Searching for Diverse, Cooperative Populations with Genetic Algorithms

Robert E. Smith
Dept. of Engin. Mechanics
University of Alabama
Tuscaloosa, AL 35487
(205) 348-1618
resmith@ua1ix.ua.edu

Stephanie Forrest
Dept. of Computer Science
University of New Mexico
Albuquerque, NM 87131
(505) 277-7104
forrest@cs.unm.edu

Alan S. Perelson
Theoretical Division
Los Alamos National Laboratory
Los Alamos, NM 87545
(505) 667-6829
asp@receptor.lanl.gov

Running Head: Cooperative Populations with GAs

Keywords: genetic algorithms, classifier systems, fitness sharing, computational immunology, multi-modal search

Abstract

In typical applications, genetic algorithms (GAs) process populations of potential problem solutions to evolve a single population member that specifies an "optimized" solution. The majority of GA analysis has focused on these optimization applications. In other applications (notably learning classifier systems and certain connectionist learning systems), a GA searches for a population of cooperative structures that jointly perform a computational task. This paper presents an analysis of this type of GA problem. The analysis considers a simplified genetics-based machine learning system: a model of an immune system. In this model, a GA must discover a set of pattern-matching antibodies that effectively match a set of antigen patterns. Analysis shows how a GA can automatically evolve and sustain a diverse, cooperative population. The cooperation emerges as a natural part of the antigen-antibody matching procedure. This emergent effect is shown to be similar to fitness sharing, an explicit technique for multi-modal GA optimization. Further analysis shows how the GA population can adapt to express various degrees of generalization. The results show how GAs can automatically and simultaneously discover effective groups of cooperative computational structures.

1 Introduction

Maintaining diversity of individuals within a population is necessary for the long term success of any evolutionary system. Genetic diversity helps a population adapt quickly to changes in the environment, and it allows the population to continue searching for productive niches, thus avoiding becoming trapped at local optima. In genetic algorithms (GAs), it is difficult to maintain diversity because the algorithm assigns exponentially increasing numbers of trials to the observed best parts of the search space (cf. Schema Theorem (Holland, 1975)). As a result, the standard GA has strong convergence properties. For optimization problems, convergence can be an advantage, but in other environments it can be detrimental. Further, even in optimization, strong convergence can be problematic if it prematurely restricts the search space.

In many settings, convergence by the GA on a global optimum is not appropriate. For example, in a classifier system (Holland, Holyoak, Nisbett, & Thagard, 1986) genetic operators are a natural way to search for a useful set of rules that collectively performs well in a task environment, each rule playing a unique and complementary role. Thus, the system needs to evolve a set of rules that are specialized to various tasks (or niches) rather than producing a homogeneous (converged) population of similar rules. As a second example, consider a computational model of the immune system in which a population of antibodies is evolving to cover a set of antigens (Forrest & Perelson, 1991). If the antibody population is sufficiently large, it clearly makes sense to evolve antibodies that are specialized to recognize different classes of antigens instead of evolving one generalist antibody that weakly matches all antigens. For these more ecological environments, genetic operators are clearly relevant for evolving a good solution, if inappropriate convergence can be avoided.

Related to the diversity problem is the issue of generalization. In ecological settings, where a coadapted set of individuals is desired, the question of generalization arises when there are not enough individuals in the population to assign one individual (or subpopulation) to each task, niche, or fitness peak. In cases such as these, a reasonable response is for the population to evolve individuals that can cover more than one peak by becoming "generalists." Note, that in traditional GA optimization, the generalist usually does not arise, since the population tends to pick one fitness peak and specialize in optimizing it. However, for settings in which coverage of solutions is important (e.g., set cover problems, recognition problems, and almost any classifier system), it is imperative to have the capacity of evolving generalists. Ideally, the degree of generality of discovered solutions should be an emergent property of the system that depends on certain characteristics of the environment and parameters of the learning system.

To date, most GA analysis has focused on problems in which each population member's fitness is independent of other population members, thus excluding coevolutionary systems such as classifier systems and immune system models. In this paper, we introduce a simple model in which each individual's fitness is functionally dependent on the rest of the population, thus capturing one important aspect of the ecological problems in which we are interested. The functional dependence is introduced through the use of a simplified bidding mechanism similar to that of classifier systems. We show mathematically that the use of the simple bidding procedure combined with a traditional GA is sufficient for the population to discover and maintain independent subpopulations. Further, our analysis shows that the model implements a form of *implicit* fitness sharing which we relate to previous (explicit) models of fitness sharing. Finally, we show how certain parameters of the model can be varied to induce it to evolve more or less general solutions to the immune recognition problem. The theoretical results presented in this paper have been confirmed by experiments reported in Forrest, et al. (in preparation).

The remainder of this paper is divided into eight sections. Sections 2 - 4 provide background material on related work, a mathematical model of selection in the GA, and explicit fitness sharing. Section 5 introduces the immune system model that motivated the work reported in this paper. Section 6 analyzes our model mathematically and shows that it maintains stable subpopulations. Section 7 shows how the generality of solutions can be controlled in the model, and Sections 8 and 9 discuss the implications of our results.

2 Related Work

In optimization, when the GA fails to find the global optimum, the problem is often attributed to *premature convergence*, which means that the sampling process converged on a local rather than the global optimum. Several methods have been proposed to combat premature convergence in conventional GAs (Booker, 1982; Deb, 1989a; DeJong, 1975; Goldberg, 1989). These include restricting the selection procedure (crowding models), restricting the mating procedure (assortative mating, local mating, etc.), explicitly dividing the population into subpopulations (common in parallel GAs), and modifying the way fitnesses are assigned (fitness sharing).

Crowding induces niches by forcing new individuals to replace those that are similar genotypically (DeJong, 1975). This is accomplished by using a "steady-state" GA which creates new individuals one at a time, inserting them into the population by replacement of existing individuals. In the crowding algorithm, an individual is selected for replacement (to "die") by choosing a subset of the population randomly and then selecting the member of that subset that resembles the new

individual most closely. The various restricted mating schemes (assortative, explicit subpopulations, local, etc.) have a similar goal as crowding, but here the population is prevented from becoming too homogeneous by limiting the hybridization effects of crossover. Assortative mating algorithms, introduced by Booker (1982; 1985), restrict crossovers to occur between functionally similar individuals. They were invented in the context of classifier systems, where the notion of "functionally similar" has a natural interpretation as the set of classifiers that simultaneously match the message list. This method has worked well for classifier systems, but is unproven in the context of pure genetic algorithms.

Several parallel GAs have been studied in which the population is explicitly divided into smaller subpopulations (Cohoon, Hedge, & Martin, 1987; Gorges-Schleuter, 1989; Grosso, 1985; Pettey, Leuze, & Grefenstette, 1987; Tanese, 1989a; Tanese, 1989b; Whitley & Starkweather, 1990) . Each subpopulation is isolated from the others in the sense that it evolves independently with occasional migrations of individuals from one subpopulation to another. Based on concepts from population genetics, the idea here is that random genetic drift will cause each subpopulation to explore different regions of the search space and that migration will communicate important discoveries among the subpopulations. This method can make the algorithm more efficient at discovering global optima, but even a very small amount of migration results in the subpopulations eventually converging on one solution, so in the end diversity is lost (Grosso, 1985).

Local mating algorithms (Collins & Jefferson, 1991; Davidor, 1991; Hillis, 1990; Muhlenbein, 1989; Spiessens & Manderick, 1991) attempt to move away from explicitly subdividing the population (explicit subdivision implies some prior knowledge about how many niches are in the environment, and their relative size). In local mating, the population is arranged geometrically (e.g., in a two-dimensional plane) and crossovers occur only between individuals that are "near" one another geographically. Once again, the idea is that random genetic variation will lead to subgroups of individuals, each exploring different regions of the search space. These methods can slow down convergence time dramatically, but by themselves cannot maintain stable separate subpopulations (McInerney, 1992).

Another method for maintaining diversity in a classifier system is called the "Pitt" approach (DeJong, 1988). First used by Smith (1980), this method concatenates a population of classifiers to form a single individual for the genetic algorithm to manipulate. Thus, the genetic algorithm is acting on populations of rule sets. Although diversity can be maintained within a rule set using this method, there is an inherent inefficiency that results from maintaining populations of rule sets rather than populatins of rules. Interestingly, something resembling the Pitt approach exists in the genetic encoding of real immune systems (**?**)

Finally, fitness sharing (Deb & Goldberg, 1989a; Deb, 1989a; Goldberg & Richardson, 1987) (discussed in detail in Section 4) induces subpopulations by penalizing individuals for the presence of other similar individuals in the population, thereby encouraging individuals to find productive uncrowded niches. While the methods discussed in the previous paragraphs (except the Pitt approach) focus on the mechanics of selection and recombination in the GA, fitness sharing leaves the standard GA unchanged and simply modifies the way in which fitness values are computed. Fitness sharing works well, but has several important limitations:

- It requires a comparison of every population member to every other population member in each generation (N^2 comparisons where N is the size of the population).
- Setting the critical parameter σ_s requires knowledge about the number peaks in the space (Deb, 1989b).
- Setting σ_s is also dependent on a uniform distribution of peaks in the search space. Although Deb (1989b) shows experiments where sharing succeeds on a problem with mildly non-uniform peak separation, it is likely fitness sharing would overlook peaks that were distributed with less uniformity (K. Deb, personal communication, 1991). Goldberg (personal communication, 1992) has suggested that a "skewing factor" coupled with fitness scaling (**?**) could help sharing to maintain multiple, non-uniformly distributed peaks. However, these techniques have yet to be fully investigated.

In spite of these limitations, fitness sharing is important because it induces stable niches (rather than simply slowing down the time to convergence), and it incorporates the concept of limited resources into the GA model.

3 Preliminaries: The Effects of GA Selection on Population Composition

Consider a population evolving in time under the GA. The reproduction and survival of an individual into the next generation depends on the individual's fitness. The exact details of how fitness affects reproduction varies among different implementations of the GA (Goldberg & Deb, 1990). Here we illustrate the effects of fitness-proportionate selection among a set of 2^ℓ individuals in a selection-only GA.[1] The individuals can be thought of as completely specified GA population members (bit

[1]Fitness-proportionate selection was adopted for several reasons: (1) it is the simplest and most widely known method of selection, (2) it is easily related to the Schema Theorem and has the important property of increasing good

strings of length ℓ), or as competing schemata of order ℓ. Let P_i^t be the proportion of individual i in the population at time t ($\sum_{i=1}^{2^\ell} P_i^t = 1$), and let f_i be the expected fitness of individual i, or in the case of Schema i, the observed fitness. Assume (as is typical in GA optimization applications) that f_i does not depend on the population proportions. Under fitness-proportionate selection, the value of P_i at time $t+1$ is given by

$$P_i^{t+1} = \frac{P_i^t f_i}{\sum_{j=1}^{2^\ell} P_j^t f_j} , \tag{1}$$

where the denominator in Eq. (1) is the mean fitness of individuals in the population at time t. It is easy to see that iteration of this equation assigns an increasing number of copies to individuals that have fitness higher than the mean. However, as the proportion of individuals with high fitness increases, the mean fitness will also increase. As a consequence, ultimately only those individuals with the highest fitness (i.e., with fitness f^*, where $f^* \geq f_i$ for all i) will increase. Let the set of individuals with fitness f^* be called $\mathcal{S}^*$. Then, the iteration of Eq. (1) will converge to a state in which the entire population is composed of individuals from $\mathcal{S}^*$. At steady state, where $P_i^{t+1} = P_i^t \equiv P_i^{ss}$ for all i,

$$P_i^{ss} = \frac{P_i^{ss} f_i}{\sum_{j=1}^{2^\ell} P_j^{ss} f_j} .$$

By the previous argument, $P_i^{ss} \neq 0$ only for those individuals in $\mathcal{S}^*$. Thus, the sum of proportions of individuals in $\mathcal{S}^*$ must equal one. Inserting these conditions into the equation above gives

$$1 = \frac{f^*}{\sum_{j \in \mathcal{S}^*} P_j^{ss} f^*} = \frac{f^*}{f^*} ,$$

which shows that *any* distribution of proportions of individuals in $\mathcal{S}^*$ is a valid steady state. In practice, *genetic drift* tends to drive a GA population to a population comprised entirely of one type of individual from $\mathcal{S}^*$. This effect has been discussed extensively in population genetics and in the GA literature (Goldberg & Segrest, 1987). For example, consider a population composed of two types of individuals, 1 and 0, each present at 50 copies and having equal fitness. If a sample of the population is chosen for reproduction and replacement of their parents so that the population size is maintained constant, then after one generation the population may be skewed randomly to say 52 1s and 48 0s. In the next round, 1s are more likely to be chosen for reproduction than 0s due to their greater representation in the population, and further skewing can occur. Thus even in the absence of fitness differences between two or more individuals, the slightest nonuniformities of distribution can be amplified by the reproduction scheme. Mutation and crossover can also randomly change

schemas exponentially, and (3) many other selection methods are derived from fitness-proportionate selection.

the population composition and can contribute to drift. Therefore, by random drift a population comprised of only one type of individual inevitably occurs.

In the above derivation we assumed that the fitness values were not a function of the proportion values. If fitness values are a function of population proportions, then convergence to a single type of individual can be avoided. In the next section we review a technique called fitness-sharing (Deb, 1989b; Deb & Goldberg, 1989b; Goldberg & Richardson, 1987) that explicitly relates fitness values to proportions to overcome the convergence associated with fitness-proportionate GA selection in multi-modal problems.

4 Fitness Sharing

Fitness sharing (Deb & Goldberg, 1989a; Deb, 1989a; Goldberg & Richardson, 1987) induces diversity in a population by penalizing individuals for the presence of similar individuals in the population. In ecological theory such strategies are called negative frequency-dependent selection (Levin, 1981). Fitness sharing is based on the idea of environmental niches with finite resources available for each niche. As the number of individuals in a given niche increases, the availability of resources in the niche decreases, resulting in an effective decrease in the viability of individuals in the niche, and the subsequent decrease in their numbers. To maintain a viable population in a niche, the population size must come into equilibrium with the availability of resources.

To illustrate the effects of sharing resources, consider the effect of replacing an individual's fitness by a *shared* fitness $f_i' \equiv f_i / P_i^t$. This shared fitness is the value used in fitness-proportionate selection. Thus, equation (1) becomes

$$P_i^{t+1} = \frac{P_i^t f_i'}{\sum_{j=1}^{2^\ell} P_j^t f_j'} \ , \tag{2}$$

or

$$P_i^{t+1} = \frac{f_i}{\sum_{j=1}^{2^\ell} f_j} \ . \tag{3}$$

Under such a scheme, selection assigns each individual a population proportion equal to its fitness relative to the sum of all fitness values *in one step*, and from then on, the proportions are stable.

In a multi-modal search problem, it would be desirable for the population to cluster around several peaks, rather than simply distributing itself based on relative fitness, as in the previous calculation, or converging to a single peak, as would a typical GA with fitness-proportionate selection. The mechanisms of fitness sharing used in previous studies (Deb, 1989b; Deb & Goldberg, 1989b; Goldberg & Richardson, 1987) cause population members to cluster around

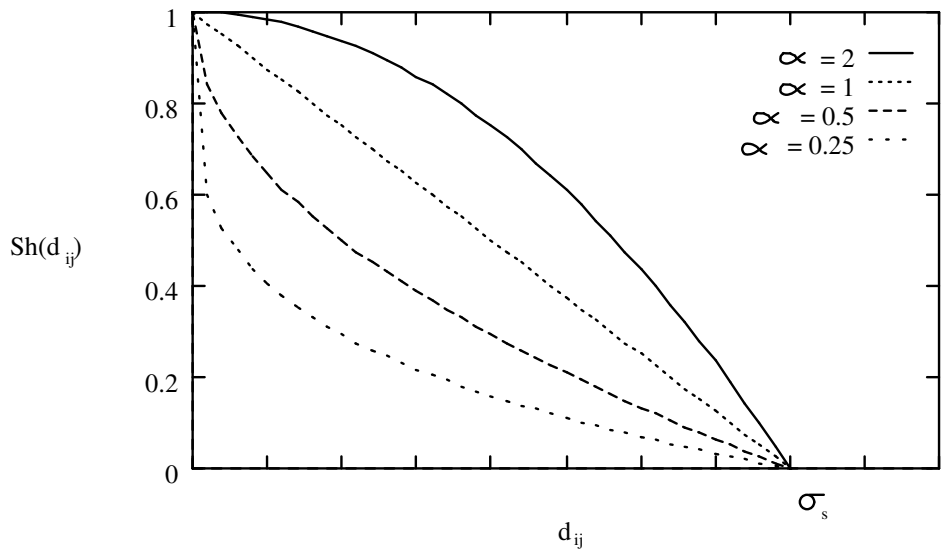

Figure 1: The sharing function $Sh(d_{ij})$ for various values of α. Note that the function is zero for d_{ij} greater than σ_s.

peaks in a search space by reducing a population member's fitness to account for the proximity of other individuals in the population (under some distance metric). Specifically, they calculate the shared fitness as

$$f_i' = \frac{f_i}{\sum_{j=1}^{N} Sh(d_{ij})} , \tag{4}$$

where d_{ij} is the distance between i and j under a given metric, $Sh(d_{ij})$ is the *sharing function*, given by

$$Sh(d_{ij}) = \begin{cases} 1 & \text{if } d_{ij} = 0 \\ 1 - \left(\frac{d_{ij}}{\sigma_s}\right)^{\alpha} & \text{if } d_{ij} < \sigma_s; \\ 0 & \text{otherwise} \end{cases} , \tag{5}$$

N is the number of individuals in the population, and σ_s and α are parameters[2]. Figure 1 depicts various sharing functions. Note that since the distance between any individual and itself, $d_{ii} = 0$, at least one term in the denominator of Eq. (4) is equal to one. Since each individual contributes one term to the sum, having many individuals close together leads to a large denominator, and hence a small fitness.

The critical parameter in the fitness sharing technique is σ_s, which dictates a cutoff distance, beyond which no sharing will occur. Deb considers the case in which there are q peaks that one

[2]The condition $Sh(d_{ij}) = 1$ for $d_{ij} = 0$ is only implied in previous fitness sharing studies. It is included here to clarify the limiting case where $\sigma_s = 0$.

wishes the GA to find and maintain, and these peaks are equidistant (in terms of the given distance metric) from one another in the search space (Deb, 1989b). He also shows how to set σ_s such that q hyperspheres of radius σ_s fill the search space. Then, under Deb's assumptions, one and only one peak lies in each hypersphere of radius σ_s.

To understand the effects of fitness sharing, consider a problem where Deb's assumptions hold, and σ_s is set appropriately. Imagine a GA population that begins with an equal number of representatives of each point in the search space (a flat population). In the first iteration of Eq. (2), where f' is the shared fitness given by Eq. (4), each individual's fitness f_i is divided by the same number, since each individual has the same distribution of "neighbors" in its σ_s hypersphere. Thus, the relative values of fitness are not changed by sharing in this step, and we can expect that individuals at the q peaks will increase in their numbers. In the next iteration, individuals at the q peaks have increased numbers relative to the rest of the population, therefore their shared fitness values are lowered. Note, however, that individuals at any given peak are affected by the proportion of individuals at or near that peak only, due to the setting of σ_s and individuals at different peaks are assured not to affect one another's shared fitness values. Now consider an individual that is not on but is *near*, i.e. within σ_s, of one or more of the q peaks. In the second application of selection, such individuals have fitness values that are decreased by the strong proportions of individuals at *one or more* peaks. Thus, the relative character of the fitness landscape (with peaks remaining effectively higher than non-peaks) is maintained, and the GA will move towards a population clustered at the q peaks. After all individuals are at these peaks, their shared fitness values must all be equal to maintain equilibrium under fitness proportionate selection. Thus, one can expect that the proportion of individuals at each peak will be directly related to that peak's relative height.

The previous arguments point out the importance of σ_s's value, and the assumptions used in setting this value. If σ_s is too large, and two peaks are within σ_s of one another, individuals at these peaks will affect one another's shared fitnesses, and one can expect that the GA will converge to a population that contains one or the other of these peaks, but not both. The limiting case of this behavior is where σ_s equals the maximum possible distance between two individuals (the radius of the search space). In this case, each individual's fitness is divided by the same value, the effects of sharing are eliminated, and one can expect the GA to converge towards one type of individual in the usual fashion. If σ_s is too small, one can expect that individuals near a peak may not be sufficiently affected by other nearby individuals, and that complete convergence to this peak may not occur (nearby individuals may remain in the final population). The limiting case of this behavior is where $\sigma_s = 0$, in which case each individual's fitness is only divided by its own proportion. Equation (3) shows that this can be expected to allot proportions to each individual based on fitness relative to

the sum of all fitnesses in one step, without a search for peaks.

Fitness sharing is an effective method for encouraging the population to maintain diverse subpopulations in all the high-fitness regions of the search space. However, as we mentioned earlier, it has several important limitations: (1) to set σ_s, the number of peaks must be known, (2) the peaks must be equidistant (or nearly so), and (3) the method requires N^2 comparisons on every generation. These limitations arise from the fact that fitness sharing is defined explicitly—what we would like is a method that avoids the explicit N^2 calculations and can discover for itself how many peaks are in the search space and can allocate trials appropriately. In the next section we introduce an algorithm for fitness sharing that does not require explicit construction of the sharing function. The method thus avoids the difficulty of appropriately choosing σ_s. Further, the method can solve problems in which the peaks are not equally spaced. Because the method arose in the context of studying how the immune system can maintain the diversity needed for it to detect multiple antigens, we briefly summarize the relevant biology before introducing the algorithm.

5 GA Simulations of the Immune System

Our immune system protects us from a wide variety of different viruses, bacteria, toxins, and other pathogenic organisms. Although there are many different host defense mechanisms employed, the first step in any of these mechanisms is the recognition of the foreign cell or molecule, which we call *antigen*. Recognition in the immune system occurs by a chemical interaction between antigen and a specific host defense molecule such as *antibody*. Thus, the problem that the immune system faces is the generation of a repertoire of antibodies with sufficient diversity to recognize any antigen.

Forrest, et al. (in preparation) study an abstract version of this problem in which antigens and antibodies are represented by bit strings of a fixed length ℓ. Recognition is assessed via a string matching procedure. The antigens are considered fixed, and a population of N antibodies is evolved to recognize the antigens using a GA. For any set of antigens, the goal is to obtain an antibody *cover*, defined to be a set of antibodies such that each antigen is recognized by at least one antibody in the population. Maintaining antibody diversity is crucial to obtaining a cover.

The initial model makes the important simplification that a bit string represents both the genes that code for a receptor and the phenotypic expression of the receptor molecule. The model includes only recognition of our idealized antigens by receptors and does not consider how the immune system neutralizes an antigen once it is recognized.

The model is based on a universe in which both antigens and receptors on B cells and T cells are

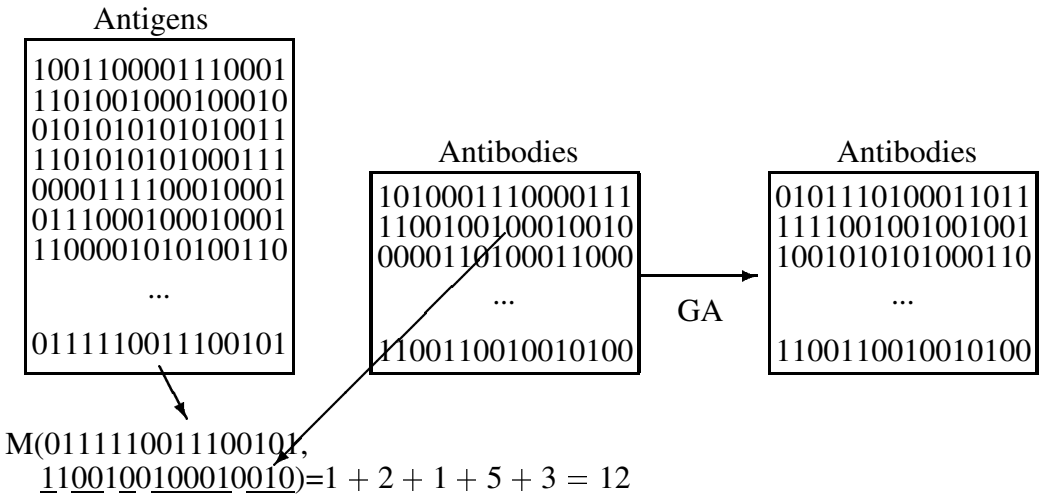

Figure 2: A schematic illustration of the immune model. The match score, M, of an antigen-antibody pair is the number of bits that are complementary.

represented by binary strings (Farmer, Packard, & Perelson, 1986). This is certainly a simplification from the real biology in which genes are specified by a four-letter alphabet and recognition between receptors and antigens is based on their three-dimensional shapes and physical properties. Further, the model does not distinguish between receptors on B cells and the soluble, secreted form of the receptor, which is antibody. However, the universe of binary strings is rich enough to allow one to study how a relatively small number of recognizers (the antibodies) can evolve to recognize a much larger number of different patterns (the antigens).

A receptor, or "antibody," is said to *match* an antigen if their bit strings are complementary (maximally different). Since each antibody may have to match against several different antigens simultaneously, we do not require perfect bit-wise matching. There are many possible match rules that are plausible physiologically (Perelson, 1989). The degree of match is quantified by a match score function $M : Antigen \times Antibody \rightarrow \Re$. This function identifies contiguous regions of complementary bitwise matches within the string, computes the lengths (l_i) of the regions, and combines them such that long regions are rewarded more than short ones. Using this basic idea, many different specific functions can be defined that are linear or nonlinear in l_i. We have studied the behavior of the model under several different functions (Forrest et al., in preparation).

Using the bit string representation for antibodies, we construct one population of antigens and one of antibodies. Antibodies are matched against antigens, scored according to a fitness function M, and replicated using a genetic algorithm. Figure 2 illustrates the basic immune model.

>From this basic model, many variations can be created by changing the details of how

antibodies are chosen to be matched against antigens. For example, we have used the basic model to study antigen populations that cannot be matched by a single antibody type. Suppose the the population of antigens is:

$$50\% \text{ - } 000\ldots000$$
$$50\% \text{ - } 111\ldots111.$$

In order for an antibody population to match these antigens perfectly, there would need to be some antibodies that are all 1s and others that are all 0s. Thus, a solution to this problem would require the GA to maintain two different solutions simultaneously. This is a simple example of a multiple-peak problem in which there are only two peaks and they are maximally different. A solution requires a population of antibodies that contains strings that are all 1s and all 0s. The hybrids formed by crossover are not useful. Maximally different peaks are used for illustration. When we have varied the Hamming distance between peaks experimentally, there is little if any effect on the performance of the model (Forrest et al., in preparation).

With a fixed set of antigens, the antibodies are initialized either to be completely random (to see if the GA can learn the correct antibodies) or initially given the answer by setting the population to include the correct antibodies (000...000 and 111...111 in the example). By giving the answer initially, the stability of the answer can be tested. Fitness scoring is as follows:

1. A single antigen is randomly selected from the antigen population.
2. From the population of N antibodies a randomly selected sample of size σ is taken without replacement.
3. For each antibody in the sample, *match* it against the selected antigen, determine the number of bits that match, and assign it a *match score*[3].
4. The antibody in the sample population with the highest match score is determined. Ties are broken at random.
5. The match score of the winning antibody is added to its fitness.
6. This process is repeated for C cycles.

In this scheme, the fitness values of antibodies are interdependent, since an antibody's proportion is only increased if it is the best matching antibody in the sample. Numerical experiments reported

[3]For the purposes of this discussion, details of the matching and scoring procedures are unimportant.

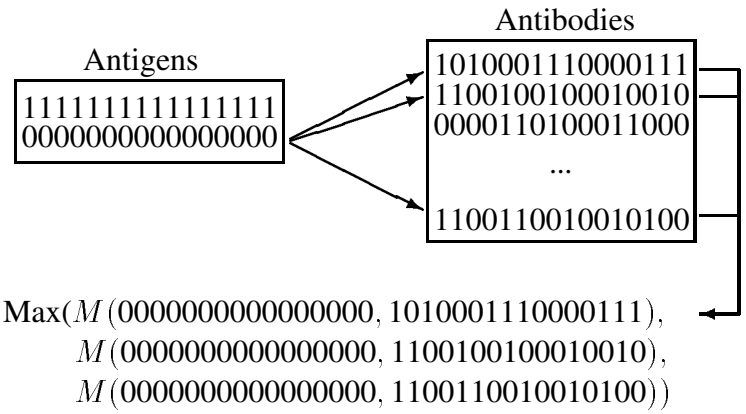

Figure 3: A schematic illustration of the two-peak problem. Each antigen corresponds to one peak. An antigen and a subset of the antibody population are each selected randomly; the antibody (from the subset) that matches the antigen best (Max) has its fitness incremented.

in Forrest et al. (in preparation) have shown that this scheme can maintain a diverse population of antibodies that cover a set of antigens, that the antibodies will occur with frequency proportional to the sampling rate for each antigen, and that the system performance seems relatively insensitive to the Hamming distance between these antigens. We will show that this procedure implicitly embodies fitness sharing. The process is iterated so that each antigen in the antigen population has a chance of being selected and each antibody in the antibody population will receive a fair evaluation of its fitness. Figure 3 illustrates the model.

This model corresponds quite closely to certain features of learning classifier systems. In effect, each antibody in our model is a highly simplified classifier rule with only a condition part (defined over the alphabet $0, 1$ rather than the more traditional $1, 0, \#$). The message-processing cycle is similar to classifier systems in that it uses a bidding mechanism. In our model, the bidding is deterministic (the closest match always wins), whereas many different bidding schemes are often used with classifier systems. Since we only allow one winner, this is analogous to a classifier system with a message list of size 1. Obviously, there are many aspects of classifier systems that our model does not incorporate (e.g., posting messages, bucket brigade learning, etc.), but even in this simplified form we can study the interaction between the genetic algorithm and bidding.

6 Emergent Fitness Sharing in the Immune System Model

To understand the mechanisms of maintaining population diversity in the immune system model, we first calculate an antibody's expected fitness. Some new notation is required. Let the distance between an antibody i and an antigen j be called d_{ij}. The following discussion assumes that d_{ij} is the number of bits of antibody i that do not match (i.e. are not complementary to) those in antigen j, although other distance metrics can be used in the following developments. Under this distance metric, antibody i and antigen j are said to *perfectly match* if $d_{ij} = 0$. The maximum distance possible between an antibody and an antigen is ℓ, the bit string length. Let $s(d_{ij})$ be the match score assigned to antibody i when it is matched against antigen j. Let $N_j(m)$ be the number of antibodies in the population that are at distance m from antigen j. Also let α_j be the probability of selecting antigen j for matching. Finally, let f_i be the expected fitness of antibody i.

Consider a given antigen j. Assume that the population is of size N, and that it contains $N_j(m)$ antibodies at distance m from antigen j. The probability that w antibodies at exactly distance m from antigen j are in a sample of size σ taken without replacement from this population, $p(w; \sigma, N, N_j(m))$, is given by the hypergeometric distribution (Freund, 1962; Hines & Montgomery, 1980)

$$p(w; \sigma, N, N_j(m)) = \frac{\begin{pmatrix} N_j(m) \\ w \end{pmatrix} \begin{pmatrix} N - N_j(m) \\ \sigma - w \end{pmatrix}}{\begin{pmatrix} N \\ \sigma \end{pmatrix}} , w = 0, 1, \ldots, \sigma \; . \tag{6}$$

Since the hypergeometric distribution will play an important role in subsequent calculations, it is important to understand how it arises. Think of the $N_j(m)$ antibodies at distance m as "successes" and the remaining $N - N_j(m)$ antibodies as "failures." We choose a sample of size σ without replacement and are interested in the probability of picking w success elements and hence necessarily $\sigma - w$ failure elements. There are $\binom{N}{\sigma}$ possible ways of picking a sample of size σ. The number of ways of picking w successes from a total of $N_j(m)$ elements is $\binom{N_j(m)}{w}$, whereas the number of ways of picking $\sigma - w$ failures from $N - N_j(m)$ elements is $\binom{N - N_j(m)}{\sigma - w}$. Thus, the fraction of times a sample is drawn with w success elements and $\sigma - w$ failure elements is given by Eq. (6).

It will be useful for later discussions to note two special cases. If the sample size $\sigma = 1$, then the probability that the sample contains an antibody at distance m is

$$p(1; 1, N, N_j(m)) = \frac{N_j(m)}{N} \; ,$$

and the probability that it does not contain such an antibody is

$$p(0; 1, N, N_j(m)) = \frac{N - N_j(m)}{N} ,$$

If the sample size $\sigma = N$, then

$$p(w; N, N, N_j(m)) = \begin{cases} 1 & \text{if } w = N_j(m) \\ 0 & \text{otherwise.} \end{cases}$$

6.1 Expected fitness of an antibody when perfect matching is required

To introduce the method of calculating the expected fitness of an antibody, we first consider the case in which an antibody receives a non-zero score only if it perfectly matches the antigen. Thus, the match score $s(d_{ij}) \neq 0$ if and only if $d_{ij} = 0$. Let the score received for a perfect match be s_p. For antibody i that perfectly matches antigen j to receive a score s_p at time t, the following conditions must be met:

(i) Antigen j must be the antigen selected for matching against. This occurs with probability α_j.

(ii) Antibody i must be in the sample of size σ.

(iii) If w antibodies in the sample perfectly match the antigen, antibody i must be chosen to be the tie breaker. This occurs with probability $1/w$.

Note from Eq. (6) that the probability that w antibodies in a sample of size σ are at distance 0 from antigen j is $p(w; \sigma, N, N_j(0))$. Given that w perfect matches against antigen j appear in the sample, the probability that this sample contains *one particular* antibody i that perfectly matches j is $p(1; w, N_j(0), 1) = w/N_j(0)$. Since events (i), (ii) and (iii) are independent, the probability that antibody i receives a non-zero score is

$$\alpha_j \sum_{w=1}^{\sigma} \frac{1}{w} \frac{w}{N_j(0)} p(w; \sigma, N, N_j(0)) = \alpha_j \frac{1}{N_j(0)} (1 - p(0; \sigma, N, N_j(0))).$$

Hence, the expected fitness of antibody i after one cycle is

$$f_i = \frac{s_p \alpha_j}{N_j(0)} (1 - p(0; \sigma, N, N_j(0))) .$$

Note that the expected fitness for C cycles would simply be Cf_i. Since C will be a common factor for all expected fitness values, it will not have a bearing on the expected behavior of fitness-proportionate selection, and, therefore, it is not considered in the subsequent discussion. Also, this equation assumes that all the antigens are distinct which for bit strings of reasonable length is a reasonable assumption.

In terms of our previous discussions on fitness sharing, the term $s_p\alpha_j$ roughly corresponds to the height of the fitness function at point j in the sequence space, i.e., in the ℓ-dimensional hypercube. The expected fitness calculation indicates that this value is divided by the proportion of individuals at that point, $N_j(0)$. This corresponds to explicit fitness sharing with $\sigma_s = 0$, where an individual's fitness is divided by the proportion of identical individuals in the population. The final hypergeometric term in the calculation is due to the sampling scheme. Its role in the analogy to fitness sharing is not immediately obvious, but it will be clarified in the following discussion.

6.2 Expected fitness of an antibody when partial matching is allowed

We now consider the general case in which an antibody receives a score for a partial match with an antigen at distance $d_{ij} = m$, where m ranges from $m = 0$ (perfect match) to $m = \ell$ (perfect mismatch). As before, in each cycle of the algorithm an antigen is picked at random, with replacement. Assume that each antigen j is selected with probability α_j. Let $\mathcal{S}_i(m)$ be the set of all antigens j at distance m from antibody i ($d_{ij} = m$). For antibody i to receive the match score in a cycle when antigen j at distance m is selected, we require

(i) No antibody at distance less than m from antigen j occurs in the sample. Recall that only the closest antibody in the sample receives the match score.

(ii) If w antibodies in the sample are all at distance m, antibody i must be in the sample and be chosen as the tie breaker. This latter event occurs with probability $1/w$.

Events (i) and (ii) are not independent. Thus to compute the probability that events (i) and (ii) are both true, we use the well known formula (Freund, 1962):

$$P(E_1 \cap E_2) = P(E_2|E_1)P(E_1)\ , \tag{7}$$

where E_i denotes event (i).

To compute $P(E_1)$, the probability of event (i), we again use the hypergeometric distribution. Recall that $N_j(k)$ is the number of antibodies at distance k from antigen j. Thus there are a total of

$$V_j(m) = \sum_{k=0}^{m-1} N_j(k) \tag{8}$$

antibodies at distance less than m from antigen j. The probability that none of these is in the sample of size σ is $p(0; \sigma, N, V_j(m))$. For later reference we define $V(0) = 0$, i.e., there are no antibodies closer than distance 0.

To compute $P(E_2|E_1)$, note that given that none of the $V_j(m)$ antibodies appears in the match sample, the probability that w of the $N_j(m)$ antibodies at distance m from antigen j appear in the

match sample is $p(w; \sigma, N - V_j(m), N_j(m))$. Not all antibodies at distance m from antigen j need be copies of antibody i. If w antibodies at distance m are in the sample, then the probability that antibody i is one of the w is $p(1; w, N_j(m), 1) = w/N_j(m)$. The probability antibody i is chosen as the tie breaker is $1/w$. Since any value of w between 1 and σ is possible, we find

$$P(E_2|E_1) = \sum_{w=1}^{\sigma} p(w; \sigma, N - V_j(m), N_j(m))/N_j(m) = (1 - p(0; \sigma, N - V_j(m), N_j(m))/N_j(m) \; . \tag{9}$$

Finally, combining the terms discussed above yields:

$$f_i = \sum_{m=0}^{\ell} \sum_{j \in \mathcal{S}_i(m)} \frac{s(d_{ij})\alpha_j}{N_j(m)} \left[p(0; \sigma, N, V_j(m))(1 - p(0; \sigma, N - V_j(m), N_j(m))) \right] \; , \tag{10}$$

where the first summation, $\sum_{m=0}^{\ell}$, considers all possible distances m from antibody i, and the second summation, $\sum_{j \in \mathcal{S}_i(m)}$, considers all antigens that are at distance m from antibody i. As in the previous simplified example, the terms $s(d_{ij})\alpha_j$ are related to the fitness available to a given antibody i, but in this case, the antibody can share in the finite fitness resources available at many distant antigens. However, like in fitness sharing, each of these resources is divided amongst the individuals that share it, as is indicated by the divisor $N_j(m)$.

Thus far the correspondence to fitness sharing is relatively straight-forward. However, the meaning of the bracketed hypergeometric terms,

$$\left[p(0; \sigma, N, V_j(m))(1 - p(0; \sigma, N - V_j(m), N_j(m))) \right] \; , \tag{11}$$

should be explained. The first term represents the probability that no antibody within distance $m - 1$ of antigen j will be selected in the sample. The second term represents the probability that, given the previous condition, at least one copy of an antibody at distance m from antigen j will be in the sample.

6.3 Relation to Explicit Fitness Sharing

To clarify the role of the hypergeometric terms in fitness sharing, we examine two special cases. Consider sample size $\sigma = 1$. In this case, the hypergeometric term (11) becomes

$$\left(\frac{N - V_j(m)}{N} \right) \left(1 - \frac{(N - V_j(m) - N_j(m)}{N - V_j(m)} \right) = \frac{N_j(m)}{N} \; .$$

Thus, for $\sigma = 1$

$$f_i = \sum_{m=0}^{\ell} \sum_{j \in \mathcal{S}_i(m)} \frac{s(d_{ij})\alpha_j}{N} \; .$$

In this special case, there is no fitness sharing, and the fitness values are independent. Essentially, the relative, expected fitness values are equivalent to those one would expect under a standard genetic algorithm. Under these conditions one would expect fitness-proportionate selection to converge to a single type of antibody. Note that this corresponds to fitness sharing with σ_s set to a value that spans the entire search space.

As a second special case, consider $\sigma = N$. If one assumes that a perfectly matching antibody exists for every available antigen in the population, the expected fitness reduces to

$$f_i = \frac{s_p \alpha_i}{N_i(0)} \; .$$

In this case, each antibody is only divided by its own effective proportion in the population. Like fitness sharing with $\sigma_s = 0$, one would expect fitness-proportionate selection to distribute the population based on relative fitness in one step, without a search for peak antibodies (see Section 4).

These special cases show that the limiting behavior of σ is similar to the limiting behavior of σ_s. To investigate the effects of other values of σ, consider the term

$$R = \left[p(0; \sigma, N, V_j(m))(1 - p(0; \sigma, N - V_j(m), 1)) \right] \; , \tag{12}$$

in which $N_j(m)$ is set to one. This simulates the situation where one antibody at distance m from the selected antigen is competing with $V_j(m)$ closer antibodies for fitness resources available from that antigen.

Figure 4 shows R plotted versus $V_j(m)$ for $N = 30$ and various values of σ.[4] Note that R is zero for $V_j(m) > (N - \sigma)$, and that R is near zero for a range of values of $V_j(m)$ lower than $N - \sigma$. Compare these curves to those in Figure 1 for $\alpha < 1$.

This graph and previous arguments imply that the hypergeometric terms correspond to a sharing function, and that σ plays a role in the immune system algorithm that is similar to that of σ_s in fitness sharing. Its value essentially implies a cutoff beyond which no sharing can occur. However, there is an important distinction to be drawn between fitness sharing and the implicit sharing in the immune system simulations. In fitness sharing, σ_s is a strict cutoff based on d_{ij}, which Deb (Deb, 1989b) recommends setting based on the volume of a hypersphere around a given peak. In the immune system algorithm, σ dictates a cutoff based on $V_j(m)$, which is the *proportion of the population within* a hypersphere of radius $m - 1$ around a given antigen. As the proportion of antibodies close to an antigen increases, the likelihood of more distant antibodies winning the

[4] Since only relative values of fitness terms are significant, curves in this figure are linearly scaled between zero and one.

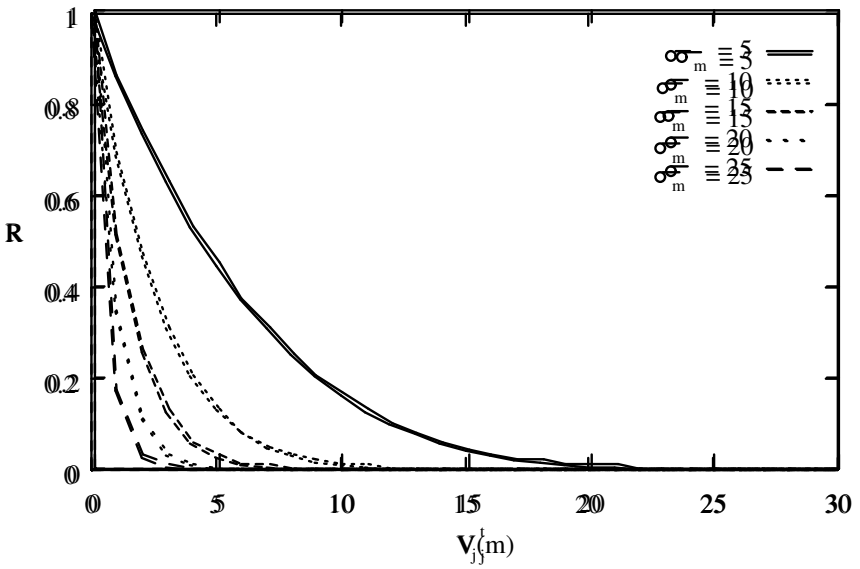

Figure 4: The hypergeometric term R, given by Eq. (12), versus V_j for various sample sizes σ. Note the similarity to $Sh(d_{ij})$ for $\alpha < 1$ and various values of σ_s (see Figure 1).

match competition decreases. Thus, effective antibodies block sharing by less effective antibodies. The sample size σ is a control on this effect. Under this scheme, the boundaries of sharing are a function of the proportion of antibodies clustered around given antigens. These emergent sharing boundaries explain much of the resilience shown by the GA in the immune system experiments.

7 Emergent Generalization

The previous section clarifies the correspondence between fitness sharing and the immune system experiments. In this section we use expected fitness calculations to explore various properties of the model, most importantly its capacity to generalize. We study the effects of the dependence of sharing boundaries on population proportions, the effects of various settings of σ, and finally, the question of generalization. Generalization is an important issue for resource-limited settings where either the population size is limited (and therefore one individual cannot be assigned to every task, niche, or fitness peak), or the resources under one peak are not sufficient to support one individual.

Consider three-bit antigens and antibodies. Note that these three bits can also be thought of as a portion of longer antibodies (schemata). Given $s(d_{ij})$ and α_j for the antigens and antibodies, the expected proportion growth equation, Eq. (1), and the expected fitness equation, Eq. (10), we can compute the expected effects of fitness-proportionate selection.

Assume that $s(d_{ij})$ is simply the number of complementary bits between antibody i and antigen j. As an illustration, consider a simulation in which the antigens `000`, `011`, and `110` are all equiprobable and none of the remaining five possible antigens occur ($\alpha_{000} = \alpha_{011} = \alpha_{110} = \frac{1}{3}$). Note that these antigens are equally distant from one another (each is at a distance of two from the others). Starting with equal proportions of each possible antibody, and a population size[5] of $N = 100$, Figure 5 shows the results of iterating the expected proportion equation, Eq. (1), with f_i given by Eq. (10), for $\sigma = 10$, $\sigma = 2$, and $\sigma = 3$. For $\sigma = 10$, the population converges to a condition in which the three antibodies that perfectly match the three antigens have equal proportions of $\frac{1}{3}$ each. Similar results occur for larger sample sizes. However, interesting results are obtained when the sample size is lowered to $\sigma = 2$. Surprisingly, the population converges to a single antibody that would perfectly match antigen `010`, an antigen which never occurs. The reason for this result is that all of the sampled antigens are within a hypersphere of radius one from this antibody. Thus, this antibody effectively *generalizes* from the given distribution of antigens. In a population that cannot maintain several peaks (due to the sharing limitations imposed by small σ) this antibody is the most effective. Given the previous results, it is interesting to ask whether the population can maintain both this generalist antibody and the three more specific antibodies at steady state. The answer is yes, given $\sigma = 3$.

The tendency for σ to act as a control on antibody generalization is further illustrated by a more complex example. Consider the situation in which each of the eight possible antigens occurs, but with different frequencies. Specifically, consider the following (randomly selected) probabilities of occurrence:

$$\begin{aligned}
\alpha_{000} &= 0.0510 \\
\alpha_{001} &= 0.0110 \\
\alpha_{010} &= 0.1581 \\
\alpha_{011} &= 0.1582 \\
\alpha_{100} &= 0.2177 \\
\alpha_{101} &= 0.0893 \\
\alpha_{110} &= 0.1210 \\
\alpha_{111} &= 0.1936
\end{aligned}$$

Other parameters are the same as in previous examples. Table 1 shows the steady-state results of

[5] In these computations, the binomial approximation to the hypergeometric distribution was used when factorials greater than 32! were needed.

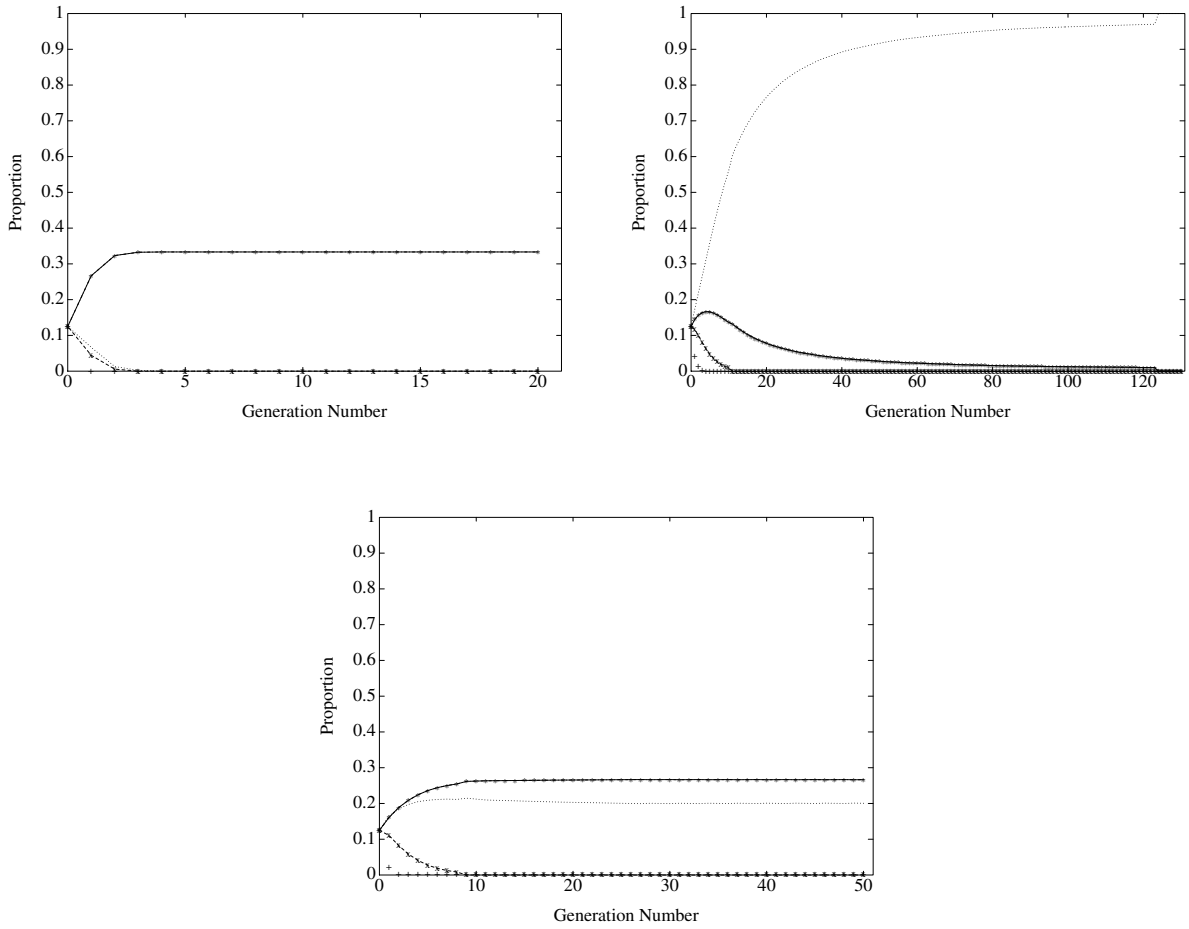

Figure 5: Changes in the antibody population predicted by iterating Eq. (1), with f_i given by Eq. (10). The population consists of the eight possible three-bit antibodies, which are matched against the equidistant antigens 000, 011, and 110. Initially all antibodies were present in equal proportion. The symbols used are — = P_{000}, - - = P_{001}, $\cdot\ \cdot\ \cdot = P_{010}$, $-\cdot- = P_{011}$, $\cdot\ \ \cdot = P_{100}$, $+ = P_{101}$, $* = P_{110}$, and $\times = P_{111}$, where P_x represents the proportion of antibodies that perfectly match antigen x. All eight proportions are plotted versus generation. In the upper left-hand plot, $\sigma = 10$, in the upper right-hand plot, $\sigma = 2$, and in the lower plot, $\sigma = 3$. In the $\sigma = 10$ case, three antibodies that perfectly match the three antigens `000`, `011`, and `110` all ultimately attain population proportions of 1/3. The graphs of three antibodies are all coincident, as are the graphs of many of the antibodies that go extinct. In the $\sigma = 2$ case, a single antibody, which perfectly matches antigen `010`, takes over the entire population. In the $\sigma = 3$ case, all four of these antibodies are maintained, those which perfectly match antigens `000`, `011`, and `110` at equal population proportions of ≈ 0.27 (coincident lines), and that which perfectly matches antigen `010` at population proportion of ≈ 0.2.

σ	# of Antibodies Retained (with $P > 0$)	Ordered List of Perfectly Matched Antigens
100	8	100, 111, 011, 010, 110, 101, 000, 001
50	7	100, 111, 011, 010, 110, 101, 000
10	6	100, 111, 010, 011, 110, 101
5	5	100, 111, 011, 010, 110
2	2	110, 111
1	1	110

Table 1: Steady-state conditions obtained by iterating Eq.(1), with f_i given by Eq. (10). The population consists of 100 three-bit antibodies. The eight possible antibodies are initially present at equal proportions. The antibodies are matched against a nonuniform distribution of eight antigens. The number of antibodies retained at non-zero steady-state proportions is indicated in Column 2. The antigens that are perfectly matched by these antibodies are listed (from highest to lowest proportion of matching antibodies) in column 3.

iteratively solving the proportion growth equations for various values of σ.

The $\sigma = 100$ case results are like those one would expect from total sharing. The system jumps to a population that is distributed across all possible antibodies in one step. The proportion of each antibody is directly related to the relative frequency of occurrence of its perfectly-matched antigen.

As σ is lowered, the population can not maintain as many antibodies at steady state. Results with $\sigma = 5$, $\sigma = 2$ and $\sigma = 1$ are shown in Figure 6. At $\sigma = 5$ the number of antibodies maintained drops to five. Antibodies that perfectly match the three least-frequent antigens (`000`, `001`, and `101`) are eliminated. The other five antibodies remain present, with the order of their proportions related to the frequency of occurrence of their respective perfectly-matched antigens.

An interesting shift in behavior occurs at $\sigma = 2$. For this parameter value, only two antibodies can be maintained. These two antibodies are those which match antigens `110` and `111` perfectly. Neither of these is a perfect match for antigen `100`, which has the highest probability of occurrence. Moreover, antigen `110` perfectly matches the antigen with the *lowest* occurrence probability of those retained with $\sigma = 5$. Yet, it has the highest steady-state proportion for $\sigma = 2$. Upon careful examination of occurrence probabilities and antigen structure, one can see the reason for these changes. The antigens within a hypersphere of radius one of `110` occur with probability 0.6904, and that antigens within a hypersphere of radius one of `111` occur with probability 0.5621. Moreover, antigens within the union of these two hyperspheres occur with probability 0.9375. Therefore, these two antibodies are excellent generalizations over the given distribution of antigens. In effect, the *fitness landscape* perceived by the GA has shifted substantially as a function of the sample size

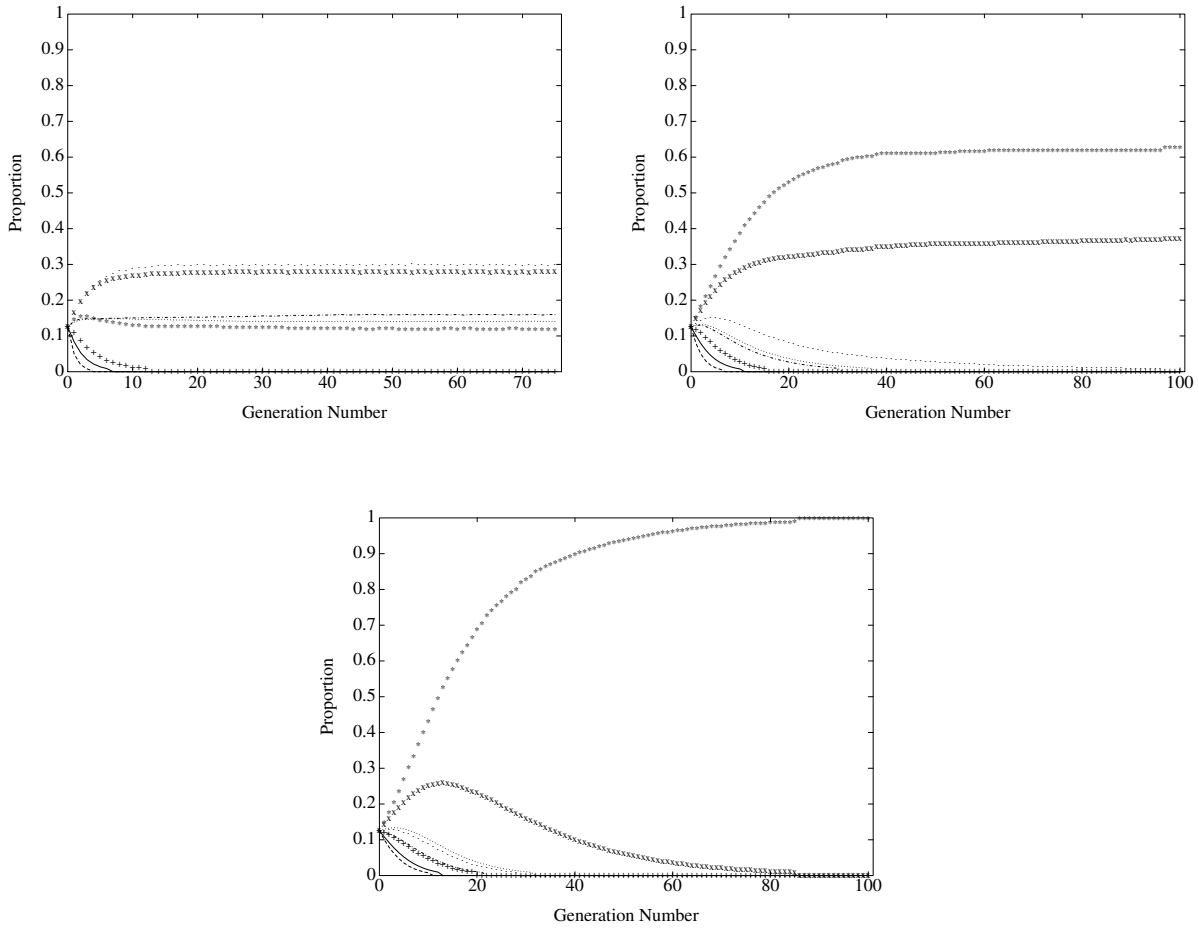

Figure 6: Iteration of the proportion equations, Eqs. (1) and (10), with $\sigma = 5$ (upper left), $\sigma = 2$ (upper right), and $\sigma = 1$ (lower), for eight three-bit antibodies matched against a nonuniform distribution of eight antigens. Symbols are as in the previous figure.

σ. When the population can only sustain a limited number of antibodies, selective pressures shift towards generalization.

For $\sigma = 1$ (where we know that the GA can only maintain a single peak) the antibody that perfectly matches antigen `110` takes over the entire population. Since antigens within a hypersphere of radius one of 110 occur with probability 0.6904 (the highest value for all possible anitbodies), this antibody is an effective generalization, given the constraint that only a single individual can be maintained.

8 Discussion

The study of biological systems can provide insights that are useful in the development of machine learning and pattern recognition algorithms. Conversely, studying biological systems with the perspective of trying to identify the algorithms that they use can provide new understanding of the operation of biological systems. Here we have shown how one can exploit insights that we have gained from studying the operation of the immune system for the development of a new algorithm for maintaining diversity in genetic algorithm search procedures. We have applied the result to a multipeak optimization problem of importance in immunology.

The immune system must be able to recognize an almost limitless population of foreign cells and molecules, i.e., antigens, with finite resources. We have formulated an abstract recognition problem of the type the immune system must solve in the form of recognizing a population of bitstrings that represent antigens. The immune system succeeds in its recognition task using both specialists (i.e. antibodies that recognize only one or very few antigens but recognize them with great accuracy) and generalists (i.e. antibodies that recognize many antigens but with limited specificity). Interestingly, the model that we have developed also discovers both specialists and generalists.

In this paper, we have developed a mathematical model to explain the behavior of the genetic algorithm simulations reported in (Forrest et al., in preparation), and used the model to make certain predictions about the ability of the simulation to develop general solutions. Based on these predictions, we then ran additional simulations and confirmed that the simulation does in fact develop generalizations when the sample size σ is smaller than the number of fitness peaks (antigens) (Forrest et al., in preparation) . This confirms the mathematical analysis presented here and demonstrates an important property of our immune model—the flexibility to develop either specific or very general solutions.

9 Conclusions

Maintaining diversity in a population is important in many computational models. For example, in classifier systems it is important to maintain a diverse set of rules that collectively perform complex tasks in varying situations. One mechanism for maintaining diversity in genetic algorithms is fitness sharing (Deb, 1989b; Deb & Goldberg, 1989b; Goldberg & Richardson, 1987). However, previous algorithms have been limited by the fact that the number of peaks or high fitness areas had to be known to correctly set σ_s, the peaks had to be (nearly) equidistant (Deb, 1989b), and N^2 comparisons are required. Here we have shown that an algorithm developed by Forrest et al. (in preparation) to study pattern recognition in the immune system has emergent properties that are similar to those of explicit fitness sharing. However, the fitness sharing in this algorithm is implicit, the number of peaks is determined dynamically by the algorithm and there is no specific limitation on the distance between peaks.

In order to help understand this algorithm we have calculated how the average fitness of an individual changes from generation to generation. Using these calculations we can predict what will happen when the environment requires the immune system model to generalize. In particular, our calculations show that the sample size σ plays the role of σ_s in fitness sharing, and controls the number of distinct individuals a population can maintain. Simulations show that σ is a control of generalization in the antibody population. As σ is reduced, selective pressures are altered such that a population of more general antibodies emerges.

The fitness sharing effects in the immune system algorithm are a result of the sample-and-match procedures employed which resemble the bidding mechanism commonly used in classifier systems. We believe that our analysis may provide a useful tool for understanding classifier system behavior. The fact that our model has the ability to generalize (even without using the # symbol explicitly) may prove relevant to the classifier system domain. An important area of further investigation is to extend our model to include more classifier-system like features. The immune system domain is an excellent testbed for classifier-system like models, since it provides a complex pattern recognition task.

Whether similar effects can be obtained as an alternative to explicit fitness sharing in multi-modal optimization problems is unclear. However, in other ecological problems the analysis performed here may have a straight forward application. For instance, in learning classifier systems, where ideas like "partial matching," "best match," and "generalization" have natural interpretations, techniques like those used here may prove useful.

In the context of classifier systems, some of the results presented here can be thought of as an

analytical confirmation of arguments presented by Wilson (**?**). Booker (1982) employs a similar technique. Wilson's experiments and approximate analyses suggest that division of reward between like-acting classifiers can lead to stable subpopulations of different classifier "concepts". Booker (1982) employs a similar technique. Although the mechanisms are different, the emergent effects in these systems are similar to those in the immune system simulations.

The techniques may also prove useful in other systems that form computational networks via genetic learning. A particular instance is in the development of polynomial networks for system modeling (Kargupta & Smith, 1991), where explicit fitness sharing was previously required.

Whether or not the techniques used in the immune system simulations can be explicitly transferred to more prescriptive applications, analysis of the type used in this paper can aid in understanding GA behavior in settings that require diversity. Improved understanding in this area is necessary if GAs are to be actively employed in systems that require cooperating sets of interacting individuals.

10 Acknowledgements

This work was partially done under the auspices of the U.S. Department of Energy. We thank the Center for Nonlinear Studies, Los Alamos National Laboratory and the Santa Fe Institute for support. Forrest also acknowledges the support of the Association of Western Universities, National Science Foundation (grant IRI-9157644), and Sandia University Research Program (grant AE-1679). Perelson acknowledges the support of the National Institutes of Health (grant AI28433). Brenda Javornik programmed the simulations referred to in this paper. Javornik and Ron Hightower both provided helpful comments on the manuscript.

References

Booker, L. B. (1982). *Intelligent behavior as an adaptation to the task environment*. PhD thesis, The University of Michigan, Ann Arbor, MI.

Booker, L. B. (1985). Improving the performance of genetic algorithms in classifier systems. In J. J. Grefenstette (Ed.), *Proceedings of an International Conference on Genetic Algorithms and their Applications*, 80–92, Pittsburg, PA. Lawrence Erlbaum.

Cohoon, J., Hegde, U., Martin, W., & Richards, D. (1987). Punctuated equilibria: A parallel genetic algorithm. In *Proceedings of the Second International Conference on Genetic Algorithms*, 148–154, San Mateo, CA. Morgan Kaufmann.

Collins, R. J., & Jefferson, D. R. (1991). Selection in massively parallel genetic algorithms. In R. K. Belew, & L. B. Booker (Eds.), *Proceedings of the Fourth International Conference on Genetic Algorithms*, 249–256, San Mateo, CA. Morgan Kaufmann.

Davidor, Y. (1991). A naturally occurring niche and species phenomenon: the model and first results. In R. K. Belew, & L. B. Booker (Eds.), *Proceedings of the Fourth International Conference on Genetic Algorithms*, 257–263, San Mateo, CA. Morgan Kaufmann.

Deb, K. (1989a). *Genetic algorithms in multimodal function optimization* (Tech. Rep.). Department of Engineering Mechanics, University of Alabama, Tuscaloosa, AL: The Clearinghouse for Genetic Algorithms. (Master's Thesis)

Deb, K. (1989b). *Genetic algorithms in multimodal function optimization* (TCGA Report No. 89002). Tuscaloosa: The University of Alabama, The Clearinghouse for Genetic Algorithms.

Deb, K., & Goldberg, D. E. (1989, Junea). An investigation of niche and species formation in genetic function optimization. In *Proceedings of the Third International Conference on Genetic Algorithms*, 42–50, San Mateo, CA. Morgan Kaufmann.

Deb, K., & Goldberg, D. E. (1989b). An investigation of niche and species formation in genetic function optimization. *Proceedings of the Third International Conference on Genetic Algorithms*, 42–50.

DeJong, K. A. (1975). *An analysis of the behavior of a class of genetic adaptive systems*. PhD thesis, The University of Michigan, Ann Arbor, MI.

DeJong, K. A. (1988). Learning with genetic algorithms: An overview. *Machine Learning*, *3*, 121–138.

Farmer, J. D., Packard, N. H., & Perelson, A. S. (1986). The immune system, adaptation, and machine learning. In D. Farmer, A. Lapedes, N. Packard, & B. Wendroff (Eds.), *Evolution, games and learning*. North–Holland. (Reprinted from *Physica*, 22D, 187–204)

Forrest, S., Javornik, B., Smith, R., & Perelson, A. S. (in preparation). Using genetic algorithms to explore pattern recognition in the immune system.

Forrest, S., & Perelson, A. (1991). Genetic algorithms and the immune system. In H. Schwefel, & R. Maenner (Eds.), *Parallel Problem Solving from Nature*, 320–325, Berlin. Springer-Verlag (Lecture Notes in Computer Science).

Freund, J. E. (1962). *Mathematical statistics*. Englewood Cliffs, NJ.: Prentice-Hall.

Goldberg, D., & Richardson, J. (1987). Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the second international conference on genetic algorithms* (148–154). Morgan Kaufmann.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Reading, MA: Addison Wesley.

Goldberg, D. E., & Deb, K. (1990). *A comparative analysis of selection schemes used in genetic algorithms* (TCGA Report No. 90007). Tuscaloosa: The University of Alabama, The Clearinghouse for Genetic Algorithms.

Goldberg, D. E., & Segrest, P. (1987). Finite Markov chain analysis of genetic algorithms. *Proceedings of the Second International Conference on Genetic Algorithms*, 1–8.

Gorges-Schleuter, M. (1989, June). ASPARAGOS: An asynchronous parallel genetic optimization strategy. In *Proceedings of the Third International Conference on Genetic Algorithms*, 422–427, San Mateo, CA. Morgan Kaufmann.

Grosso, P. B. (1985). *Computer simulations of genetic adaptation: Parallel subcomponent interaction in a multilocus model*. PhD thesis, The University of Michigan, Ann Arbor, MI.

Hillis, W. D. (1990). Co-evolving parasites improve simulated evolution as an optimization procedure. *Physica D*, *42*, 228–234.

Hines, W. W., & Montgomery, D. C. (1980). *Probability and statistics in engineering and management science* (2nd ed.). New York: Wiley.

Holland, J. (1975). *Adaptation in natural and artificial systems*. Ann Arbor, MI: The University of Michigan Press.

Holland, J., Holyoak, K., Nisbett, R., & Thagard, P. (1986). *Induction: Processes of inference, learning, and discovery*. Cambridge, MA: MIT Press.

Kargupta, H., & Smith, R. E. (1991). *System identification with evolving polynomial networks* (TCGA Report No. 91001). Tuscaloosa: University of Alabama, The Clearinghouse for Genetic Algorithms.

Levin, S. A. (1981). Mechanisms for the generation and maintence of diversity in ecological communities. In R. W. Hiorns, & D. Cooke (Eds.), *The mathematical theory of dynamics of biological populations ii* (pp. 173–194). Academic Press.

McInerney, J. (1992). *Biologically influenced algorithms and parallelism in non-linear optimization*. PhD thesis, Unviersity of California, San Diego, La Jolla, CA.

Muhlenbein, H. (1989). Parallel genetic algorithms, population genetics and combinatorial optimization. In *Proceedings of the Third International Conference on Genetic Algorithms*, 416–421, San Mateo, CA. Morgan Kaufmann.

Perelson, A. S. (1989). Immune network theory. *Immunol. Rev.*, *110*, 5–36.

Pettey, C., Leuze, M., & Grefenstette, J. (1987). A parallel genetic algorithm. In *Proceedings of the Second International Conference on Genetic Algorithms*, 155–161, San Mateo, CA. Morgan Kaufmann.

Smith, S. F. (1980). *A learning system based on genetic adaptive algorithms*. PhD thesis, University of Pittsburgh, Pittsburgh, PA.

Spiessens, P., & Manderick, B. (1991). A massively parallel genetic algorithm: implementation and first analysis. In R. K. Belew, & L. B. Booker (Eds.), *Proceedings of the Fourth International Conference on Genetic Algorithms*, 257–263, San Mateo, CA. Morgan Kaufmann.

Tanese, R. (1989a). Distributed genetic algorithms. In J. D. Schaffer (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms*, San Mateo, CA. Morgan Kaufmann.

Tanese, R. (1989b). *Distributed genetic algorithms for function optimization*. PhD thesis, The University of Michigan, Ann Arbor, MI.

Whitley, D., & Starkweather, T. (1990). Genitor II: A distributed genetic algorithm. *Journal of Experimental and Theoretical Artificial Intelligence*, *2*(3), 189–214.