# One View of Evolutionary Computation and Open Questions

## A Research Note

Robert E. Smith
Intelligent Computer Systems Centre
University of The West of England
Computing, Engineering and Mathematical Sciences Faculty
Bristol, United Kingdom
Email: robert.smith@uwe.ac.uk

## Introduction

In the following, I will attempt to characterize evolutionary computation [8][10], as a means of stimulating discussion at Math Encounters "Complex Systems in Computation" session (July, 2002). Most descriptions of EC would maintain an algorithmic focus; with a central algorithm manipulating data structures (see Figure 1).

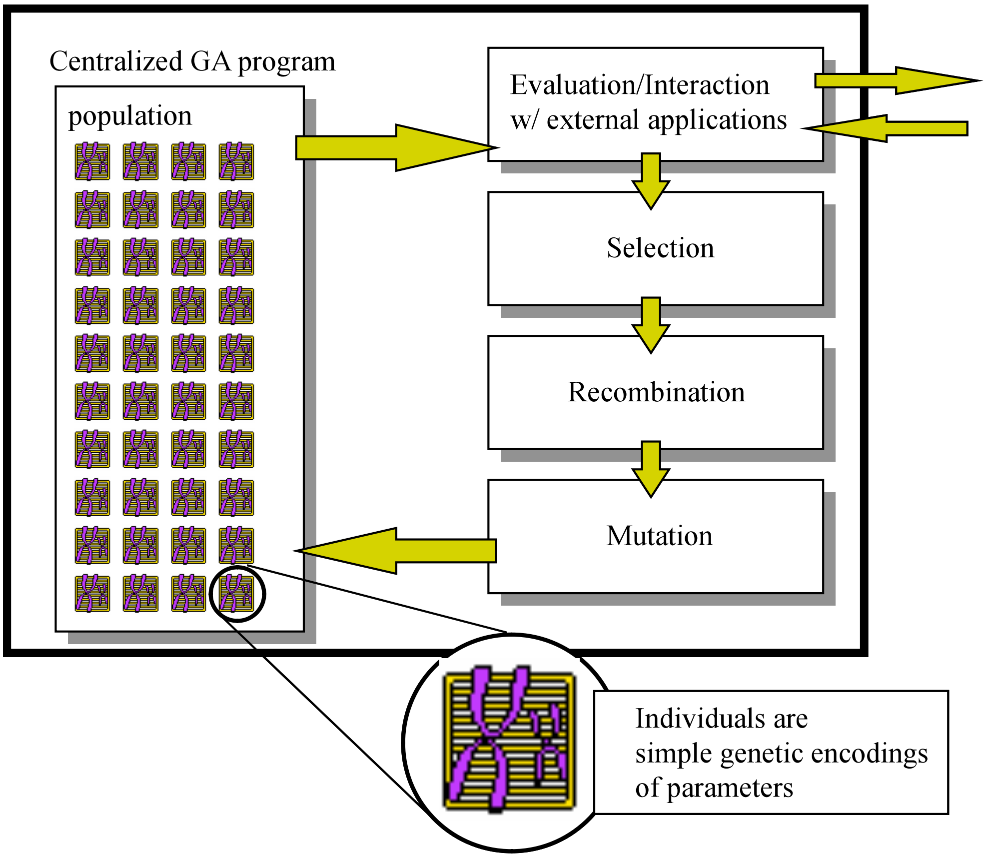

**Figure 1: A typical, centralized view of EC.**

This will be a somewhat non-standard characterization, in that I will focus on the interactions of individuals with one another (an "agent" perspective) [17] (see Figure 2).

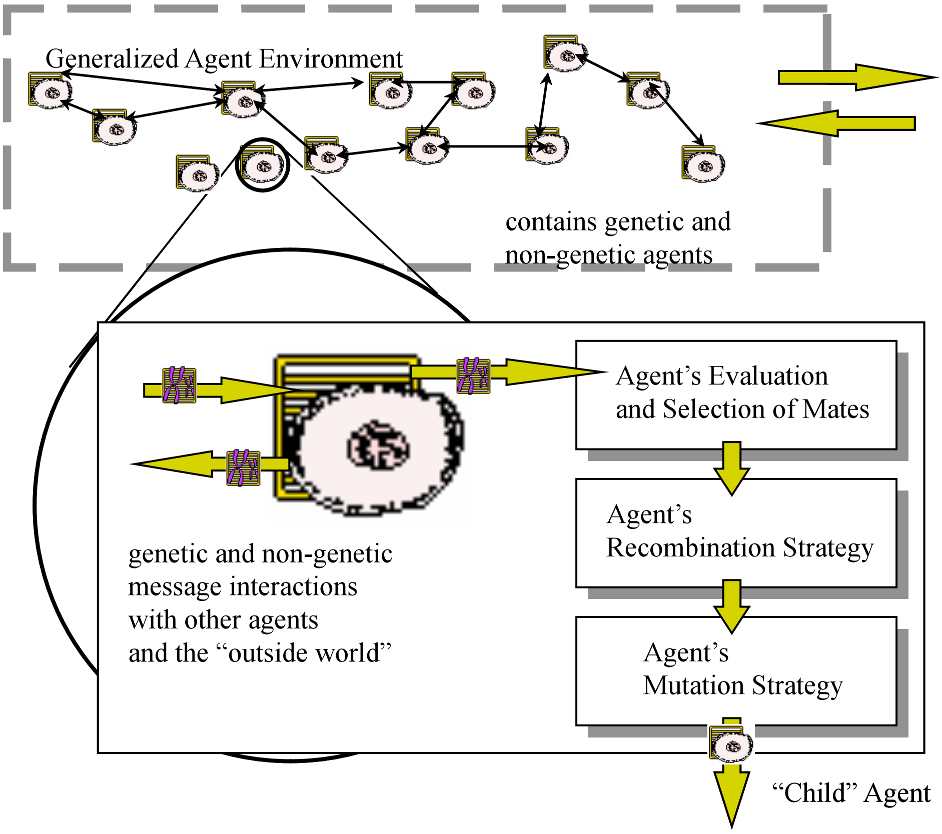

**Figure 2: An agent-based perspective on EC.**

The motivation for the different perspective offered here is generality. By adding a central, controlling agent, one can convert to the typical, algorithmic perspective, while the reverse is not always possible. Moreover, I hope this perspective no interacting agents will have a "physical" appeal.

## Basic Assumptions

Assume that there exists of set of agents, and an environment.

An agent *i* is characterized by a genotype $g_i \in \mathrm{G}$, and a phenotype $f_i^t \in F$ . The distinction between these items is that the former is not generally alterable in the agent's "lifetime", while the later can change as a part of a developmental process. The phenotype $f_i^t$ is, in general, a function of $g_i$, and the interactions between agent *i*, other agents, and the environment.

At any given time *t* an agent *i* can send a packet of information $x_i^t \in X$ to another agent. Some such packets can directly indicate the sending agent's genotype and phenotype.

At any given time *t*, an agent can generate another agent, in accordance with a set of global rules constraining this process, and a set of rules within the agent itself (which are, in general, influenced by $g_i$ and $f_i^t$). We assume that the agent has *reproduction function R.*

$$\mathrm{R}()gfgfgf \qquad \text{®}$$

At any given time, an agent can "die".

Agents receive information packets $s^t \in S$ from the environment.

Agents can send information packets $a_i^t \in A$ to the environment.

We can assume that there are neighbourhood structures over $G, F, X, S, A$ and .

## Simple Optimization

Assume that for each agent, there is a component of the agent's phenotype $f_i$ , which is a scalar, independent of time, and which we will call its fitness. To obtain this scalar phenotype value, each agent sends an action message to the environment. The environment returns a state message that is the value of $f_i$ (see Figure 3)

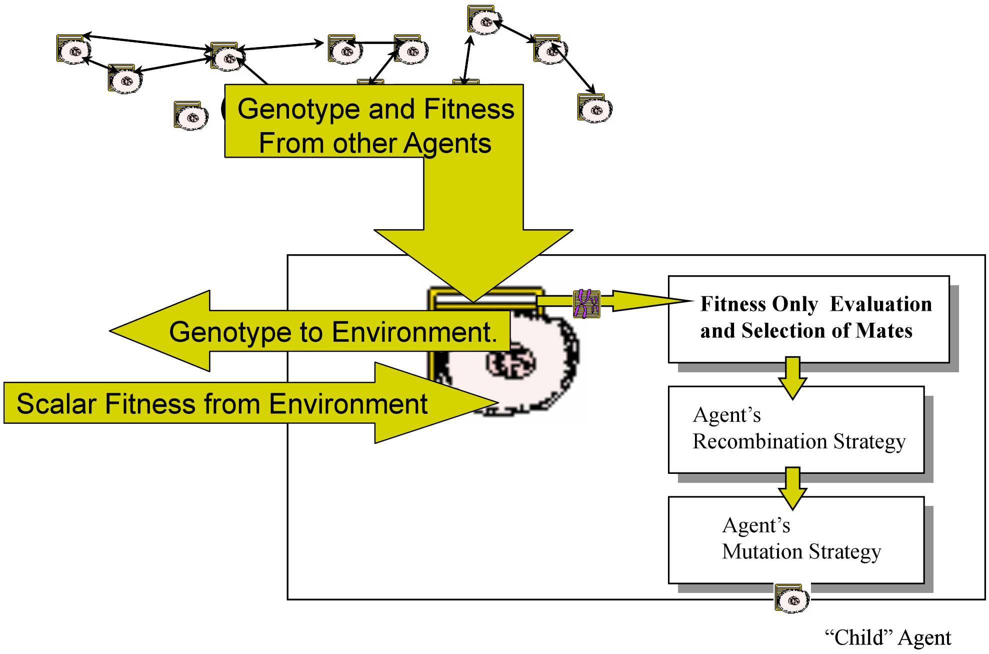

**Figure 3: An EC agent in a simple, optimization context.**

After an agent has determined its fitness in this fashion, it begins an exchange of information with other agents. Information packets exchanged between agents simply contain the sending agent's genotype $g_i$ , its fitness value $f_i$ and perhaps other information about its phenotype.

Agents store these packets as they receive them (technically as a part of their phenotype).

Every agent has the same reproduction function R. This function consists of

1. an operator that selects an information packet from the agent's store (biased on fitness)
2. an operator that combines the genotype of the selected mate with the genotype of the agent itself
3. an operator that randomly alters (mutates) the resulting genotype

Once this new genotype is constructed, the agent can produce another agent with this new genotype.

Note that the goal of the EC systems described in this simple optimization example is to construct agents that have high fitness, which is reflected in the fitness bias of the selection operator.

## Traditional GA Theory

This section will try to concisely restate some of the basic theories of EC, using an agent perspective, to indicate a direction along which EC theories can have implications for general agents. Many of the developments follow those in [8]. Although these theories have been much debated in the EC literature [11], I believe their conceptual conclusions provide a valuable perspective on self-organizing evolvable systems of general agents.

Given this outline of an agent's localized perspective, let us consider the resulting global effects. Using notation similar to that in [8], let the expected proportion of existing agents containing some subset of features *H* of its genotype at some time *t* be $P_H^t$. The expected number at time *t+1* is given by:

$$P_H^{t+1} = R_H [1 - D_H]$$

where $R_H$ is the probability of any individual agent selecting feature *H* (as a part of the selection operator in its reproduction function *R*), and $D_H$ is the probability of feature being disrupted by other aspects of *R* (i.e., recombination and mutation). Note that each of these probabilities may be a function of the proportions of agents with the given feature *H*, and other features in the neighbourhood of *H*.

This simple expression makes no assumptions that can be said to be "genetic." Our first assumption that vaguely relates to biological analogy is to cast the formula above in terms of a *reproductive plan* [9]. That is, we will assume that the probability of selecting a feature is proportional (or just directly related) to the proportion of agents that contain that feature. In other words, the more agents that persist in having a feature, the greater the likelihood that agents will adopt or retain that feature. This is clearly the case in most imaginable variations of the simple optimization example offered above, in that any give agent is more likely to receive a message containing any given feature *H* if that feature is more present in the population. In its simplest (proportional) form, this gives:

$$R_H = P_H^t S_H^t$$

where $S_H$ is a reproductive rate related to the feature's perceived utility (average fitness) across the population of agents. One common form for $S_H$ is

$$S_H = \frac{f_H}{\bar{f}^t}$$

where $f_H$ is the average fitness of agents whose genotype contains feather *H*, and $\bar{f}^t$ is the average fitness of all agents in the population at time *t*. If we say that $P_i^t$ is the proportion of genotype *i* at time *t*:

$$\bar{f}^t = \sum_i P_i^t f_i$$

and

$$S_i = \frac{f_i}{\bar{f}^t}$$

Note that this simple "proportional selection" form is often used in EC, but any increasing function of proportion would yield *conceptual* conclusions similar to those presented here.

Substituting yields:

$$P_H^{t+1} = P_H^t S_H [1 - D_H]$$

which is a proportional form of Holland's schema theorem [9]. This formula does not depend explicitly on the form of most of the internal workings of the agents (i.e., the method of encoding features, or the operators within the agents). It only depends on the assumption of a reproductive plan.

Why a reproductive plan? This "bandwagoning" onto apparently useful features in other agents is certainly not the only rational approach from an agent perspective. Agents may find it useful to run counter to what other agents do. However, a reproductive plan is certainly one reasonable strategy, and worthy of examination. Moreover, other plans that are not explicitly reproductive in character, but which use perceived utility of features in other agents to bias feature selection, may yield similar mathematical forms.

Let's assume that the agent's reasoning about desirable features is generally correct for some desirable feature $H$, and that $S_H[1-D_H]$ remains greater than one for that feature. Ignoring constraints on proportions, this dictates an exponential increase in the $P_H^t$ with respect to $t$. Is this form of increase desirable?

Holland's *k-armed bandit* argument [9] shows that, regardless of the distributions of utilities of $H$ and competing (mutually exclusive) features, a near-optimal rate of increase should be of exponential form with respect to time. A reproductive plan, like that stated above, yields this exponential form for certain features. This is an emergent effect at the system level, which only involves interactions at the agent level. The features that show this near-optimal, exponential effect are those with low rates of disruption, $D_H$, relative to the magnitude of $S_H$. In EC, such features are often referred to as *building blocks*.

At this point, note that the previous discussion is not inherently genetic, and could as easily apply to *memes* as genes [3][4]. Memes are replicators and are defined as units of "cultural" information, such as cultural practices or ideas, which are transmitted via communication and imitation between one entity and another. Clearly, memes are subjected to a reproductive plan, in the sense of [9]. In the abstraction offered here, memes are aspects of $F$. The primary difference between genes and memes is that we have an understanding of the underlying *encoding* of genes (in $\Gamma$), which do not change during an agent's lifetime, but we have no such understanding (in general) of memes (which are aspects of $\Phi$). Otherwise, the two entities behave in a conceptually identical fashion, that is, selfishly trying to maximise their reproductive success.

The remaining technical discussion in this section concentrates on the assumption of some atomic encoding unit of an agent's features in $\Gamma$. Although we may not understand what this unit is for agent memes, much of the reasoning will still ultimately hold. Moreover, the EC offshoot field of *memetic algorithms* [12] can provide some insight in light of the perspective presented here. Since all interactions between agents could be categorized as potential transmissions of genes or memes, we believe that the EC-based perspective here can provide insight into general interactions between agents that may not be specifically genetic in character.

All building blocks are treated in the emergent, yet near-optimal fashion indicated above, under a reproductive plan. Therefore, we should consider how many of these building blocks exist in a population of individuals. However, to maintain a general agent focus, we will do this without specific reference to EC details (e.g., genetic encoding). We only assume that there is a set of (discrete) atomic features, from which all other features are constructed. These atomic features are (roughly) analogous to genes in biological systems, but we are not assuming any particular, underlying encoding.

An agent that contains $M$ atomic features contains $2^M$ features, since all possible subsets of these $M$ features can be considered to be features themselves. Whether a feature can be said to have low-disruption is a function of the internal operation of the agents themselves, and the resulting disruption rates. However, let us assume that we can consider (without loss of generality) some subset (or subsets) of features of size $cM$ $(0 < c < 1)$ or less to be building blocks. The number of building blocks containing only $m$ atomic features in any such subset varies as follows:

$$N_m = \binom{cM}{m} = \frac{cM!}{m!(cM-m)!},\quad 0 \le m \le cM$$

This distribution is symmetrical about the most numerous size of building blocks, $m=cM/2$. Let us assume all possible atomic features occur in agents equiprobably, with probability $p$. The probability of a building block with any given set of $cM/2$ atomic features occurring in a given agent is $p^{cM/2}$. In a population of size $N$, the expected number of copies of a building block of this (most numerous) size is $Np^{cM/2}$. We can insure that the expected number of copies of building blocks of this size has a value of 1, by selecting a population size as follows:

$$Np^{cM/2} = 1, \quad N = \left(\frac{1}{p}\right)^{cM/2}$$

Given this simplifying, population-sizing assumption, all building blocks with $cM/2$ or more features have an expected number of copies that is 1 or less. There are $2^{cM-1}$ such building blocks in any of the subsets we are considering, in an individual agent. Therefore, we can say that the number of building blocks ($N_{bb}$) in the population has the following lower bound:

$$N_{bb} \ge \left(\frac{1}{p}\right)^{cM/2}\left(2^{cM-1}\right)$$

Rearranging yields:

$$N_{bb} \ge \left(\frac{1}{2}\right)\left(\frac{1}{p}\right)^{cM/2}\left(\frac{1}{p}\right)^{\frac{cM}{\log_2[(1/p)]}}$$

or

$$N_{bb} \geq \left(\frac{1}{2}\right)^{1} \left(\frac{1}{p}\right)^{[2][\log 2 (1/p)] + cM}$$

Rearranging, and recalling that population size is $(1/p)^{cM/2}$, yields a lower bound on the number of building blocks, in terms of population size $N$:

$$N_{bb} N \geq \left(\frac{1}{2}\right)^{[2][\log 2 (1/p)] +}$$

For binary atomic features, this gives a form of the $N^3$ lower bound often associated with genetic algorithms [8][9]. However, regardless of the assumed form of atomic features, the general estimate shows that (under certain restrictive assumptions) a large number of building blocks are implicitly treated in the near optimal fashion indicated by the k-armed bandit argument, as an emergent phenomenon of reproductive plans. This key emergent effect of reproductive plans is referred to in EC as *implicit parallelism*.

### Genetic Drift

Consider the case where all (possible) agent genotypes result in the same fitness.

In such a case (assuming an absence of disruption), one would imagine that any proportions of agent genotypes would be a steady state for the population.

$$P_i^{t+1} = S P_i^{t} = P_i^{t} = \frac{f_i}{\bar{f}}$$

However, due largely to finite-population sizes and random effects in the reproduction process, this is not the case. Inevitably, one agent will gain a proportional advantage in the population. Once this occurs, the population will show *genetic drift* towards a population that is comprised of agents with the same genotype, through an accumulation of such random effects. In practice, the only steady state is that of a population dominated by a single genotype (modulo mutations).

## Population Diversity (Species and Niching)

Let's assume that agents not only use fitness *f* message from the environment as a means to bias there selection of mates, but that they also consider the distribution of potential mates (relative to each other, and to the given, selecting agent itself) in the genotypic or phenotypic space (under the given neighbourhood structures of those spaces) (see Figure 4).

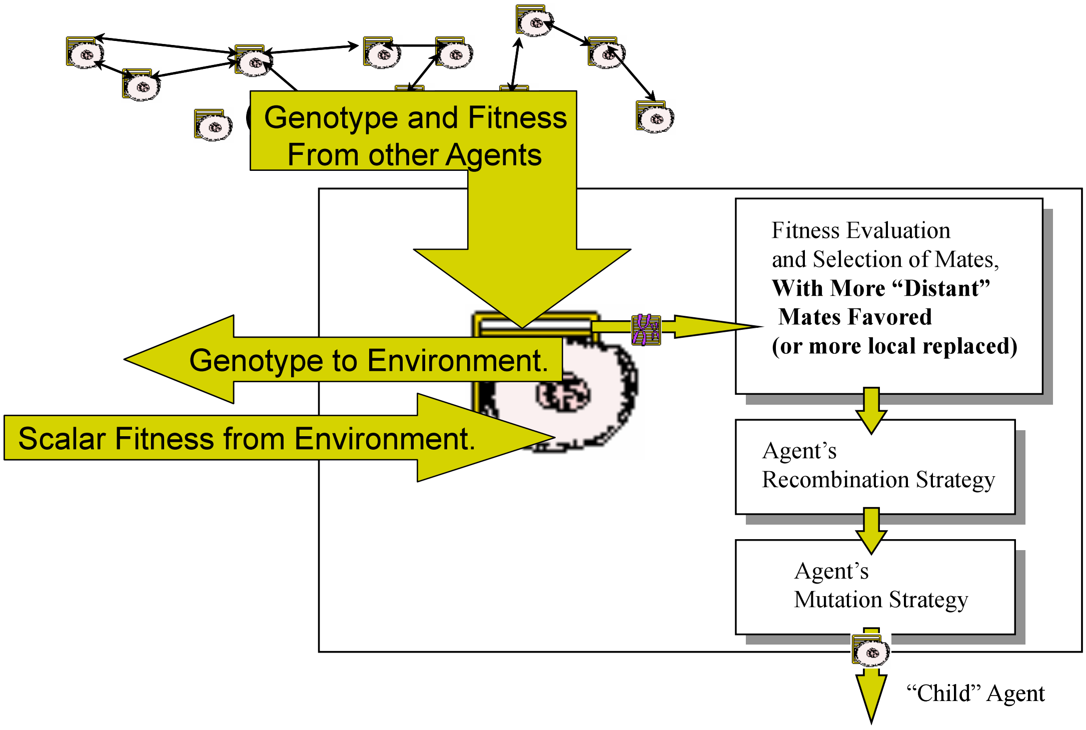

**Figure 4: Niching is induced when genotypic (or phenotypic) neighbourhood plays a role in mate selection.**

Let's consider a neighbourhood structure over genotypes, defined by a distance metric $d_{ij}$, which is the distance between genotype *i* and genotype *j*. Note that we could just as well employ a distance metric over a neighbourhood structure in phenotype space.

Assume that this bias can be expressed as the creation of an *effective fitness* from the *raw fitness* that is given by the environment. One such form of bias would be:

$$f_i = \frac{f_i^*}{\sum_j Sh(d_{ij})}$$

where $f_i^*$ is the raw fitness, $Sh(d_{ij})$ is a decreasing function of $d_{ij}$, and $f_i$ is effective fitness that is used to bias selection. Typically $Sh(0) = 1$, and $Sh(x) = 0$ for all *x* greater than some value $O$. This value is called the *niche size*. Experience shows that the form of $Sh(d_{ij})$ is less important than the value of $O$.

Consider the effects on $R_i$ in the drift case outlined above:

$$P_i^{t+1} = R_i P_i^t = \frac{f_i^*}{\bar{f}\sum_j Sh(d_{ij})} \qquad \frac{\quad}{\sum_{kj}\sum P_{kj}^t Sh\,d \, \frac{f_k^*}{\sum_m Sh(d_{km})} \qquad (\ \ )} \qquad \frac{Z}{Z}$$

where

$$A_i = \frac{1}{\sum_j Sh(d_{ij})}$$

That is, if the fitness values of all agents are equal, selection acts using this the inverse of the sharing function sum in the role of fitness for each individual. At steady state,

$$\frac{Z_i}{Z} = 1$$

for all *i*, which implies $Z_i = Z_j$ for all *i* and *j*. Given the interdependences of $Sh(d_{ij})$ values, and the fact that $Sh(d_{ij})$ increases with the proportion of individuals in any one niche, this effect tends to distribute the population on isolated peaks, that remain stable, thus overcoming drift.

The *fitness sharing* scheme discussed above is a centralized calculation, performed to induce stable (drift resistant) diversity in clusters around fitness peaks in genetic algorithms [4]. Although fitness sharing is centralized, similar effects (implicit fitness sharing) on $R_i$ can be the result of interactions between agents, and between agents and the environment, given that (in general, outside of the simple optimization case above) $R_i$ is a function of these interactions, via the agent's phenotype. A more agent-based perspective is discussed later in this document.

*Mating restrictions* are another means by which "species" and "niching" are induced or maintained in EC algorithms. A mating restriction simply eliminates some potential mates from an agent's consideration, based on examination of phenotype or genotype (typically for similarity to the agent doing the selection). Usually mating restrictions do not allow agents that are less similar than some threshold $O$ to mate (i.e., *i* and *j* cannot mate if $d_{ij} > O$).

### Multiple Objectives

In the simple optimization example above, it was assumed that there is a literal scalar "fitness" upon which selective decisions are based. In a slightly more general case, one can imagine a vector of objective function measures (implicit or explicit) upon which selective decisions can be based. In some EC algorithms this approach, coupled with speciation and niching, is used to spread solutions on the Pareto optimal front in the (vector) objective function space [5].

## Agent Perspective Proportion Equations

Implicitly, the proportion equations discussed in previous sections are constructed from a centralized perspective. Note that this assumes a global perspective on fitness, selection, and other operators.

To consider systems where each individual in the population makes its own selective decisions, one has to modify the proportion equations. First, one must consider that the fitness of each schema may differ, depending on the perspective of the individual. Let us define $f_{ij}$ as the (average) fitness of schema $i$ from the perspective of an individual containing schema $j$. Note that if all agents had the same perspective as a centralized EA, the matrix of perceived fitness values $[F]$ would be a matrix with identical elements (the effectively centralized view of fitness values) in each column (that is $f_{ji} = f_{ki} = f_i$ for all *j* and *k*).

We will similarly define:

$$R_{ij}^t = \frac{f_{ij}}{\bar{f}_j^t}$$

where $\bar{f}_j^t$ is the average fitness of the population from the perspective of individual $j$:

$$\bar{f}_j^t = \sum_i P_i^t f_{ij}$$

In a system where agents perform selection of mates with whom they reproduce children agents, one cannot in general assume that the population size will remain constant. This complicates (but does not preclude) the construction of proportion equations. For the purposes of this note, let us assume the "child replaces mother" model used in [17]. In this model, all agents can act as both fathers and mothers. Agents acting in a mother role select amongst possible agents acting in a father role, and mate with them. Any agent can act many times in the father role, but once a child is produced in the mother role, the mother "dies", and is replaced by the child.

In this case, let's draw the proportion equations, considering selection to occur in the mother role:

$$P_i^{t+1} = (1-\varepsilon)P_i^t + \varepsilon \sum_{jk} c_{ijk} P_j^t R_{kj}^t P_k^t$$

where $\varepsilon$ is the proportion of individuals that act in the mother role at any given time $t$, and $c_{ijk}$ is the probability of creating and individual that possesses schema $i$ from a mating when an individual that has schema $j$ mothers a child with an individual that has schema $k$.

To consider a selection-only model, let us assume that a mother individual selects a mate (based on perceived fitness), then replaces itself with a child that is identical to itself with probability $(1-\alpha)$, or with a child that is identical to the selected father with probability $\alpha$.

In this case, the only non-zero terms in the previous equations are those where either $j = i$, or where $k = i$. This gives:

$$P_i^{t+1} = (1-\varepsilon)P_i^t + \varepsilon\left((1-\alpha)P_i^t + \alpha \sum_j P_j^t R_{ij}^t P_i^t\right)$$

Or:

$$P_i^{t+1} = P_i^t - \varepsilon\alpha P_i^t + \varepsilon\alpha \sum_j P_j^t R_{ij}^t P_i^t$$

Clearly, a steady state for individual $i$ is reached whenever:

$$\sum_j P_j^t R_{ij}^t P_i^t = P_i^t$$

for all *i.* Thus:

$$\sum_j P_i P_j \frac{f_{ij}}{\bar{f}_j} = P_i$$

Or:

$$\sum_j P_j \frac{f_{ij}}{\bar{f}_j} = 1$$

$$\sum_j P_j \frac{f_{ij}}{\sum_k P_k f_{kj}} = 1$$

for all *i* where $P_i \neq 0$.

There are a number of possible conditions where this equation may hold true. Two straightforward conditions present themselves. To examine these, let's consider only the "effective" $\left[F\right]$ matrix, where rows and columns associated with individuals who have zero proportions have been eliminated.

In the first steady-state condition, if the effective $\left[F\right]$ is diagonal ($f_{ij} = 0$ for all $i \neq j$), a steady-state clearly exists. In the second, if $\left[F\right]$ is a matrix of identical columns ($f_{ij} = f_{kj}$ for all *i* and *k*):

$$\sum_j P_j \frac{f_{ij}}{\sum_k P_k f_{kj}} = \frac{\sum_j P_j}{\sum_k P_k} = 1$$

As an additional observation, note that in the case of a centralized EC algorithm (where every agent has the same perspective on fitness values) $\left[F\right]$ is a matrix of identical rows. Thus:

$$\sum_j P_j \frac{f_i}{\sum_k P_k f_k} = 1$$

Or:

$$\frac{f_i}{\bar{f}} = 1$$

for all *i* where $P_i \neq 0$. This is clearly only the case if all individuals with non-zero proportions have the same fitness, as in the drift case.

## *Agent-Based Fitness Sharing*

Let's assume that fitness sharing operates in the manner described earlier For the moment, let's assume each individual evaluates identical shared fitness values (as in the centralized scheme). Let's further assume that convergence has occurred, such that there is only one type of individual with non-zero proportions in each niche (individuals outside the niche are further than $O$ away from one another). Thus, the sharing function that evaluates effective fitness gives:

$$f_i = \frac{f_i^*}{P_i}$$

where the * designates the raw fitness value. Given that this is effectively a centralized algorithm, the effective $[F]$ matrix must reach a state where all individuals with non-zero proportions to have the same fitness, as in the drift case. Therefore, for a steady state:

$$f_i = f_j = \Phi$$

for all *i* and *j* with non-zero proportions, and $\Phi$ is a value that can be easily calculated. Summing the proportions:

$$\sum_i P_i = \sum_i \frac{f_i^*}{\Phi} = 1$$

Therefore:

$$\Phi = \sum_i f_i^*$$

and:

$$P_i = \frac{f_i^*}{\sum_j f_j^*}$$

This is the drift-resistant proportion vector one would expect for fitness sharing in the centralized EA with one individual in each niche.

If we consider the case where each agent potentially evaluates shared fitness values differently. If we assume that a steady state exists where every individual with non-zero proportions perceives only one individual in each niche:

$$f_{ij} = \frac{f_{ij}^*}{P_i}$$

The most obvious steady state will be when the effective $[F]$ is a matrix of identical columns (where each individual sees its sharing function as having stabilized to a single effective fitness for each individual). Thus:

$$f_{jk} = f_{jj} = \Phi$$

Summing proportions:

$$\sum_i P_i = \sum_i \frac{f_{ij}^*}{\Phi_j} = 1$$

Therefore:

$$\Phi_j = \sum_i f_{ij}^*$$

and:

$$P_i = \frac{f^*_{ij}}{\sum_k f^*_{kj}}$$

If a condition exists where there is only one individual in each niches, and the equation above yields the same values for all $i$ and $j$ with non-zero proportions, then agents with separate, different sharing functions can reach a common set of steady-state proportions. This is a sufficient condition for such a steady state, but it is not implied that it is necessary. Other steady states may exist.

As a final note, consider the case where every agent has the same view of the *raw*, pre-sharing (*) fitness of every agent, but different sharing functions. Clearly, in this case previous arguments hold, and;

$$P_i = \frac{f^*_{i}}{\sum_k f^*_{k}}$$

which is the same steady state as in centralized fitness sharing. This indicates that agents that perceive the same non-shared fitness can use different sharing functions, and if a condition exists where all agents perceive only one individual in each niche, this condition is a steady state. Once again, this is a sufficient, but not necessary, condition.

### *Agent-based Mating Restriction*

Consider the case of agent-based mating restriction, where an agent rejects any mates that are over some distance $O$ away. Effectively, in this case:

$$f_{ij} = \begin{cases} f^*_{ij} & \text{if } d_{ij} < O \\ 0 & \text{otherwise} \end{cases}$$

where $d_{ij}$ is a distance metric between $i$ and $j$ in genotypic or phenotypic space.

Note that this implies certain definite steady states of the mating restriction scheme (those where there is a single individual in each niches, and, therefore, individuals are too far from one another to perceive them as fit), as well as certain aspects of the dynamics of selection only in the scheme, via the previous proportion equation. In this case, the effective $\left[F\right]$ matrix is diagonal, which yields an obvious steady state.

## Learning Classifier (like) Systems

Moving beyond the simple optimization example above, let's assume more general interactions between agents and the environment. In particular, let's assume the environment sends agents a message indicating some aspect of the environment's state. Let's assume that an agent compares this message to some aspect of its phenotype. If certain conditions are met (i.e., the agent is "matched" to the environment state), the agent posts a message to the environment (based, of course, on certain aspects of its phenotype) (see Figure 5).

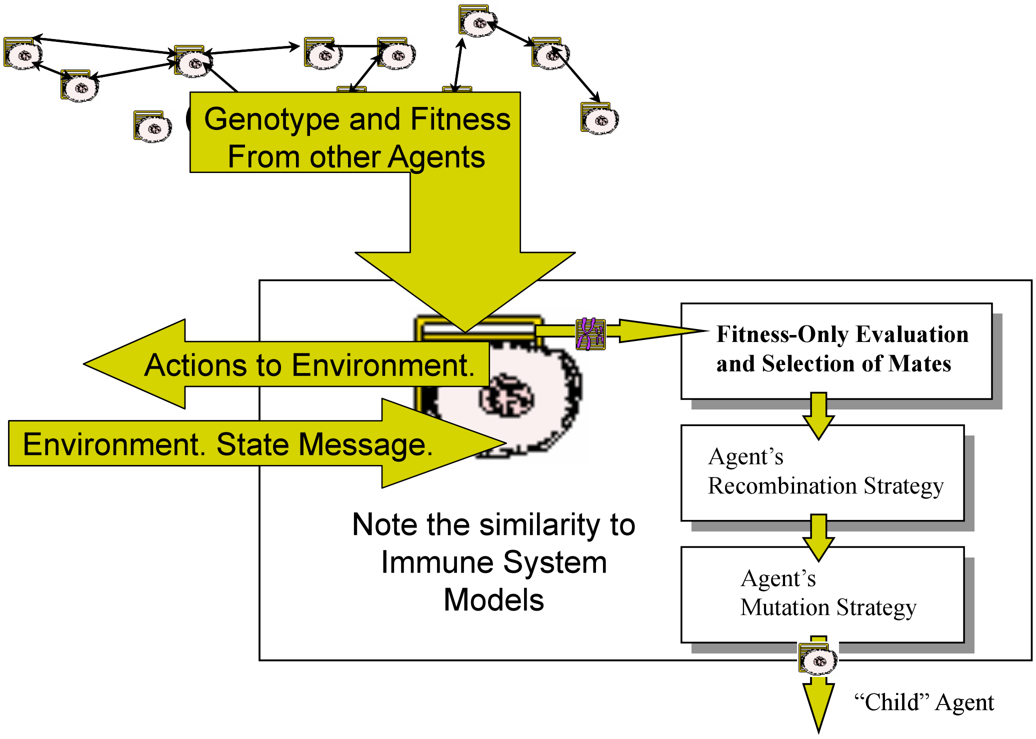

**Figure 5: A context with more general agent-environment interactions, reminiscent of *learning classifier systems*, or immune system models.**

Let's further assume that each agent can receive (via another message or as a part of the state message) something that it can interpret as punishment, reward, or cost. Let's assume that this is a scalar, with negative values representing punishments (or costs), and positive values rewards.

If we view the environment as a reinforcement learning problem [18] (see Figure 6), the system of agents can be seen as a reinforcement learning controller. However, one must assume that there is some conflict resolution in the environment to interpret agent's "action" messages, and disambiguate any of these messages that indicate conflicting actions in the environment. Note that such disambiguation can be based on one or more aspects of the agents' phenotypes.

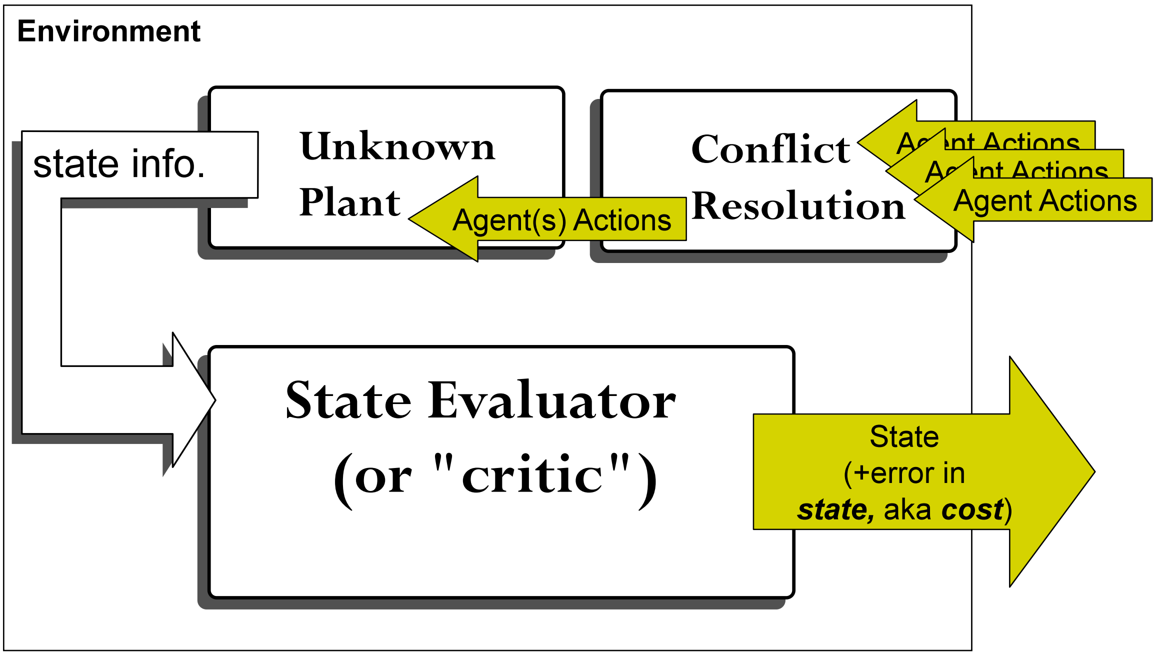

**Figure 6: Environment viewed as a reinforcement learning control problem.**

To aid in assigning rewards to agents, we can also assume from credit assignment mechanism in the environment.

Clearly, the system outlined is similar to both immune system models [7][15], and to Holland's *learning classifier system*. In Holland's original version of the LCS, a scalar fitness for each agent was used for the EC process of creating new agents, in a way similar to the simple optimization example outlined above. However, this fitness was not (in general) simply assigned by the environment. Instead, the agent implemented an exponentially smoothed online average of rewards, the result of which was used for fitness. This value (called *strength*) was also used as a basis for conflict resolution. The calculation of this fitness value is discussed further in a later section.

By adding additional reinforcement to the agent acting in the past time steps that reflects the agents that act in the current time step, this scheme can be seen as similar to any given temporal reinforcement learning scheme (e.g., Q-learning).

In a more recent (very successful, and very popular) form of LCS called XCS, an estimate of the variance of reward (rather than its average) is used to compute an *accuracy* (which has an inverse relationship to variance), which is in turn used as fitness. In XCS an estimator of average reward is still employed for conflict resolution [15].

## *Hierarchical Neighbourhood Structure*

Like the simple optimization example, speciation and niching can be employed in LCSs. In fact, since the "solution" to a problem (an effective controller formed by a group of agents) is likely to require a diverse set of agents, such effects are usually essential.

Under certain reward strategies (particularly those who divide rewards between agents that match the same environmental message and produce the same actions) it has been shown that implicit fitness sharing occurs as an emergent effect [11].

In XCS, a different strategy is used. In this system, only agents that match the same environmental message are allowed to participate in an EC mating process together. In some XCSs, this restriction is further narrowed to only include matching agents that also suggest the same action [21]. Clearly, in either case, this is a form of mating restriction.

However, in both cases, one must consider the neighbourhood structure over which sharing or mating restriction (implicitly) act. This is a structure induced by the nature of matching and acting in the environment (along with the distribution of state messages from the environment). Moreover, it is a structure that implicitly involves *generalization*.

Under such an induced structure, an agent *i* can be said to a generalization of another agent *j* if it matches all the messages that *j* matches, and some additional messages that are not matched by *j*. Under these conditions, *j* is also said to be a *specialization* of *i*. Depending on the associated LCS operations, one might have to add the additional restriction that *i* and j have the same action.

Note that *i* may also be a generalization of another agent *k*. In turn, *k* may or may not be a generalization of *j* (and vice versa).

## Relationship to RL

It is useful to briefly consider the relationship of LCS generalization to reinforcement learning. Most reinforcement learning schemes are based on a table of values upon which action decisions are based. For instance, Q-learning [18] keeps a table of Q values, one for each possible state/action pair in the environment, and whenever a state presents itself, the action with the highest Q value is preferred. Q values are updated by the following equation:

$$Q_{t+1}(s_t,a_t) = (1-\in)Q_t(s_t,a_t) + \in\left(c_{s_t}(a_t) + B\min_{a \in A(s_{t+1})} Q_t(s_{t+1},a)\right)$$

where $\in$ and $B$ are parameters, and $c_{s_t}(a)$ is the cost of taking action *a* at time *t* in state *s*. There are a number of other such tabular reinforcement learning schemes, with various types and proofs of convergence.

However, one clearly cannot scale these approaches for large numbers of states and actions. In such cases, it is necessary to generalize over the space of states (and possibly the space of actions). One can imagine an agent handling (matching) a particular subset of states, and associating with those states one or more actions, through associated, aggregate “Q” values. Since such generalization is imperative to RL scale up, and since RL problems are ubiquitous, the motivations for the discussion included here are clear.

Note that in the typical, Holland LCS scheme, the *strength* value is calculated in a fashion similar to the Q-value. In other systems, the values inversely related to estimates of variance of Q are employed. In each case, there is a relationship between RL and the effective values of the raw fitness $f^*$. This suggests there may be ways of analyzing LCS-like systems using the fitness matrix ($\left[F\right]$) concept introduced above.

### Generalization and Neighbourhood Structure

The neighbourhood structure induced by generalization/specialization relationships is critical to both the implicit fitness sharing in Holland's original LCS (and related, simpler schemes like ZCS [20]), and the mating restriction in XCS. However, in XCS, there is another significant effect. An agent is allowed to generate new agents (that is, participate in the EC process) only after a certain number of matches. Therefore, more "general" agents have more chances to produce "child" agents. This acts as a balancing effect against the accuracy-based fitness measure, which favours more specific agents. In general, this "triggering" of EC interactions is another important effect. Of course, this can be considered to be an aspect of the selection scheme for agents, as well.

The neighbourhood structure also allows for the potential formation of *default hierarchies* where specialists correct potential errors of generalists, thus forming a symbiotic relationship with desirable system level outcomes.

Finally, note that the overall goal of LCS-like systems is to maximize some overall performance measure (e.g., the mean reward delivered by the environment). However, it is often the case that one wishes this to be accomplished with a set of agents that are as general as is possible. Note that this is a structural goal that is not generally reinforced by the environment directly. Also note that this structural goal is not too distant from the general goal of forming a compressed, robust characterization of a control strategy, without embedded overfitting or underfitting (as in neural networks and inductive learning schemes).

## *Chains of Action and Other RL Effects*

As was noted above, one can include reinforcement for agents for good "sequences" of actions, in a manner that is like general reinforcement learning schemes. Of course, this induces yet another form of neighbourhood structure: that of agents that act in sequence (*chains*). Such implicit action chain can be made explicit if one includes a means of agents communicating with one another in sequence (via direct communication, or a "bulletin board" in the environment) (see Figure 7).

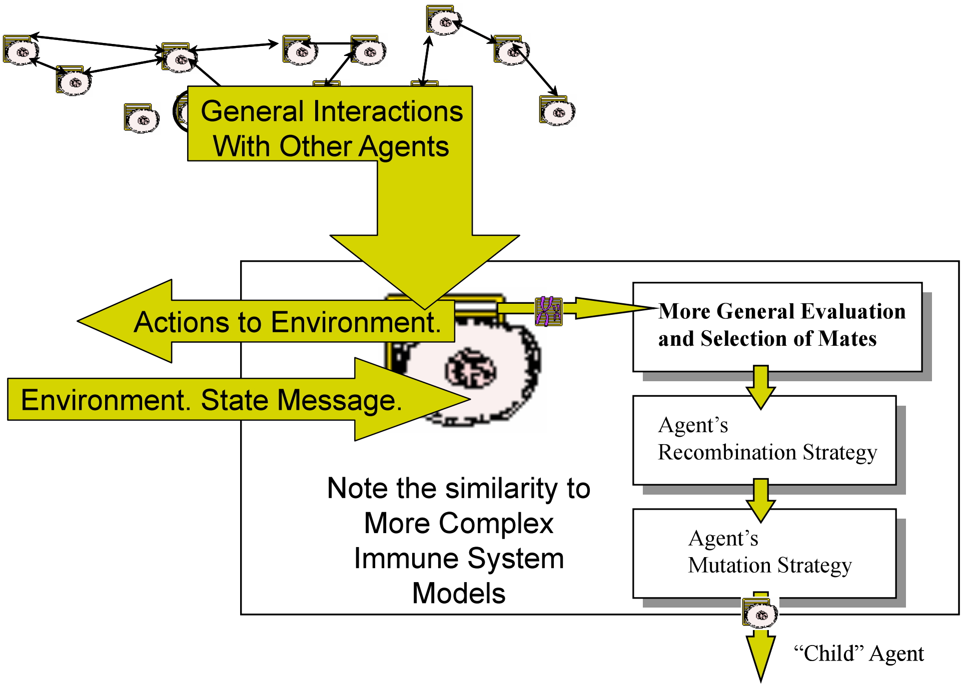

**Figure 7: A more complex, LCS or immune system like context, which includes more general messages between agents, and the prospect for explicit action chains.**

Once one allows for this effect, one can have internal state in the agent-based controller as well. This can allow for internal representation of environments that are non-Markovian.

## Open Questions

To examine open questions in EC, it is useful to first narrow one's concern to the simple optimization example.

In this case, one must ask under what conditions the EC algorithm can be found effective in terms of developing high (possibly optimal) fitness agents, when faced with general and specific environments. This can involve deep consideration of the structures of genotypes and phenotypes in relationship to reproduction functions and the fitness feedback provided by the environment (e.g., so-called *epistasis* and *deception* in genetic algorithms). Closely related to this is the consideration of the interplay of features within the so-called "fitness landscape." Traditional EC thinking suggests that building blocks (features with consistently high fitness) are juxtaposed via recombination to form better solutions. This becomes problematic when many building blocks influence one another's fitness when combined. Thus, there is a large body of EC theory that considers this interplay, particularly via *the Walsh Transform* over the functions that map features (e.g., genotypes) to fitness [16]. Such analysis is vital in understanding what sorts of problems and encodings could prove difficult for various types of EC.

Without digressing into the particulars of problem and encoding spaces, we can generally observe that, in general, an EC algorithm requires a subtle balance of effects:

- **Sampling** – the size of the population of agents
- **Selective pressure** – the manner in which agent biases its selection of mates based on fitness

- **Diversity** – largely as induced by mutation
- **Mixing** – largely as induced by recombination

The investigation of the appropriate balance is (in my opinion) the central concern of EC theory.

The "traditional" EC theory outlined above comprises one (quite primitive, but often useful) type of approach to these questions. In this approach, one considers the *expected* proportion of certain features (or individuals) in a population (often assumed to be of infinite size). In a number of studies, the expected trajectory for the population proportions is analytically evaluated. Although this approach provides only rough guidelines with regard to actual EC algorithm performance, it has often yielded useful observations (notably the theory behind fitness sharing).

This approach can be extended to consider not only the expected value trajectories, but other statistics of the evolving population [13].

Another type of approach is Markov Chain Analysis of EC. In such approaches, the state of the entire EC population is mapped onto a state in a Markov Chain, and state transition probabilities are formulated relative to the genetic operators over the population. Once this is done, the full spectrum techniques for analyzing the trajectory of Markov Chains and other mathematical techniques can be brought to bear [19].

When one begins to more deeply consider speciation and niching (via fitness sharing, mating restriction, or other operators), one becomes concerned with the additional goal of spreading the population of agents across peaks in the space. Of immediate concern are the

- **The neighbourhood structure** - over which sharing or mating restrictions are based. Of course, this has a particular interplay with the previously mentioned effects in the simple optimization example.
- **Sharing and mating restriction functions** – over this neighbourhood structure.

When one moves to more complex interactions between the agents and the environment, two additional effects emerge:

- **Conflict resolution and reward dynamics, -** specifically, which actions are allowed to occur in the environment, and how they are rewarded
- **Fitness calculation dynamics**– since a fitness is not assigned directly, and variety of fitness measures can be derived from reward. Note that these measures can be distinct from those used in conflict resolution.

The final, and possibly most important, feature of such systems is the co-evolution of individuals that have desirable relationships to the environment. In the abstraction offered here, this is directly related to the hierarchical generalization/specialization neighbourhood structure. I sincerely believe that deeper investigation of these co-evolutionary aspects of EC systems is the most needed aspect of EC theory.

## Final Comments

As a direction for further research, I suggest that in agent-based EC systems. and in some systems that could be characterized in this fashion (possibly XCS), it may be possible to characterize the matrix $\left[F\right]$ in relationship to the hierarchical nature of individuals, and the statistics upon which the centralized view of fitness is based. With such a characterization, it may be possible to make statements of the nature and stability of steady states in such systems.

## Acknowledgements

The author would like to thank Ludwig Streit for inviting me to Math Encounters at The University of Madeira, where I had the opportunity to refine these ideas, and discover many new avenues for future research. In light of this, I would like to formally acknowledge the support provided by FCT, Portugal.

## References

[1] Back, T., Fogel, D. B., and Michalewicz, Z. (1997). *The Handbook of Evolutionary Computation*. Oxford University Press.

[2] Barto, A. G., Sutton, R. S. (1998). *Reinforcement learning*. MIT Press.

[3] Blackmore, S. J. and Dawkins, R. (1999). *The Meme Machine*. Oxford University Press.

[4] Dawkins, R. (1990). *The Selfish Gene*. Oxford University Press.

[5] Deb, K. (2000). *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley.

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

*[7]* Forrest, S., Javornik, B., Smith, R. E. and Perelson, A. (1993). Using Genetic Algorithms to Explore Pattern Recognition in the Immune System. *Evolutionary Computation* 1(3). pp. 191-212

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

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

[10] Holland, J. H. (1992). *Adaptation in natural and artificial systems (second edition).* MIT Press.

[11] Horn, J. and Goldberg, D. E. and Deb, K. (1994). Implicit niching in a learning classifier system: Nature's way. *Evolutionary Computation*, 2(1). p. 37-66.

[12] Krasnogor, N. and Smith, J. (2000) MAFRA: A Java Memetic Algorithms Framework, *Workshop on Memetic Algorithms*, GECCO 2000.

[13] Prügel-Bennett. A. and Rogers, A. (2000). Modelling GA Dynamics. *Theoretical Aspects of Evolutionary Computing*, Natural Computing. Springer-Verlag.

[14] Radcliffe, N. J. (1997). Schema Processing. In [1], pp. B2.5:1-10.

*[15]* Smith, R. E. and Forrest, S. and Perelson, A. S. (1993). Searching for Diverse, Cooperative Populations with Genetic Algorithms. *Evolutionary Computation* 1(2). pp. 127-149

[16] Smith, R. E. and Smith, J. (2001). New Methods for Tunable, Random Landscapes. In Martin, W. N. and Spears, W. M., Foundations of Genetic Algorithms 6. pp. 47-68. Morgan Kaufmann.

[17] Smith, R. E., Bonacina, C., Kearney, P., and Merlat, W. (2001). Embodiment of Evolutionary Computation in General Agents. *Evolutionary Computation* 4(8). Pp 475-493. MIT Press.

[18] Sutton, R. S. and Barto, A. G. (1998) *Reinforcement Learning: An Introduction*. MIT Press.

[19] Vose, M. (1999). *The Simple Genetic Algorithm: foundations and theory*. MIT Press (1999).

[20] Wilson, S. W. (1994) ZCS: A zeroth-level classifier system, *Evolutionary Computation* 2(1). pp. 1-18.

[21] Wilson, S. W. (1995). Classifier fitness based on accuracy. Evolutionary Computation. 3(2).

[22] Wilson, S. W. (1996). Explore/exploit strategies in autonomy. In P. Maes, M. Mataric, J. Pollack, J.-A. Meyer and S. W. Wilson (eds.), *From Animals to Animats 4: Proceedings of the Fourth international Conference on Simulation of Adaptive Behaviour*. MIT Press.

[23] Wilson, S.W. (1987). Hierarchical Credit Allocation in a Classifier System. In Davis, L. (ed.) *Genetic Algorithms and Simulated Annealing*. pp. 104-115. Pitman.