated with Algorithm 1 that we could apply the infinite population model selection matrix to via \hat{U}^+S . This new transition matrix incorporates both the non-decreasing selection of the (1+1) EA and the proportional selection matrix S , and is capable of modeling the behavior of the $(\mu+\mu)$ EA (see [8]) where μ is large. Unfortunately we cannot use the Perron-Frobenius theorem to analyze this matrix. It is a reducible matrix since \hat{U}^+ contains an absorbing state at location $\hat{U}^+_{d,d}$.

Algorithm 3 is an extension of Algorithm 2 to support populations greater than 1. We will refer to it here as the $(\mu+\mu)$ MEA. Conceptually this new population version of Algorithm 2 implements a Simple GA with the slight modification that generational replacement is done with a Metropolis style selection scheme.

Algorithm 3 $(\mu+\mu)$ MEA

1. Choose $p(n) \in (0, 1/2]$
2. Choose $\alpha \in (1, \infty)$
3. Choose $\mu \in (1, \infty)$
4. Create a population P of size μ and initialize each member uniformly at random where $P[i] \in \{0, 1\}^d$.
5. For $i=1$ to μ do
6. Select an individual x from P via proportional selection
7. Create y by flipping each bit of x independently with $p(n)$
8. If $f(y) \geq f(x)$, set $P_{new}[i] := y$
9. Else set $P_{new}[i] := y$ with probability $1/\alpha^{|f(y) - f(x)|}$
10. Set $P := P_{new}$
11. Continue at line 5

The '+' notation here is kept to remain consistent with the notation used in [9]. One can make an argument that since the transition matrix of the (1+1) MEA supports going from any string to any other string it should have been referred to in [8] as a variant of a (1,1) EA. We keep the '+' notation to stress the connection with the standard (1+1) EA and use the MEA to denote the Metropolis modification.

It is important for the reader to realize two things about Algorithm 3. The first is that the algorithm has two selection operators. One of them is the Metropolis operator that acts on the choice of replacing the single parent with the mutated child, the other is the population-wide proportional selection operator that selects individuals to be parents for mutation. Equation 15 augments Equation 3 to supply this elitist operator P . The second thing to realize is that this algorithm does not implement truncation selection as is implied in the usual meaning of the $(\mu+\lambda)$ EA[8].

$$p_{t+1} = G(p(t)) = \frac{1}{\mu(p_t)} UPS p_t \quad (15)$$

6. INFINITE POPULATION MODELS

We now have a new model that is amenable to applying the Perron-Frobenius theorem as was done in section 4. An important question is how the new fixed points will compare with those of the no-crossover Simple GA. This new model will be referred to

as the $(\mu+\mu)$ IPMEA (Infinite Population Metropolis EA). More correctly, $\mu = \infty$ since this is an infinite population model. The μ notation is kept to stress the linkage between infinite population models and real GAs with large populations.

6.1 Metropolis Fixed Points for ONEMAX, NEEDLE and BINEEDLE

Figure 10 gives the infinite population model fixed point surface of the $(\mu+\mu)$ IPMEA for $q \in [0.01, 0.4]$ and $\alpha=1024$. Note that the distribution for the EA contains $1-\epsilon$ proportion of the all-ones string for very small values of q , and this proportion shrinks to about 80% by $q=0.30$ before rapidly decreasing to 58% at $q=0.40$. Notice also that the proportion of the second best string, containing $d-1$ ones, is steadily increasing as the mutation rate grows, growing to about 40% by $q=0.40$

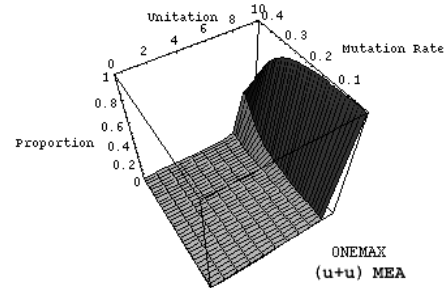


Figure 10. ONEMAX Metropolis EA fixed point surface.

It is not well visible in the figure, however by $q=0.40$ the proportion of the $d-2$ ones string grows to nearly 3%. Each successively less fit state's proportion decreases by a factor near 10^{-3} . Peeking back at the matrix in Equation 14 we can see the cause of this. As the mutation rate grows the probability of making a mutation transition from the 'all ones' string to the 'd-1 ones' string grows as well. Adding proportional selection mitigates this effect to a degree, however it still remains true when we move from the (1+1) MEA to the $(\mu+\mu)$ IPMEA. More importantly the probability of staying in the 'd-1 ones' state is fairly high.

This interesting drop off of the optimal proportion at about $q=0.3$ was curious enough to merit a full surface computation from $q \in [0.01, 0.99]$ using 80 discrete mutation rates in this range. Normally mutation rates above 0.5 are ignored by the EA community, they tend to result in optimizers that do not do much optimizing. The resulting Figure 11 has interesting structure. A partial explanation may be that at certain points in the mutation range the negative effect of a high mutation rate and positive effect of proportional selection cancel each other and come to a stalemate of two strongly attracting states.

The phase transitions displayed were not expected, and the intuitive expectation was a gentle curve that sloped down to a high membership in the half ones string while retaining the greatest membership in the all ones string.

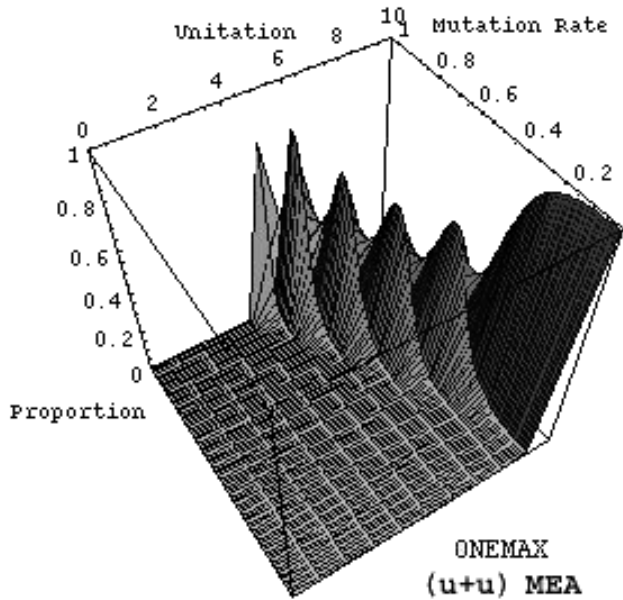


Figure 11. ONEMAX ($\mu+\mu$) IPMEA over full range of q

Figure 12 displays the fixed point surfaces for the NEEDLE and BINEEDLE functions. Note that the phase transitions displayed in Figures 4 and 5 are gone. The rising mutation rates were not enough to overcome the effect of proportional selection in favoring copies of the optimum strings, the difference in fitness between the optimum(s) and all other strings is too great. This result is a clear example of a situation where the technically (provably) non-convergent IPMEA algorithms are indeed strongly convergent in a practical sense.

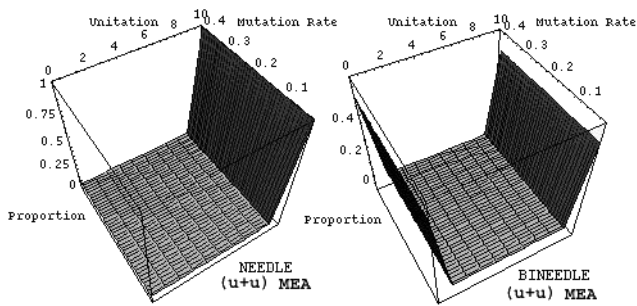


Figure 12. NEEDLE and BINEEDLE Metropolis EA fixed point surface.

6.2 Metropolis Fixed Points for the Trap Functions

The next sequence of figures revisits the fitness functions from section 4. Figure 13 gives the fixed point surface for the deceptive trap function DECTRAP. The results for the ($\mu+\mu$) IPMEA are interesting if somewhat expected. The results from the NEEDLE apply since the fitness difference between the all-ones string and the $d-1$ ones string is the same. This gives a transition matrix that is dominated at low mutation rates by the action of the Metropolis selection plus the proportional selection operator. Looking back at Figure 6, the SGA DECTRAP function

showed a phase transition at about $q=0.05$ moving the bulk of the population from the global fixed point to a population centered most of the way up the basin of the zero string. Here the phase change is pushed back to about $q=0.30$ and above that rate the global string still retains a decent proportion in the population.

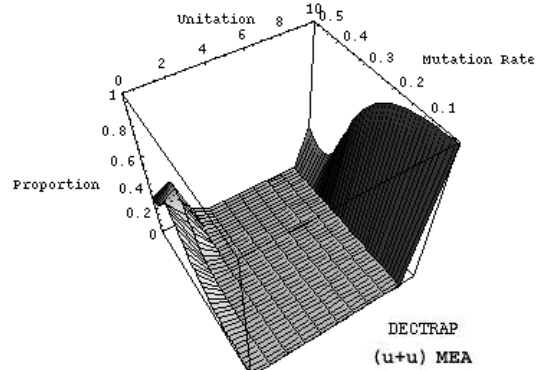


Figure 13. DECTRAP Metropolis EA fixed point surface.

Figure 14 contains the results of the IPMEA algorithm applied to the 2TRAP function. The nearly absorbing states at the all ones and all zeros strings have a very low probability of moving to lower fitness states. The effect of the double selection is too great for the mutation operator to overcome. This is in contrast to Figure 6 where there is a strong phase change as the mutation rate is about $q=0.07$. The elitism of this algorithm eliminated this phase change.

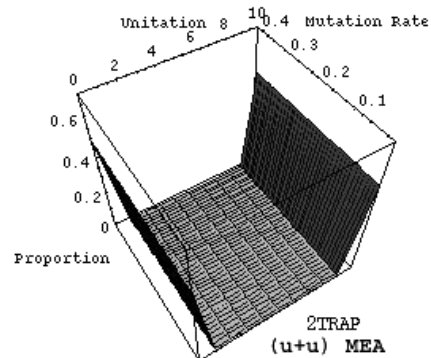


Figure 14. 2TRAP Metropolis EA fixed point surface.

Figure 15 displays the surfaces produced by the algorithm on DEC2TRAP. Figure 8 showed that the SGA had a slow degradation of the quality of convergence to the global optima, and it is eliminated with these elitist algorithms. It is also worth noting again that this function is comparatively easy to optimize as result of the binomial distribution of the unitation classes. If DEC2TRAP were formed by the opposite concatenation of two DECTRAP functions we would expect to see behavior very similar to that of 2TRAP.

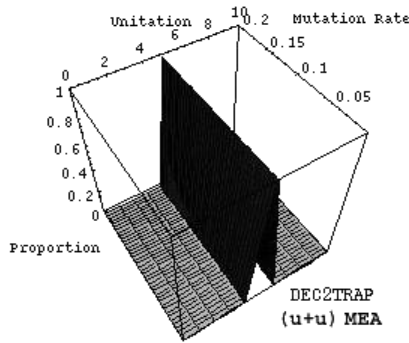


Figure 15. DEC2TRAP Metropolis EA fixed point surface.

7. A NOTE ON EMPIRICAL VALIDATION AND LIMITATIONS

In [3] Rowe validated the correctness of this type of analysis by conducting several experiments on real GAs with large populations optimizing functions of unitation. He showed that real populations do converge to the theoretical fixed points predicted by the model. Space considerations eliminated similar experiments here. It will be interesting to study the effect of various population sizes and their convergence to these fixed points.

This analysis has several limitations. It only applies to non-crossover GAs of a specific type. Crossover is a quadratic operator and in general the addition of crossover to the Vose model makes it not analytically solvable. This model also does not hold for GAs with true elitism. This analysis also only applies to functions of unitation, other families of functions cannot necessarily be reduced dimensionally to allow easier matrix computations.

8. CONCLUSIONS

Using the techniques that find fixed-points of dynamic systems models of GAs a number of fitness functions were studied to examine how those fixed points move as the mutation rate changes. In addition two new infinite population models were built to examine the performance of simple elitist GAs on these same functions.

It is common in GA research papers to use a specific mutation rate with no justification for the value chosen. Hopefully this paper will cause the reader to question the validity of the mutation rate they choose for their next GA project. The results shown here even call into question the often cited $1/d$ (or $1/l$) from Bäck [14]. As we have seen this rate of $q=0.10$ for the functions studied here would place the fixed-point dynamics above the phase change observed for several functions. In addition the value of even simple probabilistic forms of elitism is underscored.

In the future we hope to explore more extensions of common simplified GA models into a infinite population model framework and show how these models relate to commonly studied Markov models. It is also hoped that an examination of the various

schemes for dynamically adjusting the mutation rates of EAs during a run will come out of this first step. It is also likely that many more common forms of fitness functions have interesting structure when analyzed in this way.

9. ACKNOWLEDGMENTS

Thanks to Casey Gerstle for proof reading help and thanks to the reviewers for their insightful comments.

10. REFERENCES

- [1] Vose, M.D. *The Simple Genetic Algorithm: Foundations and Theory*, MIT Press, Cambridge, MA, 1999.
- [2] Strogatz, S.H., *Nonlinear Dynamics and Chaos*. Addison-Wesley, Reading, MA, 1994
- [3] Rowe, J. E. The Dynamical Systems Model of the Simple Genetic Algorithm, *Theoretical Aspects of Evolutionary Computing*, L. Kallel, B. Naudts, and A. Rogers, eds., Springer Verlag, New York, (2001) 31-57.
- [4] Rowe, J.: Population fixed-points for functions of unitation. In *Foundations of Genetic Algorithms. Vol. 5*. Colin Reeves, and Wolfgang Banzhaf (eds.). Morgan Kaufmann, (1998) 69-84.
- [5] Wright A.H., Rowe J.E., Neil, J.R.: Analysis of the Simple Genetic Algorithm on the Single-peak and Double-peak Landscapes, *Proc. of the 2002 Congress on Evolutionary Computation*, Fogel et al. Editors, IEEE Press (1999), 214-219.
- [6] Deb, K., Goldberg, D.E.: Analyzing Deception in Trap Functions. In *Foundations of Genetic Algorithms, Vol. 2*. D Whitley, editor. Morgan Kaufmann, (1992): 93-108.
- [7] Brin, M., Stuck, G., *Introduction to Dynamical Systems*. Cambridge University Press, New York (2002).
- [8] Rudolph, G, *Convergence Properties of Evolutionary Algorithms*. Verlag Kovac, Hamburg, (1997)
- [9] Droste, S., Jansen, T., Wegener, I., Dynamic parameter control in simple evolutionary algorithms, *Foundations of Genetic Algorithms, Vol. 6*, Worthy Martin and William Spears (eds.) Morgan Kaufmann (2001), pp. 275-294.
- [10] Metropolis, A.W. Rosenbluth, M.N. Rosenbluth. A.H. Teller and E. Teller, "Equation of State Calculations by Fast Computing Machines," *J. Chem. Phys.* 21 (1953) 1087-1092.
- [11] Mühlenbein, How genetic algorithms really work. I. Mutation and hillclimbing, *Proc. PPSN II (Parallel Problem Solving from Nature)*, (1992), pp. 15-25.
- [12] H. M. Taylor and S. Karlin. *An Introduction to Stochastic Modeling*, 3th Ed. Academic Press, 1998.
- [13] Kirkpatrick, S., CD Gelatt Jr., MP Vecchi, "Optimization by Simulated Annealing", *Science*, 220, 4598, 671-680, 1983
- [14] Bäck, T. Optimal Mutation Rates in Genetic Search. *Proc. of the 5th Int. Conf. on Genetic Algorithms*, Morgan Kaufmann, (1993) pp. 2-8.