Embodiment of Evolutionary Computation in General Agents

Robert E. Smith
Intelligent Computer Systems Centre
The University of The West of England
Bristol, UK
robert.smith@uwe.ac.uk

Claudio Bonacina
Intelligent Computer Systems Centre
The University of The West of England
Bristol, UK
c2-bonacina@uwe.ac.uk

Paul Kearney
Intelligent Business Systems
Research Group
British Telecom Laboratories
Martlesham Heath, Ipswich, UK
paul.3.kearney@bt.com

Walter Merlat
Intelligent Business Systems
Research Group
British Telecom Laboratories
Martlesham Heath, Ipswich, UK
walter.merlat@bt.com

Abstract

Holland's *Adaptation in Natural and Artificial Systems* largely dealt with how systems, comprised of many self-interested entities, can and should adapt as a whole. This seminal book led to the last 25 years of work in genetic algorithms (GAs) and related forms of evolutionary computation (EC). In recent years, the expansion of the Internet, other telecommunications technologies, and other large scale networks have led to a world where large numbers of semi-autonomous software entities (i.e., agents) will be interacting in an open, universal system. This development cast the importance of Holland's legacy in a new light. This paper argues that Holland's fundamental arguments, and the years of developments that have followed, have a direct impact on systems of general network agents, regardless of whether they explicitly exploit EC. However, it also argues that the techniques and theories of EC cannot be directly transferred to the world of general agents (rather than EC-specific) without examination of effects that are embodied in general software agents. This paper introduces a framework for EC interchanges between general-purpose software agents. Preliminary results are shown that illustrate the EC effects of asynchronous actions of agents within this framework. Building on this framework, coevolutionary agents that interact in a simulated producer/consumer economy are introduced. Using these preliminary results as illustrations, areas for future investigation of embodied EC software agents are discussed.

Keywords

Genetic algorithms, agent-based systems, market-based systems, coevolution.

1 Introduction

Although Holland's (1975) seminal book, *Adaptation in Natural and Artificial Systems* (ANAS), is often cited as the first on genetic algorithms, it is much more directed at general issues of adaptation in systems of distributed, self-interested agents. The book mathematically examines how populations of agents interacting in "systems" can and should adapt. Holland's work envisions systems where large numbers of computational agents interact, pursuing their own, individual interests, while leading to desirable, emergent effects at a system level.

In 1975, the concept of a large population or distinct, self-interested computational agents was an abstract one. In the past 25 years of EC research, the interacting agents in EC systems have largely been contained within special-purpose systems. In such systems, the agents are created by a single user, are usually centrally controlled, and interact in ways that are prespecified, and purpose specific.

However, with the advance of computational power and communications speed, we now live in a computational world where a large number of agents may be working on behalf of even the most casual user: searching for music, comparing pension schemes, purchasing goods and services, identifying chat partners, etc. Moreover, these agents may be collaborating with those of other users, while spawning and managing agents of their own. In more formal settings, a business, academic, or government user may simultaneously employ many software agents to manage workflow, trade goods or information, or collaboratively solve problems. In the future, even relatively simple household appliances may play a role in this churning system of interacting, computational agents.

In this world, Holland's theories (and the EC theories and practices that have developed in their wake) have new implications. Agents that interact according to these theories are no longer locked inside the laboratory conditions imposed by EC researchers and users.

This paper discusses the implications of Holland's adaptive system theories in the realm of general agents. We argue that these implications can be seen in two ways: as tools for understanding general agent interactions and as means for shaping those interactions. However, we also argue that there are critical differences between the special-purpose EC systems considered in the past 25 years and the embodied interactions of general-purpose software agents. Since agent system standards are now emerging, the implications of EC should be considered when embodied within agents that conform to these standards. As a start on this effort, a framework is introduced to provide for EC interactions between *general* software agents that are based on emerging agent standards. This framework allows for EC to take place between agents that are asynchronously created and destroyed by distributed users. The framework allows for EC as an emergent effect in an agent system, without any specific EC program.

Preliminary results with this framework are presented, showing emergent effects that are similar to those of a centralized GA. However, the agent based EC system shows distinct differences from centralized GAs, which are noted. Finally, theoretical and empirical issues that arise from this view of EC are discussed, and future research directions are suggested.

2 Individual Adaptation and Emergent System Adaptation

The key qualities that software agents exhibit are: autonomy, reactivity, proactivity, and social behavior. Moreover, agents have the possibility of mobility in complex network environments, putting software functions near the computational resources they require. Agents can also explicitly exploit the availability of distributed, parallel computation facilities (Wooldridge and Jennings, 1996).

However, these qualities ultimately depend on the potential for agent adaptation. For instance, if an agent is to operate with true autonomy in a complex environment, it may have to react to a spectrum of circumstances that cannot be foreseen by the agent's designer.

Autonomous agents may need to explore alternative reactive and proactive strategies, evaluate their performance online, and formulate new, innovative strategies without user intervention.

Areas where agents could benefit from adaptation are addressed by active research in machine learning (e.g., classification of unforeseen inputs, strategy acquisition through reinforcement learning, etc.). However, many machine learning techniques are focused on centralized processing of databases to formulate models or strategies. In contrast, EC techniques are inherently based on a distributed paradigm (natural evolution), making them particularly well suited for adaptation in agents.

Agent-based systems are filled with self-interested agents that can change, adapt, and learn. Given this, the agents can be expected to interact in ever-changing ways that range from the competitive to the cooperative. One should ask what models of *system* behavior have considered systems comprised of such agents? Clearly, the models that have developed in the wake of ANAS qualify. This section will try to concisely restate some of the basic theories in ANAS, 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 Goldberg (1989). Although these theories have been much debated in the EC literature (Radcliffe, 1997), we believe their conceptual conclusions provide a valuable perspective on self-organizing systems of general agents.

Let us consider a set of agents, each of which selfishly seeks to exploit some limited set of resources. Imagine that each agent's behavior is dictated by a set of (discrete) features, but assume an agent has no particular intelligence about how those features relate to its own success.

Given this situation, a method of biased perturbations seems an obvious avenue for progress for the individual agent. At the systemic level, a set of agents performing such perturbations is executing a biased, population-oriented random walk in a fashion similar to early evolutionary programming (EP) (Porto, 1997) and evolutionary strategy (ES) models (Rudolph, 1997). An evolutionary analogy can be made here without explicit consideration of "birth" and "death" for agents, since an agent perturbing itself can be viewed as producing a slightly mutated child, then allowing the child to replace the parent.

However, this model fails to consider how agents may benefit from one another. Let us assume that agents can exchange information in the form of messages (possibly in a market of information). How might a selfish agent exploit such information? Clearly, a viable strategy is for an agent to attempt to reason about the effectiveness and features of other agents and use any information obtained to bias its perturbation strategy towards more effective regions of the feature space.

Given this outline of an agent's localized perspective, let us consider the resulting global effects. Using notation similar to that in ANAS, let the expected proportion of existing agents containing some subset of features H at some time t be $p(H,t)$. The expected number at time $t+1$ is given by:

$$p\left(H, t+1\right) = P_s\left(H\right)\left[1 - P_d\left(H\right)\right] \tag{1}$$

where $P_s(H)$ is the probability of any individual agent selecting feature H, and $P_d(H)$ is the probability of the feature being disrupted by the selection of other features, or other effects of agents perturbing their feature sets. Note that each of these probabilities may be a function of the proportions of features.

This simple expression makes no assumptions that can be said to be "biological." Our first assumption that vaguely relates to biological analogy is to cast the formula above in terms of a *reproductive plan*. 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. In its simplest (proportional) form, this gives:

$$P_s(H) = p(H,t)\,R(H) \tag{2}$$

where $R(H)$ is a reproductive rate related to the feature's perceived utility across the population of agents. 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.

Inserting (2) into (1) yields:

$$p(H,t+1) = p(H,t)\,R(H)\,[1 - P_d(H)] \tag{3}$$

which is a proportional form of Holland's (1975) schema theorem. This formula does not depend explicitly on the form 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 $R(H)[1-P_d(H)]$ 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? In general, as a function of time (t), is it most desirable for the proportion of agents with a perceived desirable feature to increase:

- linearly: $p(H,t+1) = p(H,t) + C$
- geometrically: $p(H,t+1) = p(H,t) + Ct^D$
- exponentially: $p(H,t+1) = Cp(H,t)$, or
- or superexponentially: $p(H,t+1) = Ct^{Dt}p(H,t)$

with respect to time (ignoring eventual constraints on proportions, of course)?

Holland's (1975) *k-armed bandit* argument 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 $P_d(H)$ relative to the magnitude of $R(H)$. In EC, such features are often referred to as *building blocks*.

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:

$$N_m = \left(\frac{cM!}{m!\,(cM-m)!} \right),\ 0 \geq m \geq cM \tag{4}$$

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:

$$Np^{cM/2} = 1,\ \ N = \left(\frac{1}{p}\right)^{cM/2} \tag{5}$$

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:

$$N_{bb} \geq \left(\frac{1}{p}\right)^{cM/2} \left(2^{cM-1}\right) \tag{6}$$

Rearranging yields:

$$N_{bb} \geq \left(\frac{1}{2}\right) \left(\frac{1}{p}\right)^{cM/2} \left(\frac{1}{p}\right)^{\left[log_{(1/p)}2\right]cM} \tag{7}$$

or

$$N_{bb} \geq \left(\frac{1}{2}\right) \left(\frac{1}{p}\right)^{\left[2\left[log_{(1/p)}2\right]+1\right]cM/2} \tag{8}$$

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:

$$N_{bb} \geq \left(\frac{1}{2}\right) N^{\left[2\left[log_{(1/p)}2\right]+1\right]} \tag{9}$$

For binary atomic features, this gives a form of the N^3 lower bound often associated with GAs (Goldberg, 1989; Holland, 1975). 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*.

Each of the theories discussed above has been examined critically and with some controversy in the past 25 years. However, Holland's conceptual conclusions remain clear. If a reproductive plan holds, there is an exponential propagation of low disruption features that are generally perceived to be desirable in a population of agents. Moreover, this form of increase has certain desirable properties. Although the mathematical details can be argued, the suggested directions for considering interactions of general agents are important and allow us to draw on the 25-year legacy of ANAS.

The discussion above only alludes to the simplest aspects of evolutionary models. One can further exploit the analogy of a biological system by assuming there is some finite resource in the agent environment, and that agents with feature H consume this resource, such that the net utility of H degrades as the resource is consumed. We can model this as follows:

$$p(H, t+1) = p(H,t)\, R(H, p(H,t)) \tag{10}$$

where $R(H,p(H,t))$ is some decreasing function of $p(H,t)$. We have assumed that disruption effects are negligible (i.e., H is a building block). In a situation with one or more finite resources, $p(H,t)$ can be shown to come into some (stochastic) equilibrium where:

$$R(H, p(H,t)) = 1 \tag{11}$$

for all features related to this resource constraint. This effect, where features come into balance with respect to resources consumed, is another emergent consequence of selfish agents under reproductive plans. Some EC models have explicitly exploited this emergent "resource balancing" effect (e.g., in multiobjective optimization problems (Deb and Goldberg, 1989; Mahfoud, 1997)). Such effects have also been observed as emergent artifacts of EC system dynamics (Horn et al., 1994; Smith et al., 1993). In the most general situation, the perceived utility of a given feature in one subpopulation of agents will be a function of the presence of other features in another subset of agents. Such systems will yield complex, *coadapted* equilibrium populations of diverse agents. EC systems have exhibited such coadaptation, but analytical understanding of such effects is at the cutting edge of EC research. Moreover, there are other broad branches of EC theory and ongoing research that are applicable to general agents through the analogy suggested in ANAS and reiterated here.

Holland-like models seem an appropriate way to gain insight on the propagation of features through a system of agents and possibly to shape that propagation. Moreover, EC processes implicitly exploit parallelism, while remaining trivial to explicitly parallelize (as in an autonomous agent context). Therefore, EC methods are one of the most natural machine learning techniques to transfer general-purpose, adaptive capabilities to agent-based systems.

However, the effects of EC in laboratory environments are not necessarily the same as the effects implied above for generalized agent systems. The following section suggests that, given the emergence of agent system's standards, we must begin consideration of the EC analogy within these standards, and within real agents based on these standards.

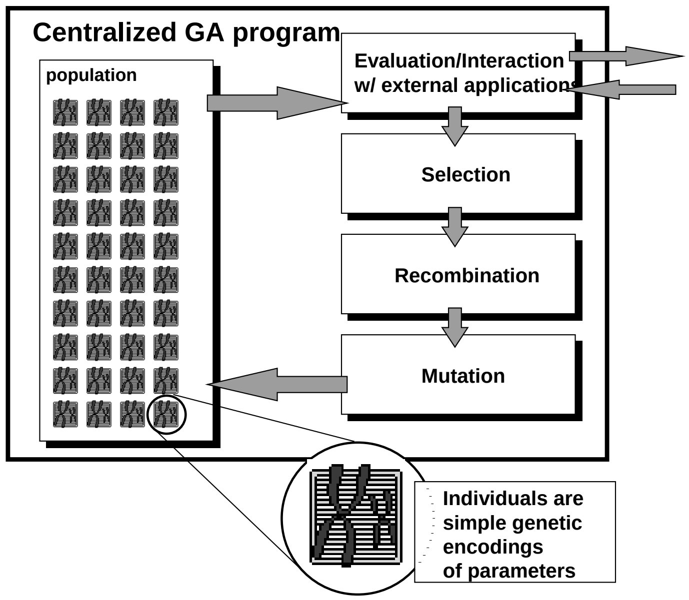

Figure 1: Structure of a typical, centralized GA.

3 Software Embodiment

In many ways, all research on coevolving EC systems is apropos to agent-based systems. However, EC systems that use actual software agents have been rare. This paper will not attempt to enter the debate on what constitutes a software agent (Franklin and Graesser, 1997). Instead, we only suggest that there are general purpose, agent-based system standards emerging, and that few EC systems have been placed within standards-based agent frameworks.

To implement an EC system, a programmer typically writes computer code that creates a population of software entities and manipulates those entities in a prespecified way. This code, and its prespecifications of behavior, are a sort of "bubble" around the software entities. EC takes place within the constraints of that bubble and may not, in fact, be generalizable to the outside of that bubble. If Holland's legacy is to be placed in the general agent analogy suggested in previous sections, we must begin to consider EC and EC-like effects outside of this bubble.

This assertion can be related to the argument for *embodiment* or intelligent robotics (Brooks et al., 1998). This argument would state that one cannot remove the intelligence of an entity from the sensors, actuators, and environment of that entity. In robotics, the philosophy of embodiment has led to the novel, successful, intelligent control systems that are closely coupled to the noise and complexity of real sensors, actuators, and environments.

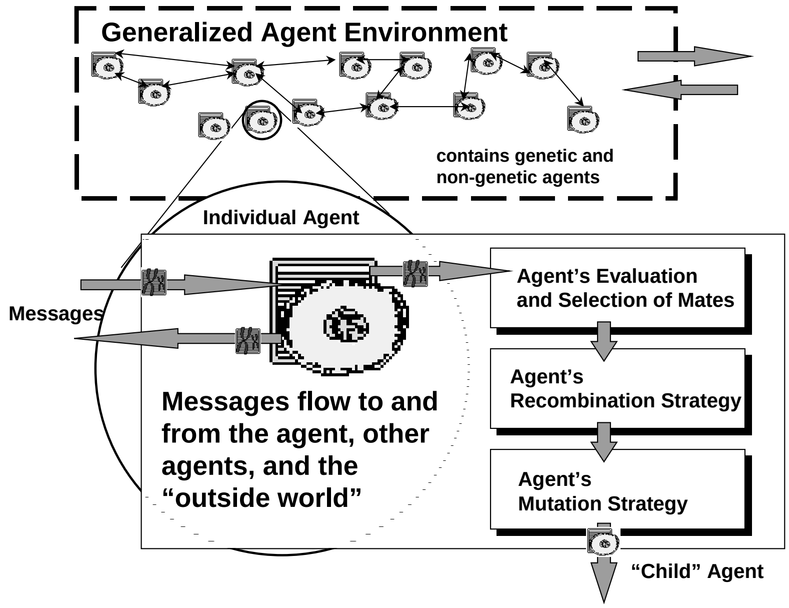

Figure 2: Structure of the EC *embodied* within a general agent system.

To design an agent-based, *embodied* EC system, one must turn the typical GA software design on its head. In common GA software, a centralized GA program stores the GA individuals as data structures and imposes the genetic operations that generate successive populations (see Figure 1).

In natural genetics, the equivalent of the centralized GA does not exist. Instead, the evolutionary effects emerge from individual agents. This makes designing an agent-based EC system a matter of straightforward analogy (see Figure 2).

Note that there are two perspectives one can take on this figure. One is to see it as a literal design of EC interaction objects and methods for a general agent environment. Another is as a viewpoint for examining general interactions of agents. By considering the former, one may be able to gain insight into the latter. However, to consider the possible complexities of general-purpose agents, one must embody those agents in a realistic context.

A clear example is Holland's (1992) ECHO system. Another notable example is Tierra (Ray, 1990), which involves the coevolution of pieces of computer code. Other schemes suggest the idea of evolving agents (Adami, 1998; Balabanovic, 1997; Moukas, 1996; Menczer and Monge, 1999; Eymann et al., 1998).

However, these systems are not designed within general software agent frameworks. With the emergence of standards-based frameworks for general agents[1], it is becoming

[1] Aglets Workbench, http://www.trl.ibm.co.jp/aglets/; FIPA, http://www.fipa.org; Objectspace Voyager,

possible to consider EC as a form of interaction between such general agents. We suggest that by examining *EC-specific* interactions in such frameworks, one can gain insight into *general* agent interactions in the appropriately embodied context. The following sections introduce a framework for examining embodied EC effects in general, standards-based agents.

4 A Particular Framework

The framework introduced here is called Egglets (Smith and Taylor, 1998). The Egglet framework is built in Java, for its machine independence, and on IBM's Aglets framework[2], for basic agent capabilities (e.g., agent messaging, agent storage and recall, etc.). The Aglet framework also allows for agent mobility. Recently, we have developed a generic agent interface that allows the Egglet framework to be used within other standards-based agent systems. Other researchers have transferred the Egglet system to other agent frameworks (Eymann et al., 1998).

It is important to be clear about the purposes of this framework. It will serve as a platform for examining and exploiting *explicit* EC interactions of autonomous, standards-based agents. However, it is hoped these examinations will allow for broader understanding of an EC model of general agent interactions, where EC data structures and operations may not be explicit.

The most fundamental aspect of the framework is the addition of two interfaces that can be implemented for general objects. These are `Randomizable`, and `Mutatable`. Simply stated, the system can be assured that any object that implements `Randomizable` has a randomize method, which generates a random instance of the object. The system can be assured that any object that implements the `Mutatable` interface has a mutate method, which randomly perturbs the object in some fashion. The two interfaces allow an agent to pursue a strategy of biased perturbation of data objects that implement these interfaces.

To account for exchange of pseudo-genetic feature sets, there is an interface `Sperm` (which described genetic feature sets that are sent as messages between agents). To account for biased combination of `Sperm` with an agents own features, there is an interface `Egg`. These interfaces specify the behaviors suggested by their names. However, note that the individual agent who creates an object that implements this interface determines the specific details of how an `Egg` operates. An `Egg` need not simply recombine two feature sets, it may process arbitrary numbers of feature sets in arbitrary ways.

A final key element of the framework is a `Plumage` interface, which agents use to advertise themselves for "mating." Note that how an agent evaluates and selects mates (through evaluation of `Plumage` objects) is entirely an internal function of that agent. Moreover, the model allows for different agents to have entirely different mating and recombination strategies.

There are a number of other details of the Egglet framework that are discussed in Smith and Taylor (1998). These are not included here for the sake of focus and brevity. However, note that the interfaces in this system could be applied to many different types of data objects, ranging from the simple strings of bits typically used in GAs, to tree structures (as in genetic programming (Koza, 1992)), or more complex data objects.

---

http://www.objectspace.com .

[2]Aglets Workbench, http://www.trl.ibm.co.jp/aglets/ .

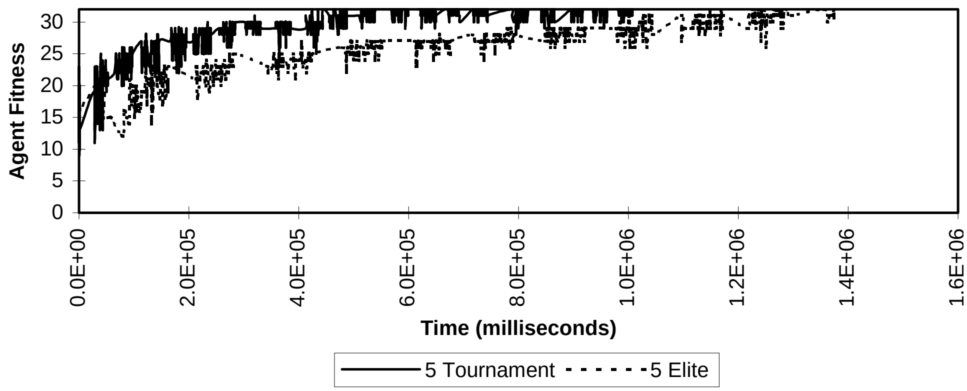

Figure 3: Results with an embodied EC agent framework, applied to a simple task, with two different agent mating strategies.

5 An Embodied Agent Mating Strategy

Note that in the embodied agent context, mating is not actively controlled from outside the agent. Instead, each agent undertakes its own internal mating strategy. The dynamics of this asynchronous (and possibly diverse) activity is an important aspect of embodied agents that deserves consideration.

As a preliminary, basic experiment, we employ the following strategy:

• If an agent has received fewer than 5 Plumage objects from other agents, it sends out a Plumage object of its own to another, randomly selected agent.

• If the agent receives a Plumage object, it replies with a Plumage object of its own.

• An agent adds Plumage objects (both solicited and unsolicited) to a list, sorted based on the agent's mating preference.

• If an agent has 5 (or more) Plumage objects on its list, it gets the Sperm object associated with the highest ranked Plumage, creates a child with this Sperm, and dies.

Note that this is similar to tournament selection in a typical GA but not identical in its effects. In a variant of this strategy, the agent will only add a Plumage to its list if the agent judges this Plumage to be as good as or better than its own. This is similar to elitist selection in typical GAs but in a localized manner.

This brings out a key point. Mate evaluation and selection are very different processes in an embodied EC system, when compared to a typical EC system. This effect is complicated by the latencies and asynchronous nature of message interchange in general agent systems. Certainly, the level of selective pressure (versus time) is different in these schemes. Although related to past work on parallel GAs, this remains an important area for further consideration within a framework of general purpose agents.

In our first experiment, all agents are created to have an underlying binary genotype,

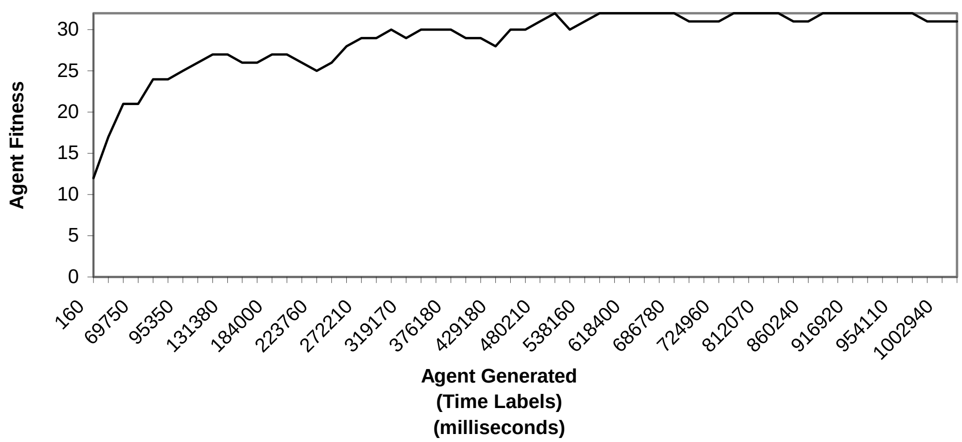

Figure 4: A single maternal line from the experiment shown in Figure 3 (without localized elitism).

to advertise Plumages that directly indicate the count of ones in this genotype, and to prefer mates with Plumages that indicate higher counts. In effect, this is the simplest of GA problems, the so-called OneMax problem. All agents trust one another on how their Plumages represent their counts of ones. Issues of agents misrepresenting themselves, and other issues of trust and security, are, of course, important areas for further research with embodied EC agents.

Results from a population of 20 agents of this sort are shown in Figure 3. Egg objects in these agents essentially implement simple point crossover and point mutation with probabilities 1.0 and 0.01, respectively.

Note that these lines are plotted versus real time (rather than GA generation number), since the concept of a generation does not exist in this asynchronous system. The results show (as expected) that the distributed, asynchronous interests of these agents yield effects that are similar to those in a centralized GA. However, note that the effects of the two selection schemes are not what we would usually expect. In a traditional, centralized GA, one would expect an elitist strategy to have faster convergence. However, in the agent-based results shown above, the scheme that is somewhat analogous to elitism converges much more slowly. Note that this convergence is measured relative to real time. The latencies and additional messages necessary for an agent to locate mates superior to itself are substantial, relative to any increased selective pressure gained by this localized elitism. In a simple way, this shows how the embodied effects of real agents can override EC intuition. In more complex agent systems, these effects will (of course) be more complicated. This points directly to the need to examine embodied agents in the EC context.

It is interesting to consider these results in another light. Figure 4 shows the fitness of a single "maternal line" of the previous simulation. Imagine this line of agents remains the "property" of a single user. This user could have genetic interactions with other users, through the framework. Such interactions would improve those agents belonging to the user.

This illustration points out an important aspect of general software agents in light of EC. Even if there is no *explicit* production of new "child" agents through a set of agent interactions, changes in an agent that result from interchanges of information with other agents can be seen as evolutionary *in effect*. This suggests the broad implications of EC theory and practice in the field of agent-based systems. Moreover, it indicates how selfish agent interactions can have beneficial EC effects for distinct, possibly distributed users.

Note that the particular processing that an agent might perform on a message from another agent does not preclude this sort of interpretation. For instance, consider an individual that uses knowledge acquired from experience to manipulate the Sperm obtained from a potential mate before recombining its features with those in the Sperm. Despite this manipulation, the analogy suggested in previous sections and many of the theories and practices of EC still hold.

6 Considering Fitness Dependence in Embodied EC Agents

The agents in the previous experiment have independent fitness values. Moreover, their only interactions are genetic. This is, of course, a far cry from the general agent interactions that we have suggested considering in the light of EC. Other work by the authors is beginning to consider the interdependency of agent utilities and the resulting coevolutionary effects.

The work presented here was performed as a part of the Dynamo[3] team in British Telecom's Intelligent Business Systems Research Group. This team is interested in multi-agent systems as a means of reducing the complexity of the software infrastructure supporting BT's business process, yielding decentralized systems that are robust, adaptive, and scalable. In Dynamo, business processes are modeled as economic networks of autonomous agents. The work presented here is a preliminary effort to explore the implications of EC within such systems. Note that this work can be considered in relationship to the economic analogies that have been employed in Holland's *learning classifier systems* (Holland et al., 1986). Also, this work has a clear relationship to a variety of other evolutionary economics research efforts (Arthur et al., 1996, 1997; Bull, 1999; Lebaron et al., 1999).

The problem setting is as follows: agents interact in a specific *economy*. Each agent in the system has a prespecified number of *workers* at its disposal. These workers can be allocated to any one of M possible *technologies*. There are N possible *goods* in the economy. Each technology converts one set of goods into another set. Technologies are a prespecified aspect on the economic world.

From an agent's perspective, the goal is to allocate its workers and maximize its *profit*. An agent must perform this task, given that the *price* of each good varies with time, with the action of other agents and with external market forces. Each of these price variations takes place through a market process of supply and demand.

From the perspective of the overall system, one goal is to demonstrate the evolutionary emergence of interesting, productive, economic interaction amongst selfish, coevolved agents. Another goal is to examine variations in emergent behavior based on variations in system parameters and individual agent behavior.

[3] Dynamo stands for the Dynamics of Adaptive Market-based Organisations.

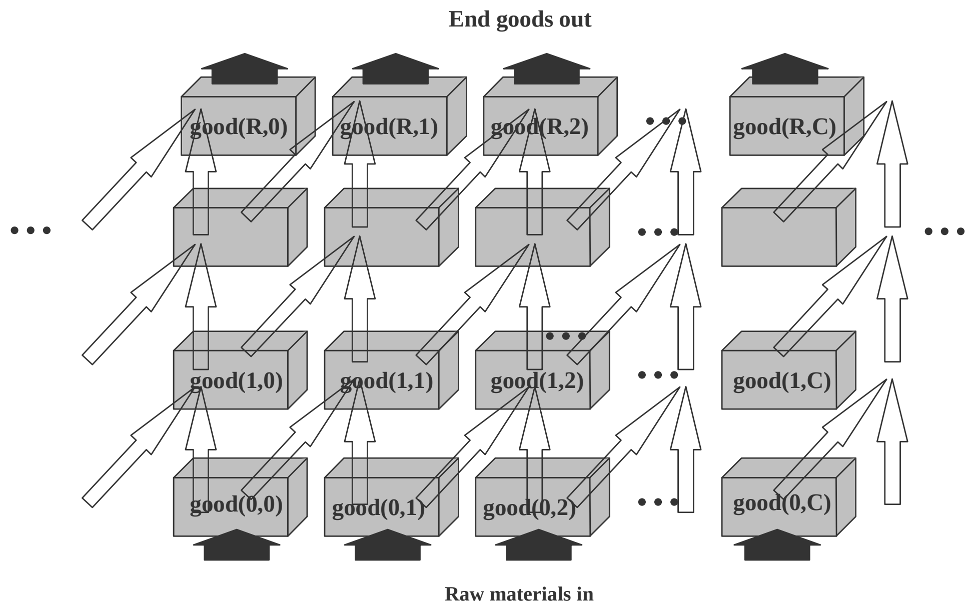

Figure 5: The simplified economic world used in preliminary experiments.

The agents in this system are referred to as producerAgents. Each producerAgent responds to "Current Prices" messages from a market. The producerAgent determines a worker allocation based on this current information and its own genotype. The producerAgent then submits an order message to the market, with negative and positive quantities representing *sell* and *buy* orders for each good.

When triggered by an internally defined condition, a producerAgent searches out mates and attempts to produce child producerAgents.

To illustrate the basic operation of the producer/consumer economic framework, a simplified economic world was constructed.

In this world, the effect of supply and demand is simulated by basing each good's price P on a simple function of the market's excess store of that good S, which can take on negative or positive values. Agents sell and buy all goods from the market, and prices in the market are set as follows:

$$P = e^{-\lambda S} \tag{12}$$

where λ is a parameter.

Prices are only recalculated at the end of a *trading day*. A trading day begins with the market broadcasting a current price message to every producerAgent. The market then asynchronously processes orders from each producerAgent that received this current price message. The trading day is complete when all these orders are processed. Prices are then recalculated, and a new trading day begins.

In its genotype, each producerAgent has a Boolean gene for each of the M technologies.

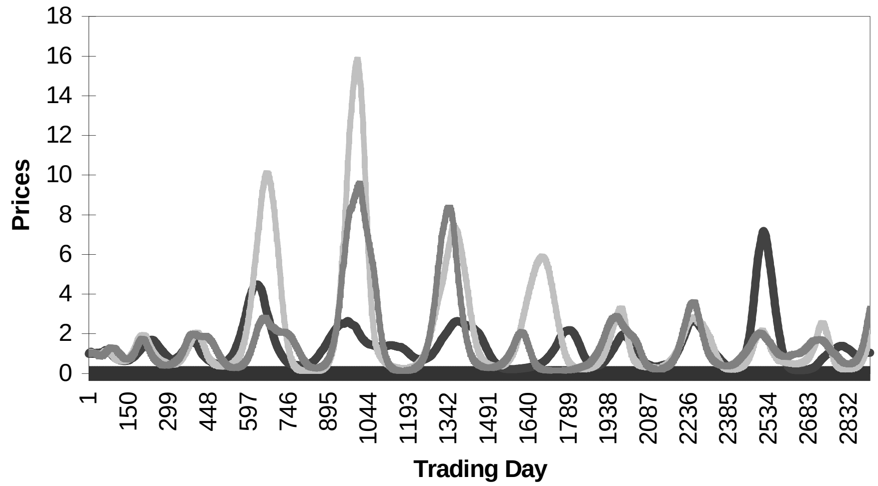

Figure 6: Prices of three intermediate goods as a function of trading day in an evolving economy of producers and consumers.

A producerAgent allocates workers evenly to each technology marked true in its genotype.

In the preliminary experiment, producerAgents trade for 10 trading days, and then they (asynchronously) broadcast and receive Plumage and Sperm objects to locate mates. The tournament-like mating strategy used in previous experiments is employed without local elitism. Note that mating goes on asynchronously, and that producerAgents do not trade while pursuing mates.

In the preliminary experiment, it was desirable to have a set of technologies and goods that interacted in a uniform, extensible manner. Therefore, the following scheme was employed. Goods were "stacked" into R rows and C columns, as shown in Figure 5.

In this figure, the unfilled arrows represent technologies. Each technology takes two goods in one row, and converts them into a single good in the next row. Such conversions are conservative, such that x quantity of one good, combined with y quantity of another good, yields $(x+y)$ quantity of the third good. Note that the world "wraps around" at the right and left edges. Thus, the world is expandable via the R and C parameters, and goods in the $R-2$ intermediate rows are treated identically.

The filled arrows in Figure 5 represent external suppliers of raw materials and consumers of end goods. The raw material goods (in the lowest row) and the end goods (in the highest row) are controllable through the supply and demand of external suppliers and consumers. Note that the total number of goods $N=R*C$, and the total number of technologies $M=(R-1)*C$. The preliminary experiment presented here is the minimal meaningful form of this world. There is one layer of intermediate goods ($R=3$). For simplicity, $C=3$, giving a total of $N=9$ goods and $M=6$ technologies. External supplies and demands are manipulated such that there is a fixed price for the raw materials and another fixed price for end goods. In our preliminary experiment, the prices of these raw materials and end goods are both

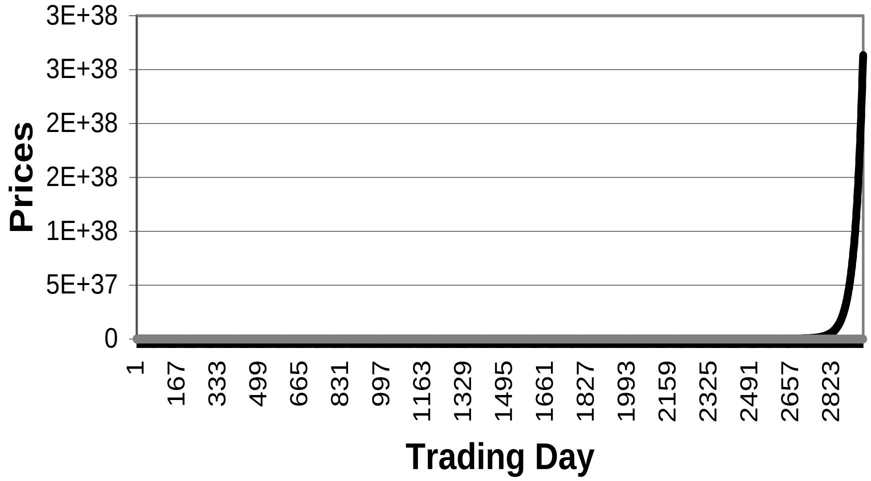

Figure 7: Prices of three intermediate goods with random initialization and no (EC) evolution. Note increased scale versus Figure 6.

maintained at a value of one. Fifty producerAgents are employed.

Prices of the three intermediate goods in this simulation are shown in Figure 6. Variations in these prices are due to evolution of the agents involved in the market. As prices for intermediate goods rise (due to low supply and high demand), agents evolve to produce these high-profit goods. This causes supply to increase, demand to fall, and prices to fall. Then the agents evolve towards being consumers of the low-cost intermediate goods. Note that the agents involved have no memory of price history.

This is an example of the complex interplay of agents whose evaluations of genetic features are a dynamic function of their population's composition. The oscillations seen here are a by-product of ongoing coevolution of the agents. The agent's tend to "chase" one another to profitable areas of their economy, deplete it, and then chase towards the new profitable area this depletion creates. In this sense, these complex behaviors can be seen as similar to those observed in the El Farol Bar Problem (Arthur, 1994; Fogel et al., 1999).

To further illustrate that the complex oscillations emerge from coevolution, consider a similar simulation but with randomly composed agents (i.e., genotype initialized at random), shown in Figure 7.

Since there is an initial, random misbalance in the population of agents, some goods are permanently produced in larger quantities than others, leading to an exponential explosion in some goods' prices and an exponential collapse in others. Similar price explosions occur if the agents are composed in a deterministic but homogeneous fashion (e.g., if their genotypes are set to all "true" values).

Clearly, the system with evolution has more interesting (and desirable) economic interactions of the agents. However, eventual price stability is not achieved. Other studies have noted that it is difficult to get stable, non-oscillatory behavior in complex markets and supply-chain type networks, whether using standard adaptive algorithms from the supply chain literature (Parunak et al., 1998), evolving rule-based systems (Arthur et al., 1996), or real humans in an experimental setting (Sterman, 1989).

In our simulations, the lack of price stability may be due to a lack of price sensitivity and memory in the composite agents. These agents do no logical, differential allocation between the technologies they control, nor do they remember any price history. Experiments with more intelligent agents in this application are an area of current investigation.

Several further experiments with the system used here suggest themselves immediately. In the preliminary experiments shown here, supply and demand are manipulated to maintain constant prices for raw materials and end goods. However, a more realistic simulation would consider fixed flow rates for raw materials and end goods, or flows that are dependent on prices.

Another problem variation lies in the complexity of technology interactions. In the preliminary experiments presented here, there is a uniformity of the economic world, induced by the similarity of all technologies. Breaking symmetries in the economic world, through the introduction of more complex technologies, is an interesting area for investigation.

As noted above, the most significant simplification in the preliminary experiments is that the agents assign their workers blindly, as dictated by their genetic code. All reactivity to price patterns occurs on the evolutionary time scale.

An obvious next step is to build simple price reactivity to the agents. As a first step, this would involve no learning of price patterns during the agent's lifetime. The agent would only have the capacity to shift worker allocation in reaction to current prices, due to a strategy dictated by its genetic code.

A second step would be to allow the agents to adapt internal parameters during their lifetimes. This would involve memory of past price patterns within the agent, and resulting *lifetime learning*. Through such learning, an individual agent would have the capacity to predict and anticipate price patterns.

Given the nature of the EC agents framework presented here, these extensions (and others) are straightforward. In fact, in a separate effort, researchers are already employing a variant of the framework used here to examine agents interacting in a similar economic world, but with interchange of real-valued parameters as pseudo-genetic material (Eymann et al., 1998).

7 Future Study with Embodied EC Agents

The key points we intend to emphasize are the following:

- *General* agent interactions and related changes in agent behavior can be viewed as EC-like interactions, whether or not *explicit* EC is being employed.
- This suggests that 25 years of experience with "contained" EC theory and practice has important implications for the open, more universal, agent-based systems of the future.
- However, to adequately consider these effects, we must begin to *embody* EC capabilities within "real" (standards-based) agents, such that realistic effects can be appropriately

considered, utilized, and transferred to general understanding of EC-like agent behavior.

The final point is key to future research in this area. When general agents exchange genetics-like information (or information that we wish to view as genetics-like), many agent-based systems issues must be directly considered.

Specifically, various market-based system issues are of importance (Sandholm, 2000), including:

- Perceived market value of the material exchanged,
- Trading protocols,
- Exchange contracts,
- Secondary information markets, and
- Mechanisms for insuring trust.

Moreover, issues of (emergent or imposed) social behavior of agents (Conte and Castelfranchi, 1995) also are of importance, e.g.:

- The capacity to reason about other agents' beliefs and goals;
- Emergent (or imposed) adoption of institutions (like *social norms*) and the impact of fixed institutions on evolution and emergence; and
- Agent participation in collective action, implying social, group, and collective commitment, to facilitate the accomplishment of common plans.

Finally, general EC issues are of concern, including:

- The effects of mate selection strategies that operate from within a diverse population of asynchronous agents;
- The effects of interactions between agents with different selection, reproduction, recombination, and mutation strategies; and
- The effects of agents that process and infer information about genetic material, before reproducing or recombining.

In considering the extension of EC paradigms to general agent interactions, each of these facets will constitute important areas for future research.

In conclusion, the theories of Holland, and those of his predecessors, need not be thought of as being limited to the specific class of contained algorithms we now know as evolutionary computation. With the possibility of a future where ubiquitous, semi-autonomous agents interact in a vast computational sea, these theories have new meanings, which require us to refocus and see the last 25 years of research in a new, promising light.

Acknowledgments

Authors Smith and Bonacina gratefully acknowledge support provided through a research grant from British Telecom. The authors would also like to thank Lashon Booker and Rick Riolo for many useful comments that contributed to this paper.

References

Adami, C. (1998). *Introduction to Artificial Life*. Springer Verlag, New York, New York.

Arthur, W. B. (1994). Inductive reasoning and bounded rationality (the el farol problem). *American Economic Review*, 84:406–411.

Arthur, W. B., Durlauf, S. N., and Lane, D. A. (1997). *The Economy as an Evolving Complex System II*. Addison Wesley Longman, Reading, Massachusetts.

Arthur, W. B., Holland, J. H., LeBaron, B., Palmer, R., and Talyer, P. (1996). Asset pricing under endogenous expectations in an artificial stock market. Technical Report 96-12-093, Santa Fe Institute, Santa Fe, New Mexico.

Balabanovic, M. (1997). An adaptive web page recommendation service. In *Proceedings of The First International Conference on Autonomous Agents*, pages 378–385, ACM Press, New York, New York.

Brooks, R. A., (Ferrell), R. B., Irie, C., Kemp, M. et al. (1998). Alternate essences of intelligence. In *Proceedings of AAAI-98 Conference*, pages 961–968, MIT Press, Cambridge, Massachusetts.

Bull, L. (1999). On using ZCS in a simulated continuous double-auction market. In Banzhaf, W., Daida, J., Eiben, A. E., Garzon, M. H. et al., editors, *GECCO-99: Proceedings of the Genetic and Evolutionary Computation Conference*, pages 83–90, Morgan Kaufmann, San Mateo, California.

Conte, R. and Castelfranchi, C. (1995). *Cognitive and Social Action*. Taylor and Francis, London, England.

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*, pages 42–50, Morgan Kaufmann, San Mateo, California.

Eymann, T., Padovan, B., and Schoder, D. (1998). Simulating value chain coordination with artificial life agents. In *Proceedings of ICMAS'98*, pages 423– 424, IEEE Computer Society Press, Piscataway, New Jersey.

Fogel, D. B., Chellapilla, K., and Angeline, P. J. (1999). Inductive reasoning and bounded rationality reconsidered. *IEEE Transactions on Evolutionary Computation*, 3(2):142–146.

Franklin, S. and Graesser, A. (1997). Is it an agent, or just a program? : A taxonomy for autonomous agents. In *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*, pages 21–35, Springer-Verlag, New York, New York.

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

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

Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems*. 2nd Edition, MIT Press, Cambridge, Massachusetts.

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

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

Koza, J. R. (1992). *Genetic Programming: On The Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, Massachusetts.

Lebaron, B., Arthur, W. B., and Palmer, R. (1999). The time series properties of an artificial stock market. *Journal of Economic Dynamics and Control*, 23:1487–1516.

Mahfoud, S. W. (1997). Niching methods. In *The Handbook of Evolutionary Computation*, pages C6.1:1–4, Institute of Physics Publishing, Bristol, England.

Menczer, F. and Monge, A. E. (1999). Scalable web search by adaptive online agents: An InfoSpiders case study. In Klusch, M., editor, *Intelligent Information Agents: Agent-Based Information Discovery and Management on the Internet*, Springer Verlag, New York, New York.

Moukas, A. (1996). Amalthaea: Information discovery and filtering using a multiagent evolving ecosystem. In *The Proceedings of the Conference on Practical Applications of Agents and Multiagent Technology*.

Parunak, H. V. D., Savit, R., and Riolo, R. (1998). Agent-based modeling vs. equation-based modeling: A case study and users' guide. In Sichman, J. S., Conte, R., and Gilbert, N., editors, *Proceedings of Workshop on Multi-agent systems and Agent-based Simulation*, pages 10–25, Springer, New York, New York.

Porto, V. W. (1997). Evolutionary programming. In *The Handbook of Evolutionary Computation*, pages B1.4:1–10, Institute of Physics Publishing, Bristol, England.

Radcliffe, N. J. (1997). Schema processing. In *The Handbook of Evolutionary Computation*, pages B2.5:1–10, Institute of Physics Publishing, Bristol, England.

Ray, T. (1990). An approach to the synthesis of life. In *Artificial Life II*, pages 371–408, Addison-Wesley, Reading, Massachusetts.

Rudolph, G. (1997). Evolution strategies. In *The Handbook of Evolutionary Computation*, pages B1.3:1–6, Institute of Physics Publishing, Bristol, England.

Sandholm, T. (2000). Agents in electronic commerce: Component technologies for automated negotiation and coalition formation. *Autonomous Agents and Multi-Agent Systems*, 3(1):73–96.

Smith, R. E., Forrest, S., and Perelson, A. S. (1993). Searching for diverse, cooperative populations with genetic algorithms. *Evolutionary Computation*, 1(2):127–149.

Smith, R. E. and Taylor, N. (1998). A framework for evolutionary computation in agent-based systems. In Looney, C. and Castaing, J., editors, *Proceedings of the 1998 International Conference on Intelligent Systems*, pages 221–224, ISCA Press, Raleigh, North Carolina.

Sterman, J. (1989). Modeling managerial behavior: Misperceptions of feedback in a dynamic decision making experiment. *Management Science*, 35(3):321–339.

Wooldridge, M. and Jennings, N. (1996). Software agents. *IEE Review*, pages 17–20.