Symbiogenesis in Learning Classifier Systems

Andy Tomlinson
Larry Bull
Faculty of Computer Studies & Mathematics
University of the West of England
Bristol BS16 1QY, U.K.
{andy.tomlinson, larry.bull}@uwe.ac.uk

Abstract Symbiosis is the phenomenon in which organisms of different species live together in close association, resulting in a raised level of fitness for one or more of the organisms. Symbiogenesis is the name given to the process by which symbiotic partners combine and unify, that is, become genetically linked, giving rise to new morphologies and physiologies evolutionarily more advanced than their constituents. The importance of this process in the evolution of complexity is now well established. Learning classifier systems are a machine learning technique that uses both evolutionary computing techniques and reinforcement learning to develop a population of cooperative rules to solve a given task. In this article we examine the use of symbiogenesis within the classifier system rule base to improve their performance. Results show that incorporating simple rule linkage does not give any benefits. The concept of (temporal) encapsulation is then added to the symbiotic rules and shown to improve performance in ambiguous/non-Markov environments.

Keywords
animat, cooperation, evolution, genetic linkage, symbiosis

1 Introduction

> I called this process symbiogenesis, which means: the origin of organisms through the combination and unification of two or many beings, entering into symbiosis
>
> K. S. Merezhkovsky 1920 [15, p. *xx*]

Anton De Bary introduced the term symbiosis to describe the living together of differently named organisms. Although De Bary used the term to include parasitism, where one organism benefits to the detriment of the other(s) involved, it is more commonly used to describe associations in which none of the partners are adversely affected. The most intimate of symbioses result in the partners becoming gentically linked, with their reproduction synchronized. Margulis has presented a symbiogenetic explanation for the origin of eukaryotes from symbiotic prokaryotes [17]. Maynard Smith and Szathmary [18] have further suggested that symbiogenesis was key to at least one other "major transition" [19] in evolutionary history—the emergence of chromosomes through the genetic linkage of symbiotic genes.

Previous computer-based models of symbiogenesis include Ikegami and Kaneko's [14] investigation of "genetic fusion" (the linking of genomes) in a noisy version of Axlerod's [1] Iterated Prisoner's Dilemma. It was shown that (repeated) linking is most effective when the optimal solution can be constructed as a combination of partial solutions. A comprehensive review of natural and artificial symbiogenesis can be found in [6].

In this article we examine the effects of adding the process of symbiogenesis to the learning classifier system architecture [13]. Results indicate that a simple rule-linkage mechanism does not lead to improved performance over an equivalent system with the appropriate rate of rule discovery. Encapsulation is then added to the linked rule sets such that other rules outside of the linked complex cannot share in their functionality; cheats that can potentially exploit the existence of the cooperative structures are excluded. This is shown to lead to improved performance in difficult environments containing ambiguous (non-Markov) inputs.

2 ZCS: A Simple Learning Classifier System

Learning classifier systems (LCSs) are rule-based systems consisting of a population (ecology) of interacting rules each in the form of a condition-action string. System utility is assigned by the external environment and distributed to individual rules through a reinforcement learning algorithm. New rules are generated via a genetic algorithm (GA) [10]. ZCS [25] is a "zeroth-level" LCS without internal memory, where the rule base consists of a number (N) of rules in which the condition is a string of characters from the ternary alphabet $\{0, 1, \#\}$ and the action is represented by a binary string (# represents a don't care). Associated with each rule is a strength scalar that acts as an indication of the perceived utility of that rule within the system. This strength of each rule is initialized to a predetermined value termed S_0.

Reinforcement in ZCS consists of redistributing strength between subsequent "action sets," or the matched rules from the previous time step that asserted the chosen output or "action." A fixed fraction (β) of the strength of each member of the action set ([A]) at each time step is placed in a "common bucket." A record is kept of the previous action set $[A]_{-1}$ and if this is not empty then the members of this action set each receive an equal share of the contents of the current bucket, once this has been reduced by a predetermined discount factor (γ). If a reward is received from the environment then a fixed fraction (β) of this value is distributed evenly amongst the members of [A]. Finally, a tax (τ) is imposed on all matched rules that do not belong to [A] on each time step to encourage exploitation of the stronger classifiers.

ZCS employs two discovery mechanisms, a global ("panmictic") GA and a covering operator. On each time step there is a probability p of GA invocation. When called, the GA uses fitness-proportional (roulette wheel) selection to determine two parent rules based on strength. Two offspring are produced via mutation (probability μ) and crossover (single point with probability χ). The parents then donate half of their strengths to their offspring who replace existing members of the rule base. The deleted rules are chosen using roulette wheel selection based on the reciprocal of rule strength. If on some time step, no rules match or all matched rules have a combined strength of less than ϕ times the rule-base average, then a covering operator is invoked that generates a new matching rule with a random action. The default parameters presented for ZCS, and unless otherwise stated for this article, are $N = 400$, $S_0 = 20$, $\beta = 0.2$, $\gamma = 0.71$, $\tau = 0.1$, $\chi = 0.5$, $\mu = 0.002$, $p = 0.25$, and $\phi = 0.5$.

Thus ZCS represents a "basic classifier system for reinforcement learning that retains much of Holland's original framework while simplifying it so as to increase ease of understanding and performance" [25]. For this reason the ZCS architecture has been chosen to examine the basic behavior of classifier systems with the process of symbiogenesis added. The reader is referred to [25] for full details of ZCS.

3 Symbiogenesis in a Learning Classifier System

Wilson and Goldberg [28] were the first to suggest that rule linkage may help LCSs form symbiotic rule structures, what they termed rule corporations. Grefenstette [9] had pre-

viously observed that the Pittsburgh-style classifier system directly permits evolution of coadapted sets of rules under the genetic algorithm. Wilson and Goldberg suggested that similar benefits could be achieved in a Michigan-style system if for purposes of reproduction classifiers could form cooperative clusters. According to Wilson and Goldberg [28], the rule base of a "corporate classifier system" (CCS) would contain not only single rules, but also clusters or corporations of rules. These corporations would only be reproduced or deleted as a unit, hence synchronization is assumed, and formed by a mutation type operator. For reproduction, the fitness of a corporation would be dependent upon the strengths of its members, possibly the average strength, such that it would be advantageous for rules to link together rather than remain single. If average strength was used to determine the fitness of a corporation then this may be sufficient to encourage corporate linkage as increased stability is generally advantageous in coevolutionary environments (e.g., [4]).

3.1 ZCCS: A Zeroth-level Corporate Classifier System

The nature of a corporate rule structure is such that it must be able to grow and shrink in size or change internally due to the actions of crossover and mutation. Corporations are formed using rules already present in the rule base and there can be any number of corporations in the population at any time, up to a maximum of half the size of the rule base. (This extreme instance can occur if each rule pairs with one other rule.)

Holland [12] has proposed the Echo system, an artificial ecosystem simulation in which agents move from site to site, interacting with each other and local resources. The agents in the system are given the ability to increase in size and complexity as individual agents join "complex aggregates" or merge together to form "macro-agents." The proposed approach to this is to give each simple agent a long chain of "tag" strings in addition to its basic chromosome. Some of these strings will remain dormant in the single agent and will only be activated if the agent joins to form a higher level structure (triggered by a pattern-matching procedure). This suggests one possible approach to the implementation of corporations within a classifier system based on the idea of dormant linkage templates.

In ZCS, a rule consists of a condition, an action, and also a strength value. It is also reasonable to assume that it has some form of reference such as an index number. In this work an implementation of a CCS has been facilitated by adding a few more parameters.

If corporations are viewed as chains of rules, then a rule can at most be directly linked to only two other rules. If this approach is taken then each rule will require two link parameters ("link forward" and "link back") that when active reference other rules within a corporation. These links will be initialized as inactive but when two rules are selected for joining, then one of each rule's links ("link forward" for one rule, "link back" for the other) will be set to reference the other rule. In a corporate classifier system rule linkage is used to encourage associations between rules through the formation of interdependent rule chains.

In addition to this each rule also contains a "corporate size" parameter and a "corporate ID" parameter included to facilitate subsequent processing. Initially size is set to 1 and corporate ID is left inactive. Within corporations, all rules will hold the same values for size and corporate ID, and these are set during the formation of the corporation, either through "corporate joining" or through the action of crossover by the GA. The classifier system keeps a record of how many corporations have been formed and this is used to determine the ID reference for each new corporation.

Initially coupling/linkage occurs panmictically with random probability on each time step, in the same manner as the GA. An initial coupling probability of 0.25 (once every four time steps on average) was decided on but exhaustive testing is required

to determine an optimum rate. This optimum rate is likely to be dependent on such factors as rule-base size, GA activity, and the nature of the task to be learned.

Within the rule base, rules are selected for linkage using a fitness proportional roulette wheel policy with slot size based on strength. A number of possible alternative policies exist for selecting partners to join to, for example, deterministic (based on strength) or random. Previous related work in this area [4] indicates that for arbitrary tasks, these policies tend to offer similar benefits to each other.

If the forward link of the first rule selected, or the back link of the second, is already activated then that rule is already corporate and the corporation is scanned for the appropriate end rule (i.e., the rule in that corporation with an inactive "forward link" or "back link," respectively), and this becomes the selected rule. Furthermore if the first rule is corporate, say belonging to corporation X, then the second rule is selected from the set: [P] − [X], where P represents the population. If this precaution is not taken then there is the risk of forming "circular" corporations.

Based on the proposals of Wilson and Goldberg [28], corporate activity influences the discovery mechanisms but does not directly influence the activity of the production system. For this reason it was decided to give each rule one further parameter, fitness. For single rules this is the same as the strength value, but for corporate rules the strength and fitness values may be different. The strength parameter is used as before by the production system; however GA activity is now guided by rule fitnesses. Within a corporation all rules are given a fitness value equal to the *average* strength of member rules. The rules' strengths, however, are left unaltered.

Having defined the nature of corporations and proposed a method for their formation it is now necessary to determine what modifications must be made to the discovery component. Rule replacement, be it by the cover operator or the GA, like the roulette wheel selection for reproduction, is based on the reciprocal of rule fitnesses, not strengths. If a corporate rule is selected for replacement then the corporation is first disbanded, then the selected individual is tagged for replacement. These are the only modifications required by the covering operator; however, the GA alterations require further attention.

As in ZCS, two rules are selected for reproduction according to a roulette wheel policy based on fitness. When considering linked rules, corporate fitness (i.e., the average fitness within the corporation) is used during selection. The crossover site is selected as usual for ZCS (i.e., single point crossover) and a single offspring rule is created from the two parent rules. This differs from the original ZCS (which produces two children from crossover) but the rate of genetic input (rule replacement rate) is consistent with ZCS as the GA rate is set to 0.25 (once every four time steps on average). The new rule inherits one-third of the strength of each parent if crossover is employed (or half of the parent's strength if it is not). The motivations for producing a single offspring were to simplify the mechanisms of this inherently more complex design in its prototype phase.

The offspring rule inherits "equivalent" links to the "link back" of the first parent and the "link forward" of the second parent. These links, however, will have to be set not to refer to rules in the original corporations but to the equivalent rules in the new corporation.

For example, corporation X consists of rules 1, 2, and 3; corporation Y consists of rules 4, 5, 6, and 7 (Figure 1); and rules 2 and 5 are selected for reproduction. The new offspring from crossing rules 2 and 5 is termed rule 8; however rule 2 linked back to rule 1 so the new corporation (Z) will also require a copy of rule 1 from corporation X, and likewise copies of rules 6 and 7 from corporation Y. The copy of rule 1 is called rule 1′, and those of rules 6 and 7 are called rules 6′ and 7′, respectively. Corporation Z produced by this corporate crossover operation contains the following rules: [r1′, r8,

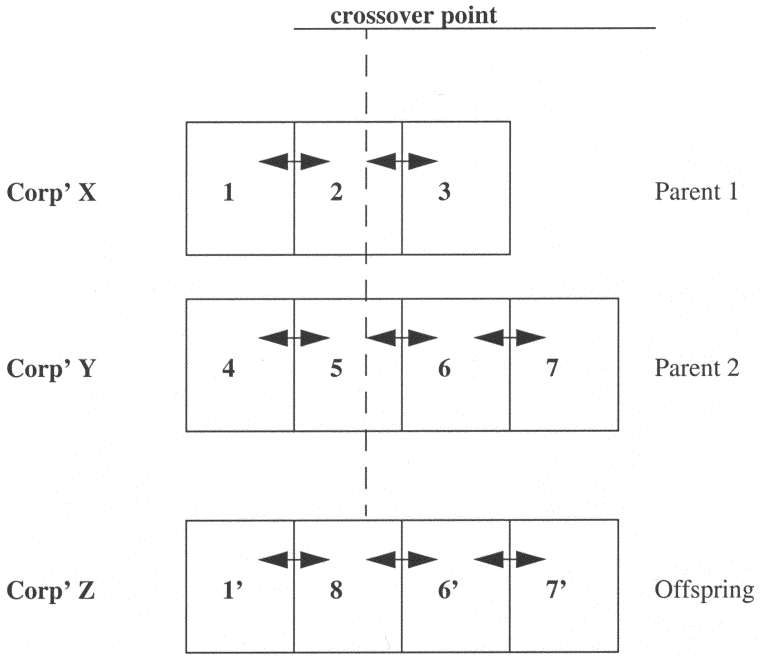

Figure 1. Corporate crossover.

r6′, r7′]. In this way the offspring rule, rule 8, is linked back to the facsimile of rule 1 (rule 1′) and linked forward to the facsimile of rule 6 (rule 6′).

Each additional rule that is reproduced by crossover donates half of its strength to its offspring as above for reproduction without crossover. Mutation is applied only to the new rule derived from crossover (i.e., rule 8 in the example).

The basic ZCS model was modified to act as a prototype corporate classifier system (ZCCS). Modifications were implemented as described in the last section and all other system parameters were maintained as in Wilson's original experiments.

3.2 The Tasks

ZCCS was tested in the same environments as Wilson's original ZCS experiments, Woods 1 and Woods 7. A record was kept of system performance for each trial and also the mean number of corporations active during each trial. It is uncertain what benefits may be demonstrated here by this prototype corporate model, which conforms to Wilson and Goldberg's original specifications. As part of an initial investigation it is still useful to evaluate system performance on these familiar multi-step tasks.

Woods 1 is a two-dimensional rectilinear grid of dimensions 5×5. Sixteen cells are blank, 8 contain trees, and 1 contains food (Figure 2). The classifier system is viewed as an "animat" [24] traversing this map in search of food. At the start of a trial the animat is positioned randomly in one of the blank cells and can move into any one of the surrounding 8 cells on each time step, unless it is occupied by trees. The environment is toroidal so if the animat moves off one edge it appears on the opposite edge of the map. If the animat moves into a "food cell" then the system receives a reward from the environment in the form of credit, and the animat is relocated as before. This signifies the end of one trial and the start of a new one.

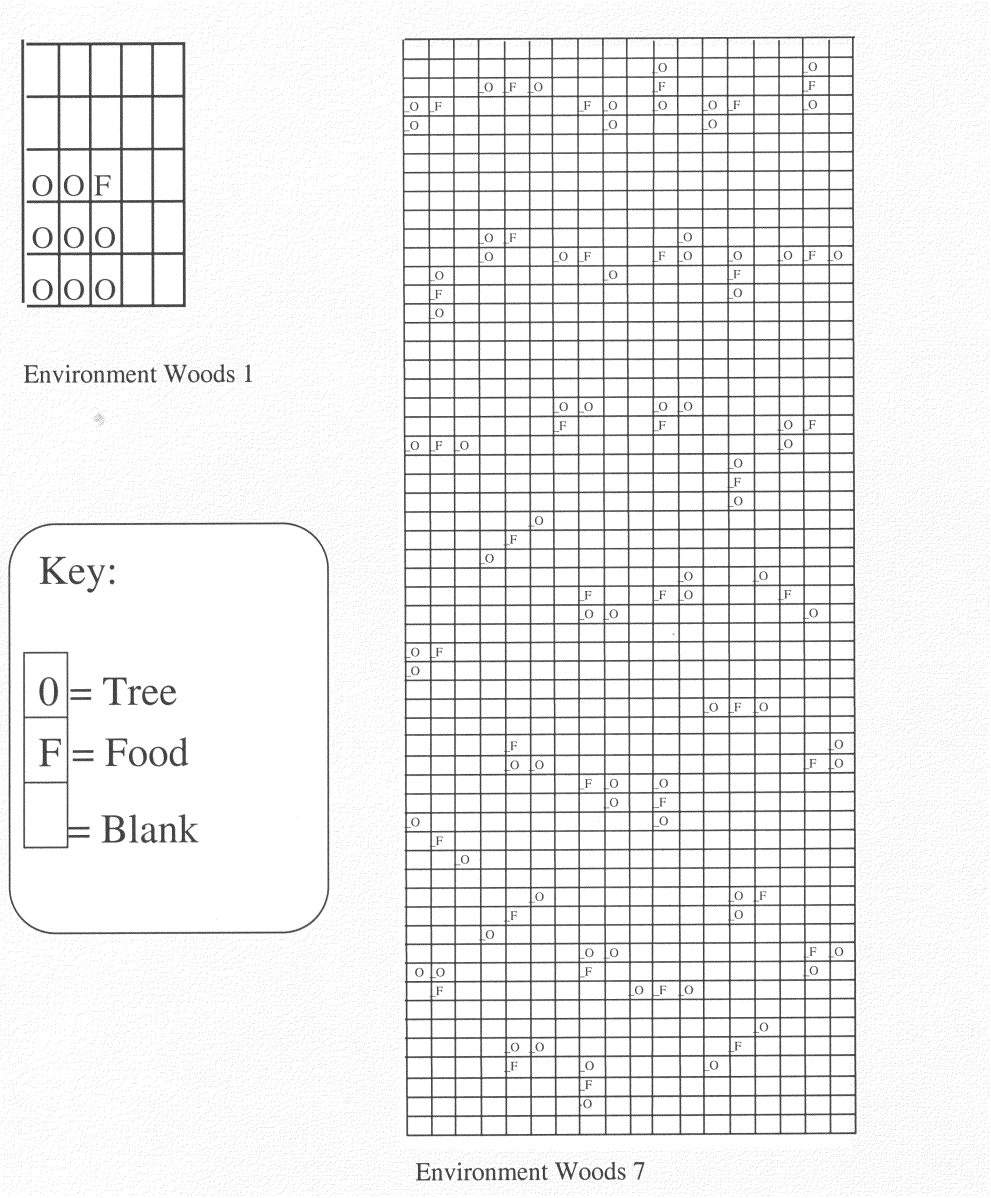

Figure 2. The woods environments.

On each time step the animat receives a message from the environment that describes the surrounding 8 cells. The message is encoded as a 16-bit binary string with 2 bits representing each of the 8 cells. A blank cell is represented by 00, food (F) by 11 and trees (O) by 10 (01 does not occur). The message is ordered with the cell directly above the animat represented by the first bit pair, and then proceeds clockwise around the animat.

The trial is repeated 10,000 times and a record is kept of a moving average (over the previous 50 trials) of how many steps it takes for the animat to move into a food

cell on each trial. If the animat moved randomly then its average performance is about 27 steps per trial. Optimum performance in Woods 1 is 1.7 steps.

Woods 7 (see Figure 2) is a similar yet somewhat more demanding environment than Woods 1. Like Woods 1, it is a toroidal grid but its size has been expanded to 58×18 cells. Evenly scattered around the map are 57 cells occupied by food. Each of these cells has rocks positioned randomly in 2 of the 8 surrounding cells. The rest of the map is blank. Unlike Woods 1, this is a non-Markovian environment that provides the system with many misleadingly ambiguous stimuli. In fact, given the animat's sensory abilities there is only so much that the system can learn in Woods 7. Wilson [25] claims that a system with arbitrary memory could reach food in an average of 2.2 steps. A ZCS-type system equipped with a (reasonable) "perfect" rule set (consisting of about 20 rules) can obtain food in 4 steps on average. Random search in Woods 7 reaches food in 41 steps on average. These two environments are used here to examine ZCCS.

In both Woods 1 and Woods 7 there is no discernible difference between the performance of ZCCS and ZCS. Figures 3 and 4 are performance plots of ZCS and ZCCS in the woods environments. The curves are averages of 10 runs with a 50-point moving average filter applied (as [25]). In Woods 1 the number of corporations rose to 40 in 100 trials and then climbed slowly to 80 by the end of the run. In Woods 7 there were 50 corporations after 100 trials but this value dropped to 40 after 500 trials and remained at this level for the duration of testing.

Hence these experiments have demonstrated that it is possible to implement a corporate classifier system as proposed by Wilson and Goldberg [28]. The corporate classifier system used for these experiments can be considered merely a template design kept as minimal as possible. There are many ways in which system design could be expanded or modified to achieve more directed gains based on the concept of symbiogenesis [e.g., 21]. Some of these are now considered.

4 Symbiont Encapsulation

Maynard Smith and Szathmary [19] note that genetic linkage between "naked" replicators is not sufficient for more complex evolutionary structures to emerge. That is, a linked set of cooperating entities at a given physical location are still open to exploitation from neighbors that are not linked to them and hence do not necessarily share in their evolutionary future. Hence, whereas passive localization can initiate cooperative relationships, only through the active formation of a protective membrane can a symbiogenetic entity perpetuate itself effectively. We now consider this within our corporate classifier system by first including an analog to spatial location and then a membrane to exclude such "cheats."

4.1 Logical Structure: Niche-based Linkage

Although the position of the rules within the rule base of an LCS is not significant to their use, they do have logical relationships with each other.

In this section corporations are now encouraged to encapsulate chains of inference. This is similar to the idea that some form of coupling between successively firing rules may be beneficial [7]. Corporate links here take on a temporal connotation and imply that the rules within a corporation are placed so as to fire in succession and thus to map a proposed plan of action during the solution of a multiple time-step problem.

This is achieved by making linkage a niche operation, or more precisely a cross-niche operation. Coupling occurs between subsequent match sets. This means that on time-step t there is a possibility that a rule that matches the current message from the environment may link to a rule that matched the stimulus at time $t-1$. This encourages the structuring of meaningful sequences of rules.

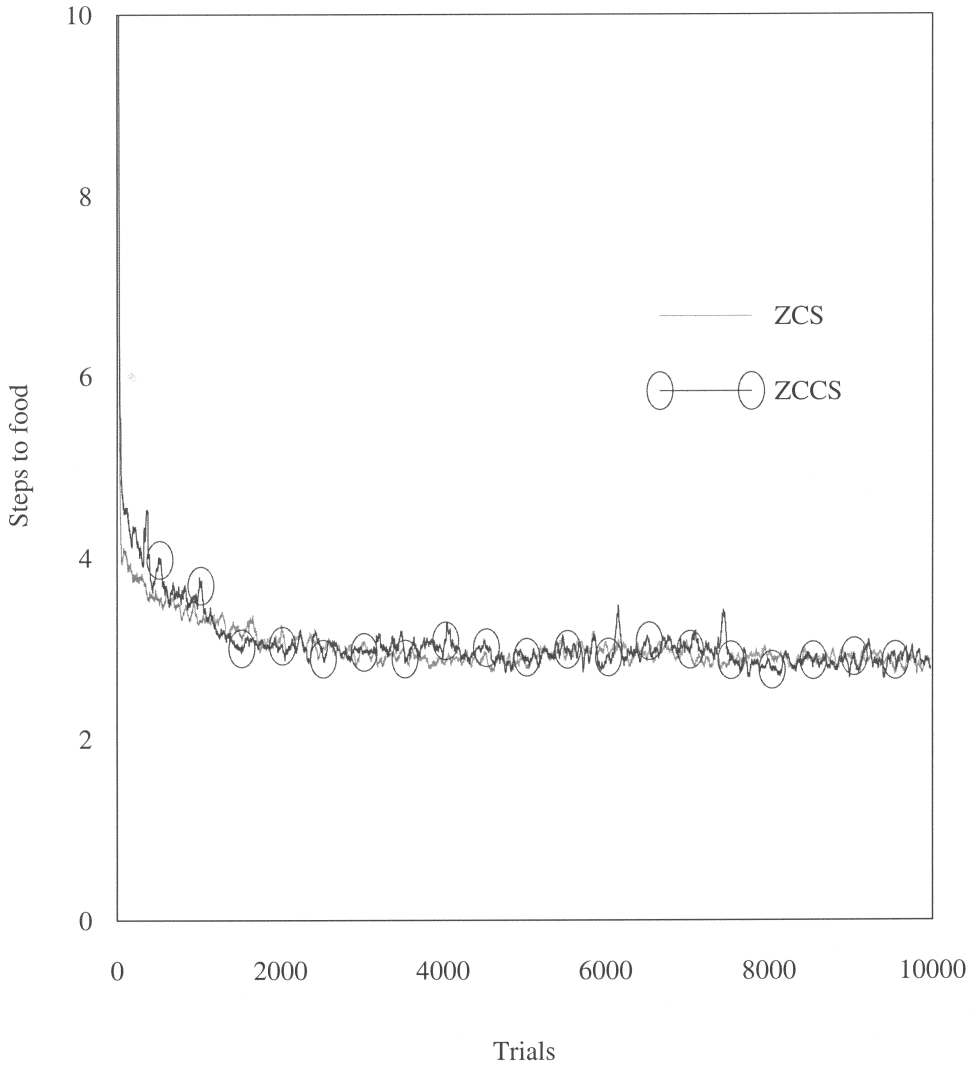

Figure 3. ZCS vs. ZCCS in Woods 1.

To be selected for coupling, a rule must be in the current match set (termed [M]) and its appropriate link must be inactive. Coupling occurs over two time steps. On the first, a rule in [M] is selected probabilistically (roulette wheel, based on strength) from those with an inactive "link-forward"; on the second, a rule in the new match set is selected again probabilistically from those with an inactive "link-back." Rules already in a corporation are not allowed to join to rules within their own corporation.

In all environments used during testing, the system was reset after receipt of a reward from the environment on some time step. Corporate links are not allowed to form between this reward time step and the first one of the following trial as this "move" does not represent any form of causal transition under the control of the system. To further maintain temporal integrity among corporate rule strings the GA is adjusted to operate within match sets. The idea of a niche GA operating in the match set was suggested by Booker [2] and later by Wilson [26] to introduce mating restrictions and

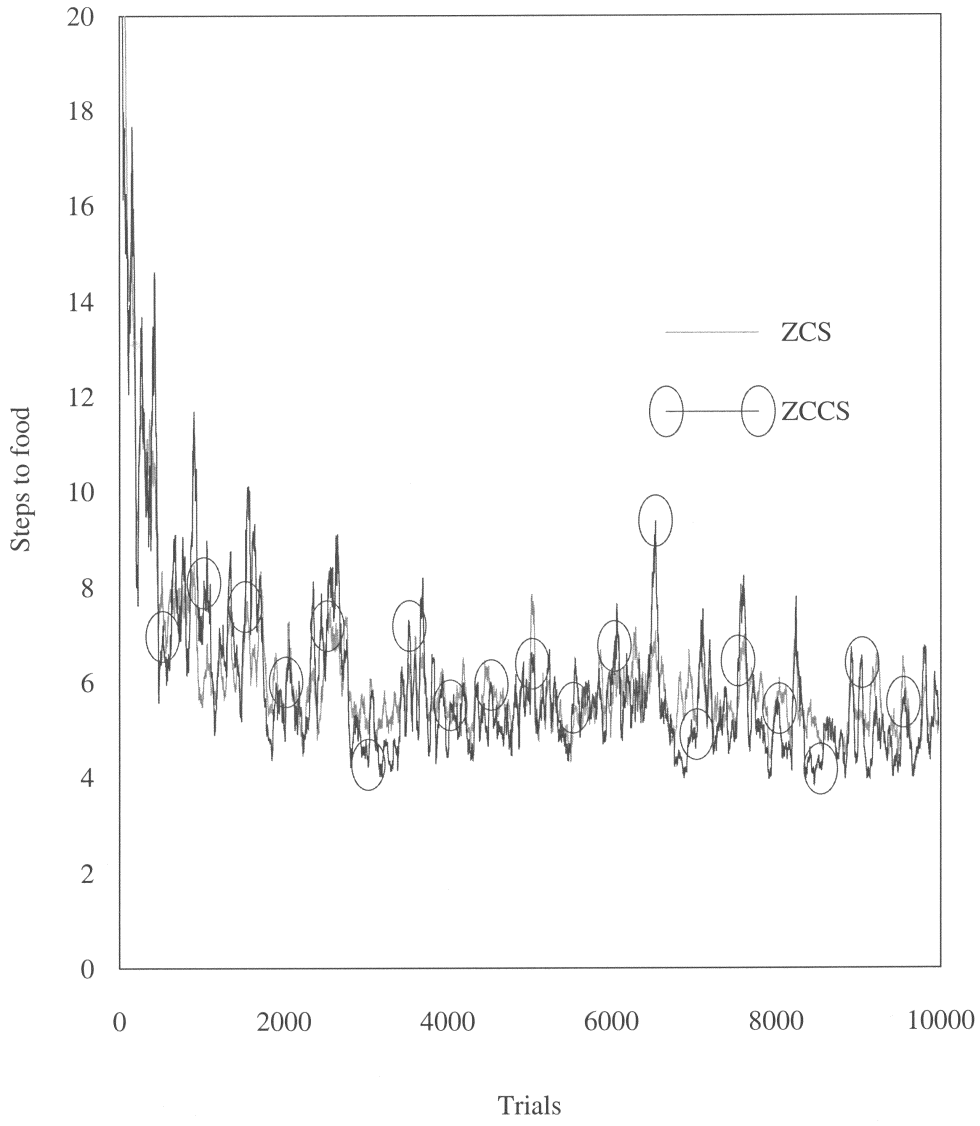

Figure 4. ZCS vs. ZCCS in Woods 7.

thus to assist the GA in producing more meaningful offspring, as like breeds with like. In the corporate classifier system the adjustment is made so that if corporations are selected for crossover then the resultant corporation should still represent a meaningful series of responses to experienced stimuli.

In later work [27] with the XCS design (a system in which classifier fitness is based on accuracy) Wilson further adjusted the GA to operate in action sets. In this mode of operation the GA is used to optimize the generality of rule representations (according to an accuracy criterion). Although it is possible that the corporate classifier system would also benefit from this modification it was considered that the initial design should first be assessed with mechanisms applied to match sets. Future work will consider the merits of moving the GA and linkage mechanism to the action sets.

Preliminary testing of ZCS with a niche GA indicated that GA activity became focused on the more frequently visited states, with the result that niche occupancy for such states with high mean payoff values became excessively large and states with low payoff (especially infrequently visited ones) generally had lower, possibly inadequate niche occupancy, due to the combination of a niche GA and the ZCS replacement policy (based on the reciprocal of rule fitness).

A simple, if somewhat ad hoc solution is to replace rules from within the same niche that the GA was operating in, on the provision that there are already a minimum number of rules representing the niche. It was decided that for all tests this minimum number of rules would be 20. For a population of 400 rules, at least 20 niches can be maintained at this level of occupancy even if all rules are 100% specific. This setting has been found to be adequate for all environments used for testing.

4.2 Adding a Membrane: Temporal Persistence

Early testing of the system with these modifications showed that because of the dissipation of rewards and payoffs among action sets due to the common bucket of ZCS, useful corporations did form but they never fully established themselves within the system and they exhibited lower fitness than their peers within the respective match sets. Consequently their presence made little difference to activities of the performance component and their chances of reproduction were poor (results not shown). In Woods 1 a marginal improvement in performance compared to ZCCS could be observed, possibly due to the introduction of a niche GA; in Woods 7 the modified system demonstrated no discernible performance improvements over the ZCCS design.

Therefore the performance component was adjusted to respond to the presence of corporations. Action selection in the production system is determined stochastically, according to the relative strengths of the rules within the current match set. A roulette wheel policy is employed that selects a rule whose action becomes the system's action. Now, if this rule is corporate and its link forward is active then it is tagged as being in control of the system. On the subsequent time step, if the subsequent rule in the corporation is a member of the new match set then it automatically receives control of the system and forms an action set of size 1. In this way the corporation keeps control of the performance component and is solely responsible for system decisions until either a reward is received from the environment or, on some step, the next rule in the corporation chain does not match the current stimulus. When either of these events occurs the performance component returns to normal operation. Further, "internal" corporate rules (i.e., all but the first rule in the corporation) are flagged as internal and only respond to stimuli during periods when the corporation to which they belong has control of the system. This modification is made to further encapsulate corporate rule structures and thus to reinforce the intercorporate rule codependencies, an analog for membrane formation described above. This mechanism, referred to as "persistence," allows corporations to prove their true worth directly without being interrupted and without the final reward being dissipated among parasitic rules that tend to accumulate in action sets close to rewards. A corporation that indicates a useful series of actions will soon achieve a fitness value that reflects its capabilities.

The final modification to the performance component consists of not charging tax on time steps when a corporation holds control. In ZCS tax is applied to encourage exploitation of the stronger classifiers, and therefore to increase pressure against weaker classifiers. If, due to persistence, performance component control is held by a corporation on some time step then the usual activities of this component have been suspended, that is, action selection on that time step is not based on free competition between competing "hypotheses" in [M]—the appropriate corporate rule is automatically selected to make this decision. In this situation, belonging to [M] and not to [A] is

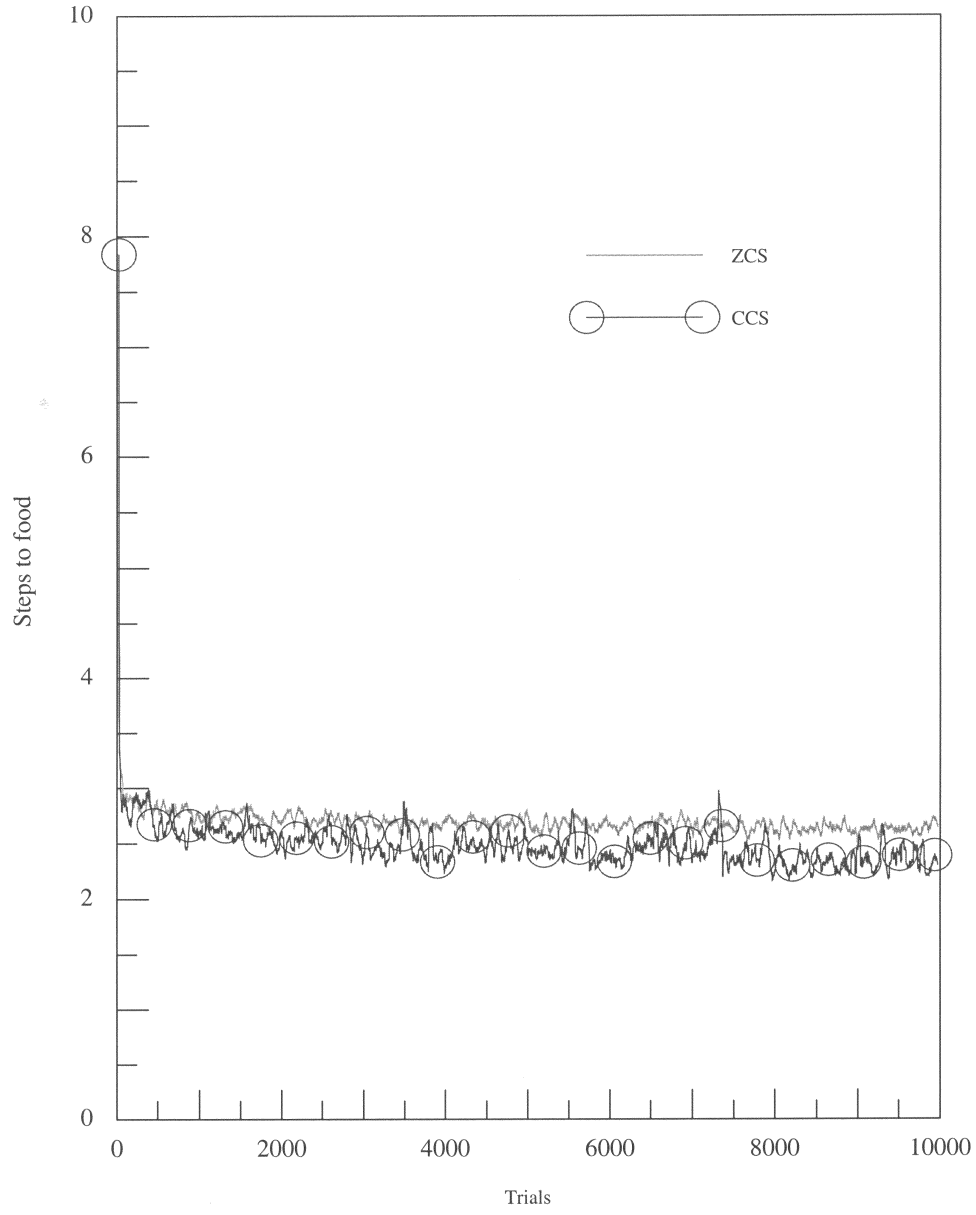

Figure 5. Performance in Woods 1.

not necessarily an indication of low utility, and so it is less appropriate to charge tax to these rules on such a time step. The system was adjusted to include these modifications.

4.3 Results

The modified corporate classifier system (termed CCS as opposed to ZCCS) was initially tested in the two environments used above, Woods 1 and Woods 7. Figures 5 and 6 show graphs of the average steps taken to reach food over 10 runs. In Woods 1 the system performed well, reaching an average of about 2.2 steps to food over 10,000 trials.

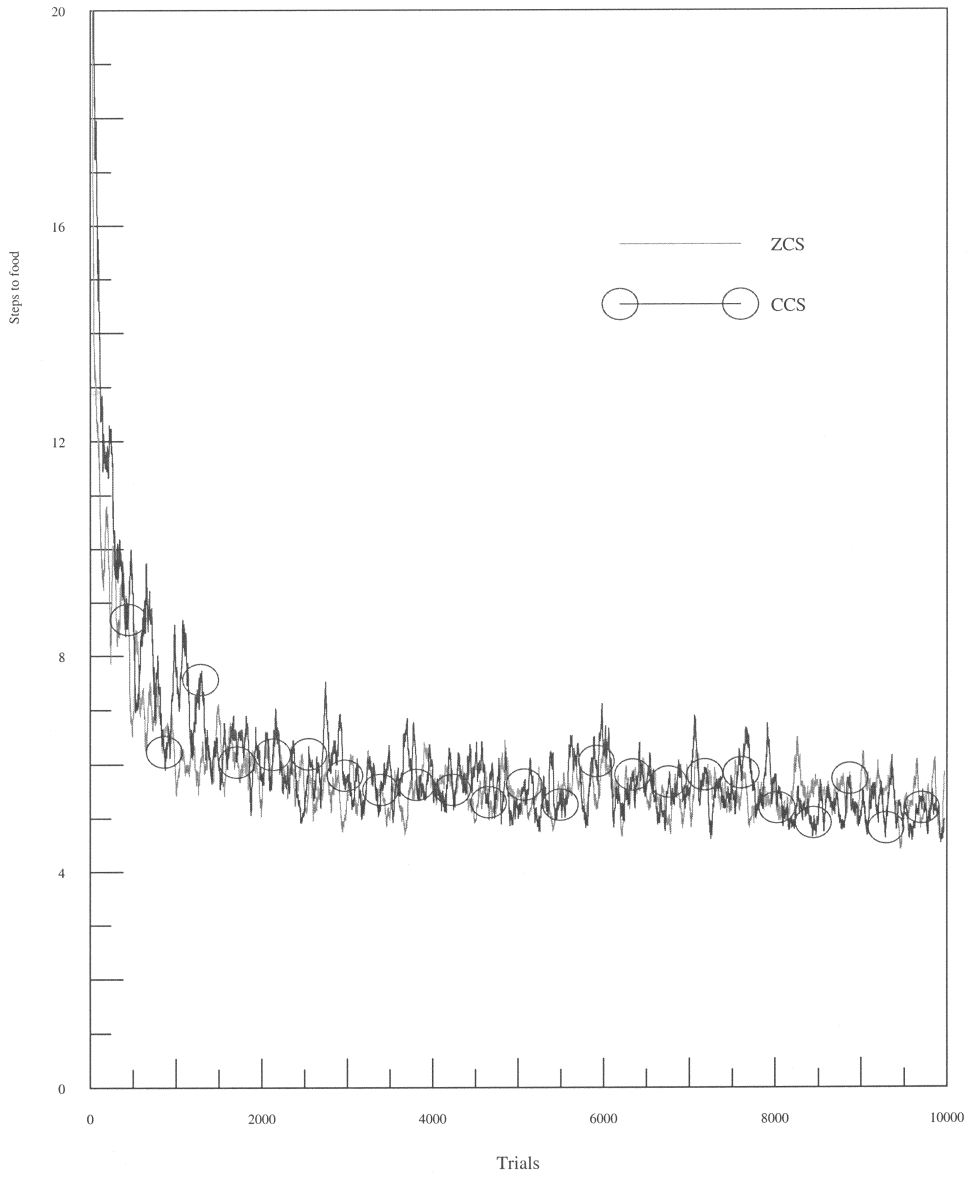

Figure 6. Performance in Woods 7.

The optimum performance in Woods 1 is 1.7, and ZCS achieved an average of about 3 steps to food (see Figure 3). In these tests the ZCS GA was modified to operate in the match set and the rule replacement rate was increased to one rule/time step on average, to facilitate a fair comparison. The modified ZCS achieved a rate of 2.6 steps to food; in this simple Markovian environment CCS can be seen to provide some benefits.

In Woods 7 CCS could not improve on the performances of ZCS or ZCCS. There is, however, a possible explanation for this lack of improvement. CCS is designed to learn and structure sequences of actions to encapsulate useful chains of inference while tackling multi-step tasks. If during a trial the first few steps of the task do not provide any

useful information on which the system can base decisions, it is unlikely that a useful corporation will evolve that could guide the system to more certain terrain in anything like optimum time. If, alternatively, the system, at the beginning of a trial, is given some form of useful information with which it can to some extent discriminate about the immediate environment, then it is possible that corporations will evolve that could steer the system through subsequent "blank" periods toward some eventual reward. In Woods 7 (see Figure 2), at the start of a trial, the animat is positioned randomly in an unoccupied cell. From this cell it must, on average, reach food having traversed 2.2 cells to match Wilson's projected optimum performance for a system equipped with arbitrary memory. However, in Woods 7 it is quite possible that the animat will not have received any information at all in the first couple of steps. When considering the initial random positioning in Woods 7 there is a 55% chance (number of blank cells surrounded on all sides by blanks / total number of blank cells) that the animat will be placed in a cell surrounded by blank cells on all eight sides. In other words for over half of all trials the system is initially provided with no information at all. This pronounced characteristic of the environment suggests that Woods 7 is not a particularly useful measure of system performance.

4.4 Simple Delayed Reward Environments

An alternative test environment is now presented that allows for a clearer differentiation between competing systems' capabilities. The new test is a simple variable multi-step environment. On each of N time steps the system is presented with a stimulus and must select one of A actions, where A is a variable integer value that defines the breadth of a maze. N is the number of states or nodes to a reward and thus defines the maze depth. After N steps, the system receives a reward from the environment and a new task then begins. The size of the reward depends on which route the system chooses and so over time the system learns the optimum reward-yielding route through the maze. There is, however, more than one maze. There can be up to M_z different mazes. The system is informed which particular maze it is being presented with only on the first time step of each trial. On all subsequent steps the stimulus is representative only of the current time step in the trial. The maze is selected randomly at the start of each trial.

Figure 7 illustrates a simple task of this type with A set to 2, N set to 2, and M_z set to 2. The environmental stimulus at each time step is also included. In this example, a reward of 1,000 is awarded for one route on each maze. All other routes receive a reward of 0.

In the example in Figure 7 the message length L is set to 3. With L set to 3 there are eight possible stimuli and so for a two-maze problem the maximum depth will be 7, as the first time step (ts_0) takes two stimuli. Similarly, with L set to 3 a four-maze problem may have a maximum depth of 5, and so on.

Clearly ZCS will be unable to master more than a single maze due to the sensory ambiguity after the first time step; however, CCS should be able to tackle multiple-maze trials. In the task depicted in Figure 7 some form of linkage is necessary for the system to be able to determine the appropriate move on the second timestep (ts_1). In maze 1 the correct action is 0 and in maze 2 it is 1 (the stimulus however is the same—001).

4.4.1 Modifications to the Discovery Component

In CCS, for a fixed GA activation rate, the rule replacement rate becomes a function of the mean size of corporations in the population (dependent on their relative fitness values). To counteract this, the GA activation rate is made variable in CCS. The system is provided with a base rate that is set equal to the GA activation rate of an equivalent ZCS reference model. During training this rate is regularly adjusted according to the levels of cohesion within the rule base.

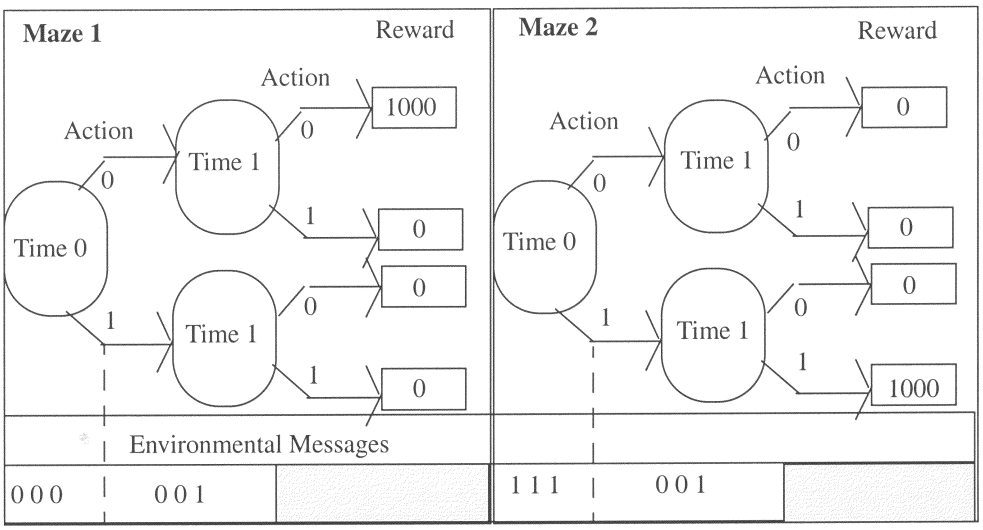

Figure 7. Simple delayed reward environment—task 2:2.

Intuitively, an approach that is likely to provide accurate information regarding rule replacement is to keep a record of the mean number of rules reproduced on each invocation of the GA (i.e., the mean size of offspring corporations). The new GA activation rate is set equal to the base rate divided by the current mean offspring count, hence the resultant replacement rate can be maintained at a reasonably consistent value throughout testing. Each time the GA is activated the mean size estimate is adjusted according to the Widrow–Hoff delta rule (as employed by Wilson to update rule parameters in XCS [27]):

$$S_m = S_m + \beta(S - S_m) \tag{1}$$

where S_m = mean offspring size (initialized to 1), S = size of current offspring, and β = system learning rate.

With, for example, a base GA rate of 0.25 and $S_m = 2$, the resultant GA rate becomes $0.25/2 = 0.125$. So corporations of size 2 tend to be reproduced by the GA in CCS but this will occur only half as often as an equivalent ZCS model would reproduce single rules. The unavoidable consequence of this is a reduction in the crossover (and also mutation) rate in CCS compared to the ZCS model, and this is likely to result in a proportional degradation in the system's rule-discovery capabilities. However, the most significant factor in these comparisons appears to be the rule-replacement rate and its effect on convergence within the rule base, and so to provide a fair indication of relative ZCS performance the variable GA-rate approach has been adopted for all subsequent tests.

4.4.2 Performance Comparison in Delayed Reward Environments

CCS should be able to overcome the ambiguities present in the delayed reward environments. Also presented are plots of ZCS performance for comparison. General system parameters are the same as for the tests in the Woods environments for all systems. Although the initial ZCCS design was tested in these environments, results were not significantly different from those of ZCS and so for clarity have not been included. This is unremarkable as ZCCS lacks the niche pressures of the later CCS design and further, it does not apply corporate persistence.

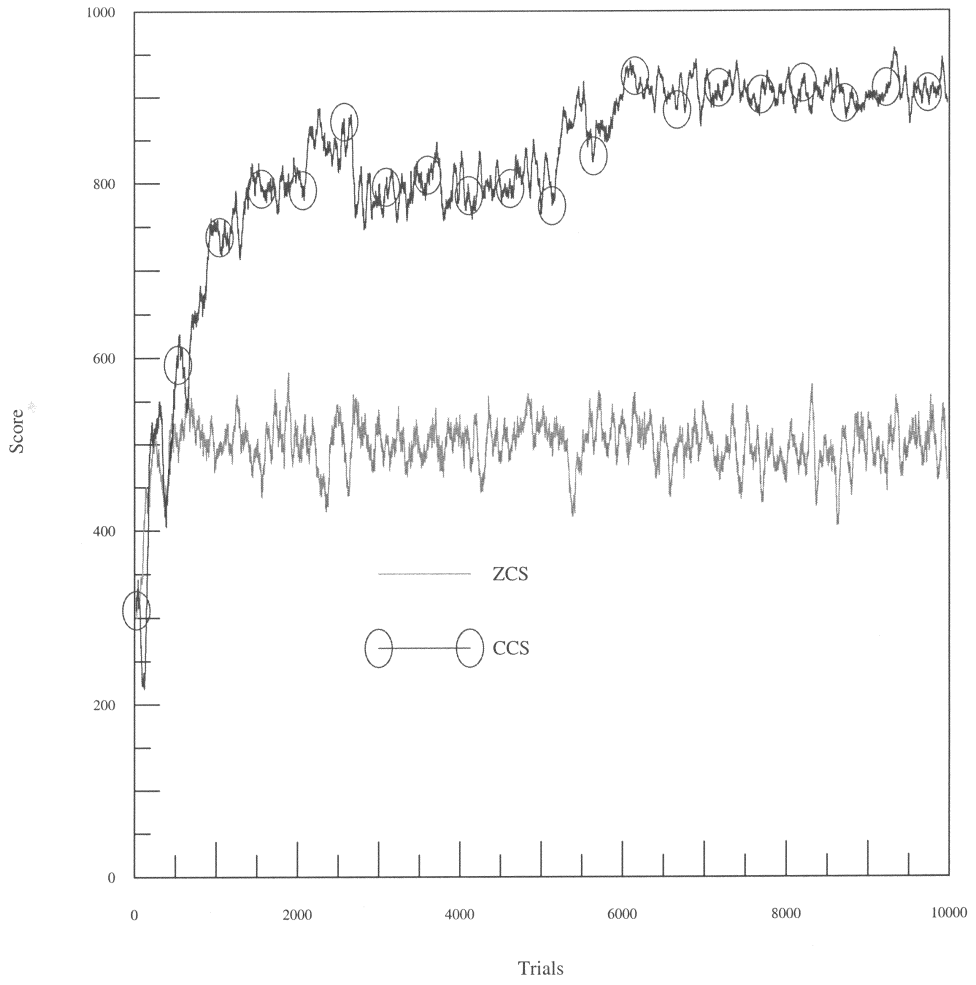

Figure 8. Performance in Task 2:2.

For the purposes of these tests the mazes are kept relatively simple. A is set to 2 for all tests so on each time step the system merely has to make a 1-bit binary decision. L is set to 3 and the systems are tested on tasks consisting of two and four mazes of depths of 2 and 3 (Figures 8, 9, 10). All parameters are set as in previous tests with a coupling rate of 0.25 and a base GA activation rate of also 0.25 for CCS. These graphs show the average scores of 10 runs for each system over 10,000 trials, again with a 50-point moving average filter applied to smooth the curves. In Figures 8–10, task 4:3, for example, represents a task consisting of four mazes of depth 3. In all of these maze tasks optimal performance is 1,000.

Again, the ZCS GA operates in the match set and the adaptive GA rate employed by CCS ensures that the rule-replacement rate remains a reasonably consistent parameter among the test systems.

It is clear that ZCS is not really equipped to tackle these problems. If M_z is set to 1 then ZCS can operate and is able to learn a single maze but with four mazes ZCS will be intentionally correct on average about one in four times at best (i.e., when the map

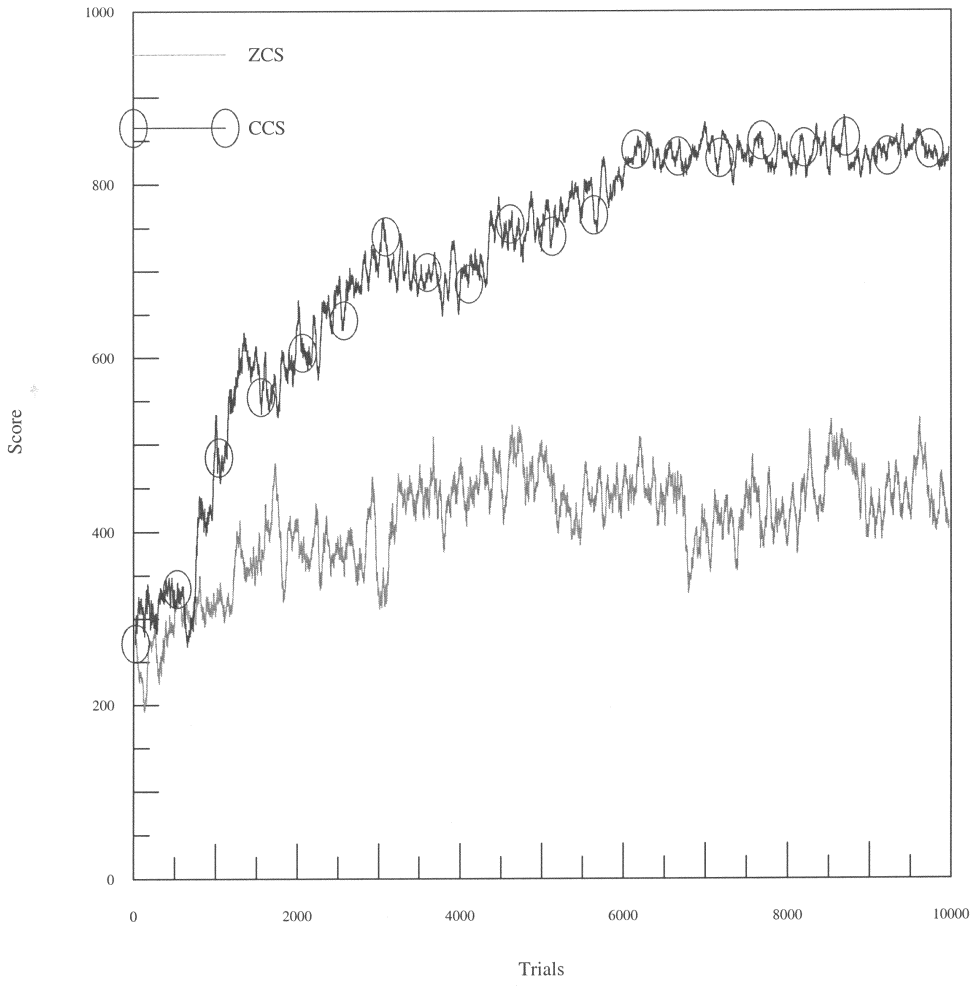

Figure 9. Performance in task 4:2.

Table 1. Two typical corporations.

	ID	Condition	Action	Link <-	Link ->
Corporation 3846	349	000	0	—	134
	134	0#1	0	349	—
Corporation 3931	202	#1#	1	—	328
	328	0#1	1	202	—

it has learned is presented). It can be seen that as N is increased the system becomes increasingly unable to locate the reward. At the end of a 10,000-trial run on task 2:2 (see Figure 7), the CCS rule base was examined. Two typically observed corporations are presented in Table 1.

Corporation 3846 responds to maze 1 (Figure 7). At time 0 rule 349 responds to the presented stimulus (000) and proposes action 0. If rule 349 wins the auction then at time 1 rule 134 will, as it matches the new stimulus (001), automatically be in control

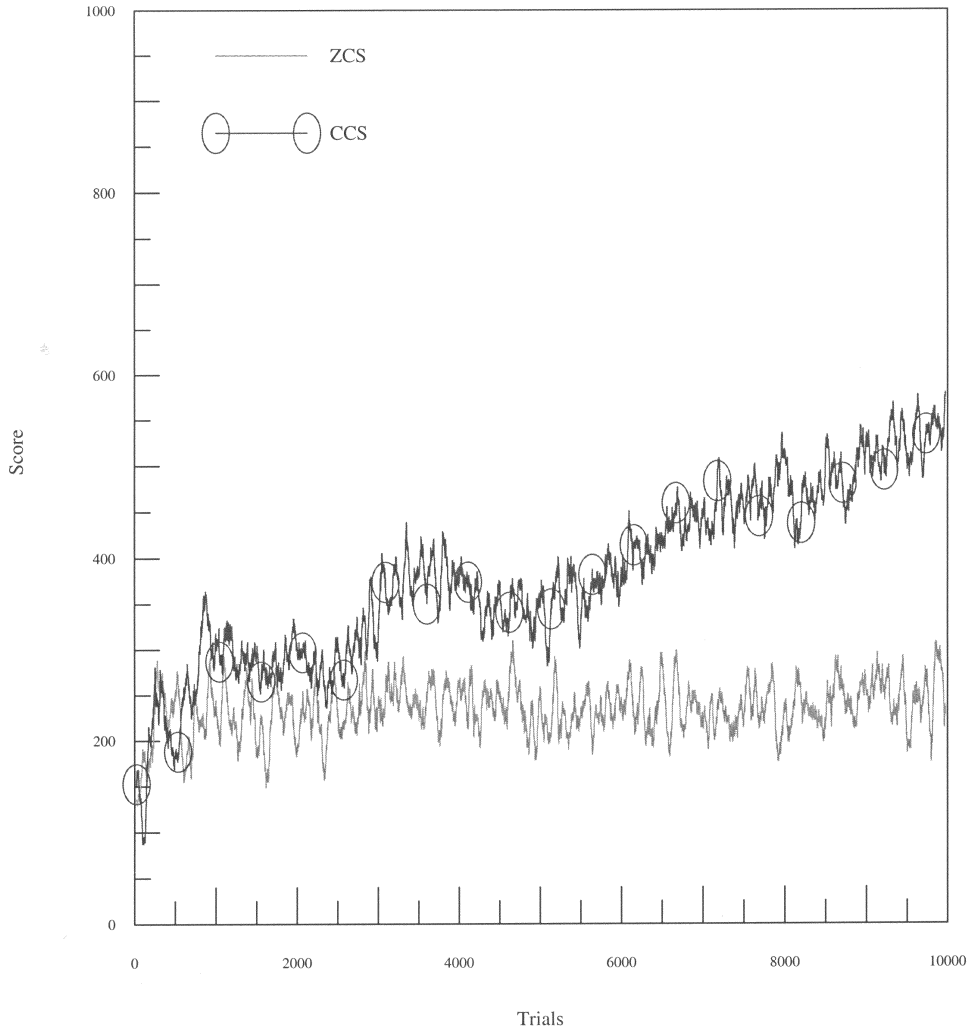

Figure 10. Performance in task 4:3.

of the production system. Rule 134 matches the stimulus and then proposes the correct action for maze 1 at time 1. This is a strong corporation and many copies of it can be seen in the rule base.

Corporation 3931 responds to maze 2 (Figure 7). Rule 202 matches the stimulus at time 0 and proposes the correct action. On the subsequent time step, rule 328 matches the new stimulus and again proposes the correct action. This is also a strong corporation, copies of which can be seen in the rule base. In CCS these two corporations alone are sufficient to solve task 2:2, whereas ZCS is unable to tackle the ambiguity present on time step 1.

The average number of corporations formed in CCS after 10,000 trials in task 2:2 is just under 200 and these are all of size 2. Virtually every rule in the population (size 400) has linked to a partner to form a two-member corporation. When N is increased to 3, corporations of size 2 form initially (about 80 by trial 300) and corporations of size 3 form more slowly (50 by trial 300). By trial 2,000 there are only 2 or 3 corporations of

size 2, but just under 130 of size 3. Most rules in the population (about 390 of 400 rules) therefore now belong to three-member corporations.

CCS offers a much improved performance on all tasks; however, the results indicate that the discovery components of both systems found their performance impeded as N increased. Although these mazes are quite simple the tasks are compounded in complexity firstly by the presentation of more than one maze and secondly by the high non-Markov properties of the longer mazes. In a sense these are "needle-in-a-haystack" tasks. There is only one correct path to a reward; any deviation leads to a payoff of 0. As the task length is increased the genetic search is made increasingly difficult. A "correct" corporation, that is, one of the same length as the task and advocating the optimal action on each time step, must be formed before any reliable positive feedback is received.

The above results show that adding the process of symbiogenesis to the simple ZCS can give improvements in performance in complex environments that contain ambiguous sensory inputs. We now briefly show that the same benefits are obtained when symbiogenesis is implemented in the more complex XCS [26] system.

5 XCS: A Learning Classifier System

The most significant difference between XCS and most other classifier systems (e.g., ZCS) is that rule fitness for the GA is not based on rule predictions (strengths) but on the accuracy of these predictions (see also [8]). This approach was adopted to encourage XCS to form efficient generalizations and a complete and accurate mapping of the search space X × A => P from inputs and actions to payoff predictions (rather than simply focusing on the higher payoff niches in the environment). XCS also uses a niche GA [2] rather than the typical panmictic scheme of ZCS. In XCS the classifier strength parameter is replaced by three new ones: prediction (p), prediction error (E) and fitness (F).

On each time step match sets are created as in ZCS (Section 2). A system prediction is then formed for each action in [M] according to a fitness-weighted average of the predictions of rules in [M]. The system action is then selected either deterministically or randomly (0.5 probability per trial). An action set is then made as in ZCS, giving the appropriate system output; a reward may or may not be received. If [M] is empty covering is used as in ZCS.

Reinforcement in XCS consists of updating the three parameters, p, E, and F for each appropriate rule. A rule fitness (F) is updated every time it belongs to $[A]_{-1}$. The fitness is updated according to the relative accuracy of the rule within the set in three steps:

1) Each rule's accuracy K_j is determined according to: $K_j = \exp[(\ln \alpha)(E_j - E_0)/E_0)]^*0.1$ This function falls off exponentially for $E_j > E_0$, that is, if a rule's error value is greater than some reference value (typically 0.01).
2) A relative accuracy K'_j is determined for each rule by dividing its accuracy by the total of the accuracies in the set.
3) The relative accuracy is used to adjust the classifier's fitness F_j using the moyenne adaptive modifée (MAM) procedure: If the fitness has been adjusted $1/\beta$ times, $F_j < -F_j + \beta(K'_j - F_j)$. Otherwise F_j is set to the average of the current and previous values of K'_j.

Next E_j is adjusted using P (see below) and the current value of p_j. The Widrow–Hoff technique is used as follows:

$$E_j = E_j + \beta(|P - p_j| - E_j).$$

Finally p_j is adjusted. The maximum $P(a_i)$ of the system's prediction array is discounted by a factor $\gamma (0 < \gamma < 1)$ and added to any external reward from the previous time step. This value is called P and is used to adjust the predictions of the rules in $[A]_{-1}$ using the Widrow–Hoff delta rule with learning rate $\beta (0 < \beta < 1)$:

$$p_j = p_j + \beta (P - p_j).$$

As noted above, the GA acts in match sets [M], that is, niches. Two rules are selected based on fitness. In this article we assume a fixed-size rule base N; macro-classifiers and varying N used by Wilson [26] are not incorporated here, that is, the system is initially seeded with a randomly generated fixed-size rule base and not primarily grown by the early activations of the cover mechanism. Rule replacement is based on the estimated size of each match set a rule participates in with the aim of balancing resources across niches. The GA is triggered (see also [3]) within a given match set if the number of time steps since its last invocation in that set passes a fixed threshold, based on the average time stamp of the rules. Typically this parameter is set to 25.

Wilson presents a generalization hypothesis [26] that states that XCS, with its accuracy-based fitness and a niche GA, could result in evolutionary pressure toward classifiers that would not only be accurate, but both accurate and maximally general. The default parameters presented for XCS, and unless otherwise stated for this article, are: $N = 800, E_0 = 0.01, \beta = 0.2, \gamma = 0.71, \chi = 0.8, \mu = 0.01, \phi = 0.5, p_0 = 10, \alpha = 0.1, P_\# = 0.5$. For the last 1,000 trials, deterministic action selection only is used (under which the GA is never active) to facilitate a full evaluation of the knowledge held in the rule base. The reader is referred to [26] for full details of XCS.

6 Symbiogenesis in XCS

6.1 Implementation

The linkage mechanisms of CCS are now implemented in a version of Wilson's XCS. During comparisons, the GAs of both XCS and the corporate system, CXCS, produce a single offspring rule or corporation on each invocation.

Rules within the system are given the same linkage components as those in CCS and linkage occurs again between rules from subsequent match sets at a fixed rate (typically a 10% probability on each step). Like the GA, linkage occurs only on exploratory cycles and so is also turned off for the last 1,000 trials of testing. In CCS, rule selection for linkage could be either random or probabilistic, or deterministic, based on the relative strengths of rules within the niches. The equivalent parameter to ZCS/CCS rule strength in XCS is the prediction parameter. In XCS it is the accuracy of the prediction that is used to evaluate rules, and it is not in keeping with XCS philosophy to base discovery decisions on the prediction parameter alone. Accuracy and fitness are also discounted as possible weightings for rule selection for linkage. In CXCS it is possible that rules that appear to be inaccurate when evaluated alone are precisely the rules that could benefit from rule linkage. If the inaccuracy is due to some sensory deception then the context of a corporate rule chain may limit a rule's activation to instances in which its action results in a more predictable consequence. In context the rule becomes more accurate. This is the main motivation for developing CXCS. With this in mind, selection for linkage in CXCS is determined randomly from rules (whose appropriate link is unattached) within the niche, imposing no bias based on the system's current perception of rule utilities. This seemed to be a reasonable policy for the prototype corporate XCS model. It is also possible that selection for linkage, within a niche, could be determined according to prediction with some benefit and this will certainly be investigated in future work.

As in CCS, corporations are reproduced and evaluated collectively. As such, rules within a corporation should share certain parameters used by the discovery component. These are fitness, which determines a rule's chance of selection for reproduction, and the estimate of mean match set size, which determines a rule's chance of being selected for replacement; two parameters are introduced, "corporate fitness" and "corporate niche size estimate." For single rules these parameters are identical to their existing fitness and match set size estimates. For linked rules, these values can be determined in a number of ways. Each rule could be given the average fitness and match set size estimate of all rules within the corporation. Alternatively, corporate fitness could be based on the lowest exhibited fitness within the corporation. In this way, a corporation is considered only as accurate or fit as its weakest link. This approach certainly offers the theoretical advantage of a bias against unwanted parasites within corporations. It is also possible to give each rule in the corporation a corporate niche size estimate equivalent to the smallest represented niche in the corporation. This policy considers that although one rule in a corporation may belong to a well-occupied niche or niches, the next rule may be the sole resident in another. In the initial design, corporate fitness for each rule in a corporation will be set to the lowest exhibited fitness within the corporation. Corporate niche size estimates will be determined as the mean match set size estimate within that corporate unit.

As in CCS, corporations can, while they continue to match presented stimuli, maintain persistent control of the performance component. In CXCS, corporations can take control during both exploration and exploitation cycles; however, in this respect functionality differs slightly between the two system modes. On each step, after action selection, during standard performance component cycles a rule is selected from the action set according to some policy and if this rule is corporate (i.e. it has an active link forward) then that corporation is given control of the system. During exploration cycles this rule is selected randomly from [A] and during exploitation cycles the rule is selected deterministically according to rule bids. Again, as in CCS, *followers* (rules with an active "link-back" component) are given only limited access to [M].

When comparing systems that introduce different numbers of offspring per invocation of the GA it is important to consider the differences in relative rule replacement rates. Without such consideration it is possible to generate quite misleading comparisons of systems as rule replacement concerns tend to be among the more fragile aspects of classifier system design [e.g., 23]. To counteract this, a variable element is introduced into the CXCS GA activation, as before. The system records the number of rules reproduced on each invocation of the GA (i.e., the size of offspring corporation, S_c). When the existent activation policy indicates that the GA should fire, a further mechanism will only allow the GA to fire with probability set according to the reciprocal of the mean offspring size parameter, S_m (initialized to 1). This estimate is adjusted on each invocation of the GA according to the standard Widrow–Hoff delta rule [26] with the learning rate parameter β (typically 0.2), that is, $S_m < -S_m + \beta(S_c - S_m)$. This modification to the GA activation mechanism ensures at least a more consistent rate of rule replacement throughout testing; however, the drawback is that a corporate system, compared to a standard system, will incur a relative reduction in crossover events. The more significant factor is perhaps the rule replacement rate and its effect on convergence within the rule base, and so here, the variable GA activation policy is adopted for all tests.

An implementation of CXCS, as described above, is now compared to XCS in the series of delayed-reward tasks used above, which can only be solved by the formation of internal associations between rules.

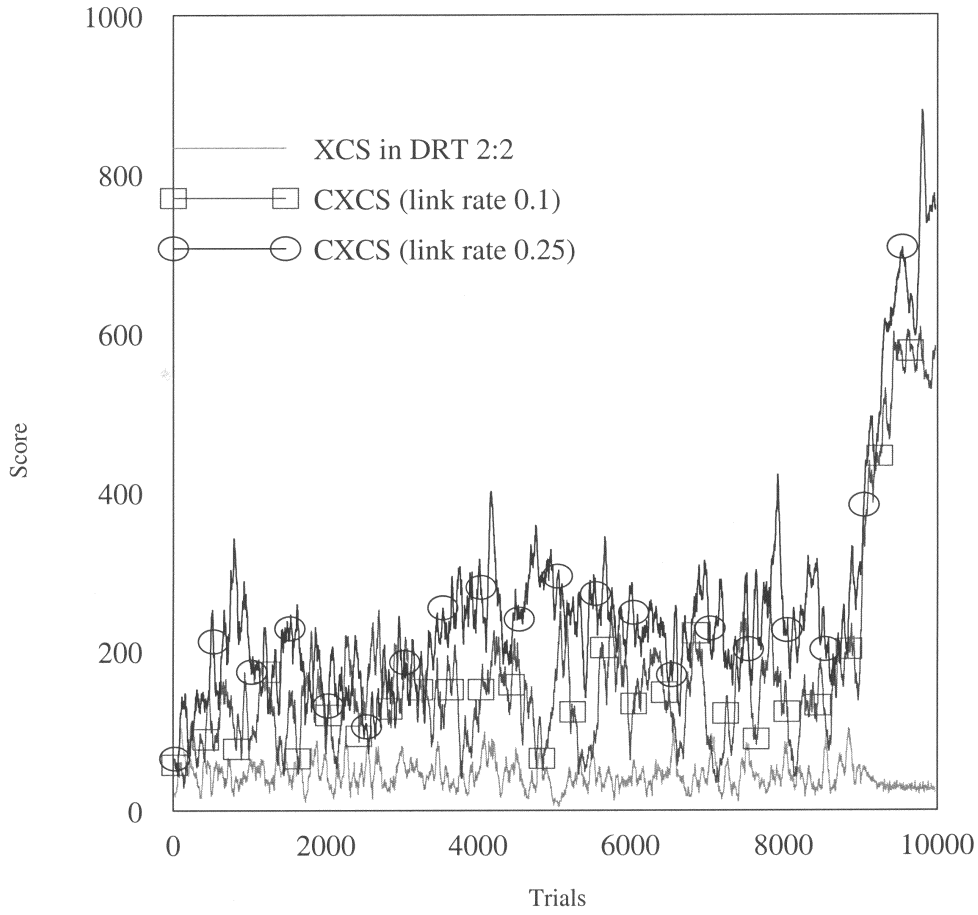

Figure 11. Performance in delayed reward task (DRT) 2:2.

6.2 Results

Tests are now conducted in the previously presented series of three delayed-reward tasks (DRTs). On each time step the system must choose one of two actions (0 or 1). One path in each maze will yield a reward of 1,000; all others return a reward of 0. The mazes are set up so that if the system selects the same action on each step through the maze it will be guaranteed a reward of 0. This precaution ensures that the successful solution of the mazes is not achievable by a single completely general rule and will in fact require some form of cooperative behavior within the rule base. Tests consist of a series of 10,000 trials and all curves are again averages of 10 runs. The plots (Figures 11–13) represent the average score over the last 50 trials. Figures 11–13 include plots of CXCS with linkage rate set to 0.1 and also 0.25. The plots reveal that CXCS performance improves significantly during the last 1,000 exploitation trials. This indicates that CXCS may benefit from a modification to the action selection policy. This possibility is considered further in Section 6.3.

According to the XCS action selection strategy, on a fixed proportion of trials selection will be random from all advocated actions. On such an exploratory trial all rules are likely to receive variations in profit according to the different contexts in which they fire (even if all rules are 100% specific). This will clearly result in low perceived accuracy

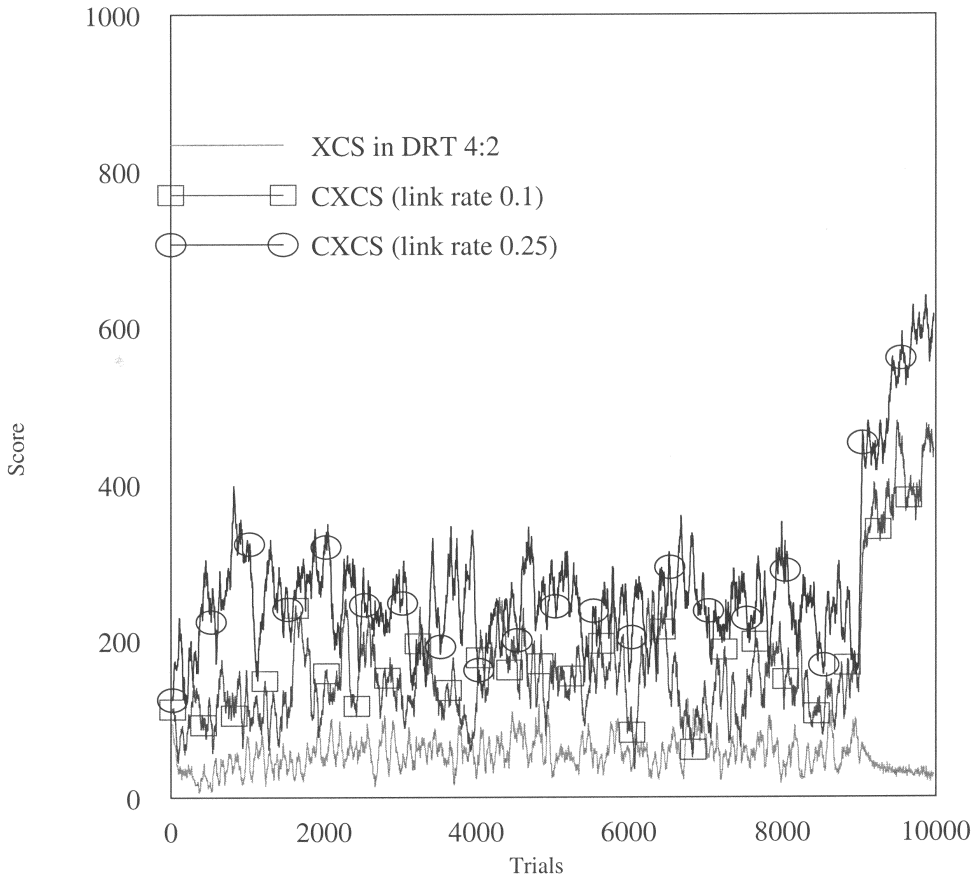

Figure 12. Performance in DRT 4:2.

for all firing rules. XCS accuracy is determined by a quite severe function with a sharp cutoff beyond the acceptable error margin. In such mazes it is possible that all firing rules on any time step will exhibit a perceived accuracy of 0, leading to each rule having a resultant relative accuracy, and thus a fitness based on the reciprocal of the niche size. Although, during an exploitation cycle, rules representing the optimal system action may be present in the niche, with the bid based on the prediction scaled according to fitness, there will be no discernible bias toward the higher reward-yielding action and so the system has certain difficulties mapping the maze.

The effects of the above problem are illustrated by the state of the rule base at the end of testing and also by the continual activation of the cover operator, especially during the last 1,000 exploitation trials. On a 2:2 task for instance, the rule base will contain many rules of varying specificity that match the "second step stimulus" (001). All of these with the exception of the fully general ### rules have a fitness value of 0. The prevalent ### rules tend to increasingly occupy both niches and thus their prediction values will slowly fall to 0 (due to the previously mentioned precautions taken with the reward scheme designs), and over this period their fitness values will gradually rise to maximum fitness, 1,000 (or 1, as it were). Eventually [M] prediction on the second step falls below the threshold parameter value and the cover operator is invoked. This process continues throughout the test period.

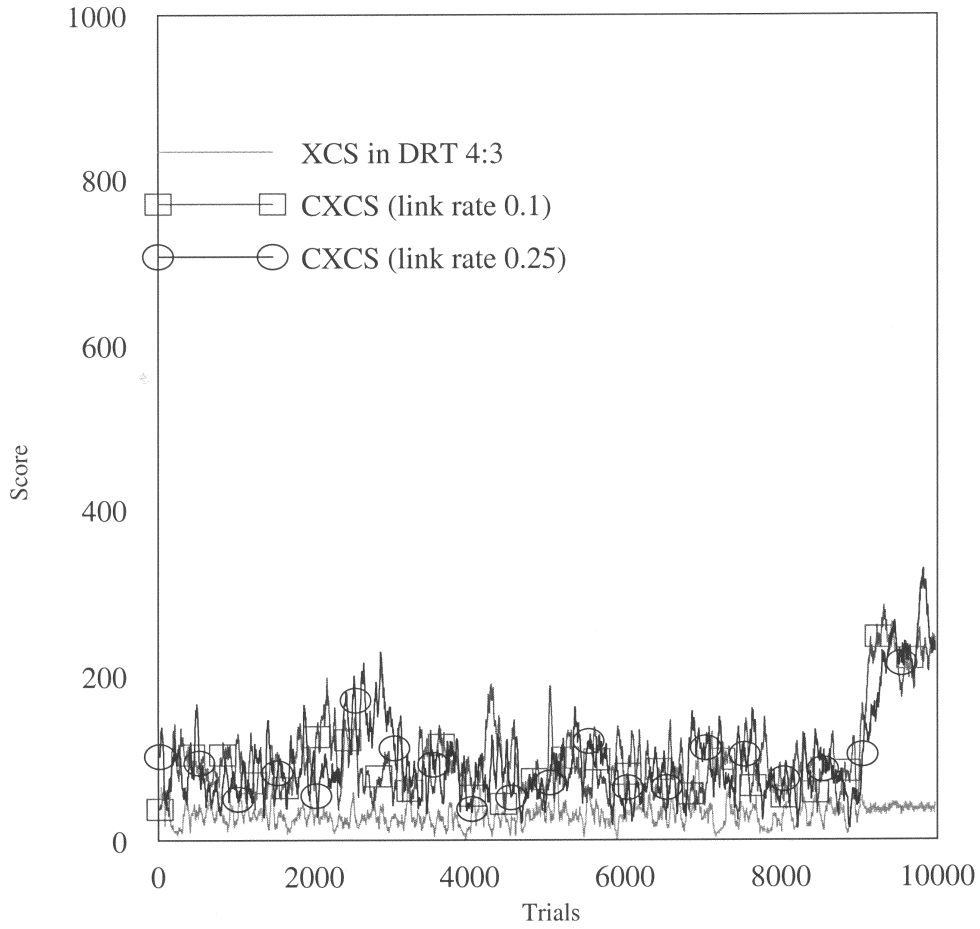

Figure 13. Performance in DRT 4:3.

CXCS does exhibit some ability to map these "internal association" tasks; however, results are somewhat disappointing and certainly do not compare to the equivalent results produced by running the CCS model in the same tasks (see Section 4). In a 2:2 task, for example, the resultant rule base at the end of testing will typically contain about 100 corporations. These will be of indeterminate length and there will be a number of excessively long corporations. In the equivalent CCS test, the resultant rule base would typically contain approximately 200 corporations, of which just about all will be of length 2. In other words, the entire population had linked to form corporations of the appropriate length to tackle the presented task.

Finally, in the next section, we show that the CXCS mechanisms can be modified to provide significant performance improvements in the more complex delayed reward tasks.

6.3 A Modification to CXCS: Look Ahead

Wilson [26] suggests as a future avenue of research that the concept of fitness based on accuracy of prediction could be extended to classifiers with expectons. An expecton

takes the same form as a rule *condition* component but is a prediction of the resulting sensation or stimulus. The idea is based on previous work by Riolo [20] and Holland [11] and has recently been used by Stolzmann [22]. Such a system does not match stimulus action pairs with prediction values, but with some anticipated stimulus. Fitness for such rules may not be based only on accuracy of prediction but may also, or separately, represent the accuracy of the expecton in predicting the next sensation. This principle can be adapted for use in CXCS.

Although rules in CXCS do not have an expecton, if their "link forward" is active, they are still associated with information about some anticipated stimulus. This is in the form of the condition of the rule to which they are linked. Actually such a rule may be associated with information regarding any number of anticipated future stimuli, dependent on the length of the corporation.

With this in mind, an alternative measure of fitness may be employed in CXCS for rules with active forward links, based not only on the accuracy of their prediction but also scaled according to the ratio of "control hits" (C_b) to the total number of times that control is received (C_c). If a rule takes control of the performance component and on the next time step the following rule in the corporation matches the subsequent stimulus and therefore inherits control of the system successfully, then this is considered to be a "control hit." If the rule fails to match the presented stimulus, then this is a "control miss." As such, this ratio acts as a measure of how accurately the corporation is representative of some perceived aspect of the test environment. C_b and C_c are updated each time the rule takes control of the system (but on formation of $[\mathrm{M}]_{+1}$). The control ratio parameter is adjusted as follows: $C_r = C_b/C_c$. This is determined prior to fitness calculations. On fitness adjustments, the relative accuracy parameter, K', is scaled according to the control ratio before the fitness parameter is updated in the usual manner. As previously, corporate fitness is consistent for all rules in the corporation and is based on the lowest member fitness in the corporation. Corporate niche size estimate is again set to the mean match set size estimate of member rules.

Figures 14–16 show performance plots of this modified version of CXCS in the same series of three delayed-reward tasks as used previously. Rule linkage rate is again set to 0.25 and all other parameters are as in the previous section. Results show that the new fitness evaluation produces improved results in the more difficult delayed-reward tasks. At the end of testing the system generally contains about 100 corporations, most of which are of the appropriate length for the particular task (i.e., of length 2 or 3). CXCS final performance in these tasks matches that of CCS.

7 Conclusions

Symbiogenesis has been the major factor in at least one of the most important steps in evolutionary history—the evolution of eukaryotic cells [17]. Consequently the evolutionary significance of symbioses in general is becoming accepted [19]. In this article we have examined the effects of incorporating the process of symbiogenesis into a machine learning architecture that maintains an ecology of interacting rules to solve a given task and uses evolutionary computing techniques to refine those rules. The aim here is to improve the performance of the LCS architecture in complex problem domains, that is, to view symbiogenesis as a universally beneficial adaptation process, regardless of the medium in which it is instantiated.

Section 3 showed that adding rule linkage to a simple LCS does not give noticeable benefit. In Section 4 we added the related aspects of spatial location and membrane formation to the system. Here temporal closeness and execution persistence become the respective analogs and were shown to improve performance greatly as the diffi-

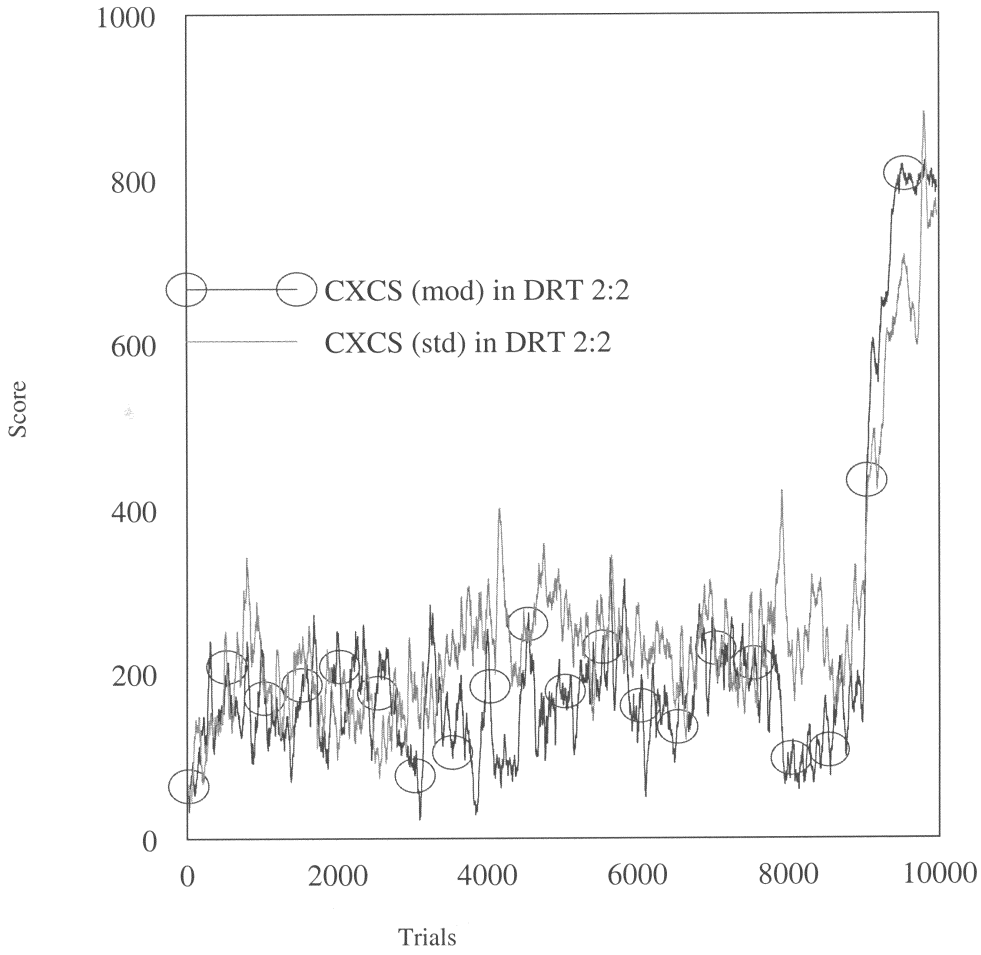

Figure 14. Performance in DRT 2:2.

culty of the tasks faced increased; rule corporations were shown to bestow memory upon the system, allowing for sensory ambiguities. Section 6 showed that the same general mechanisms could be used to similar effect within another more complex LCS framework—XCS.

We are now extending this work in a number of ways: Maynard Smith and Szathmary [19] note that, along with symbiogenesis, the process of genome duplication, with subsequent functional divergence, has been significant in the major transitions of evolution and this is now being incorporated into our system (see also [16]); further, they note that the emergence of multi-entity structures from a single reproductive entity, as occurs in multi-cellularity, for example, was another significant step in the evolution of complexity and ways to incorporate these kinds of structures are also being examined (after [5]); and finally, we are currently applying the architecture to the real-world problem domain of road traffic junction controller design (EPSRC grant no. ER/R06748).

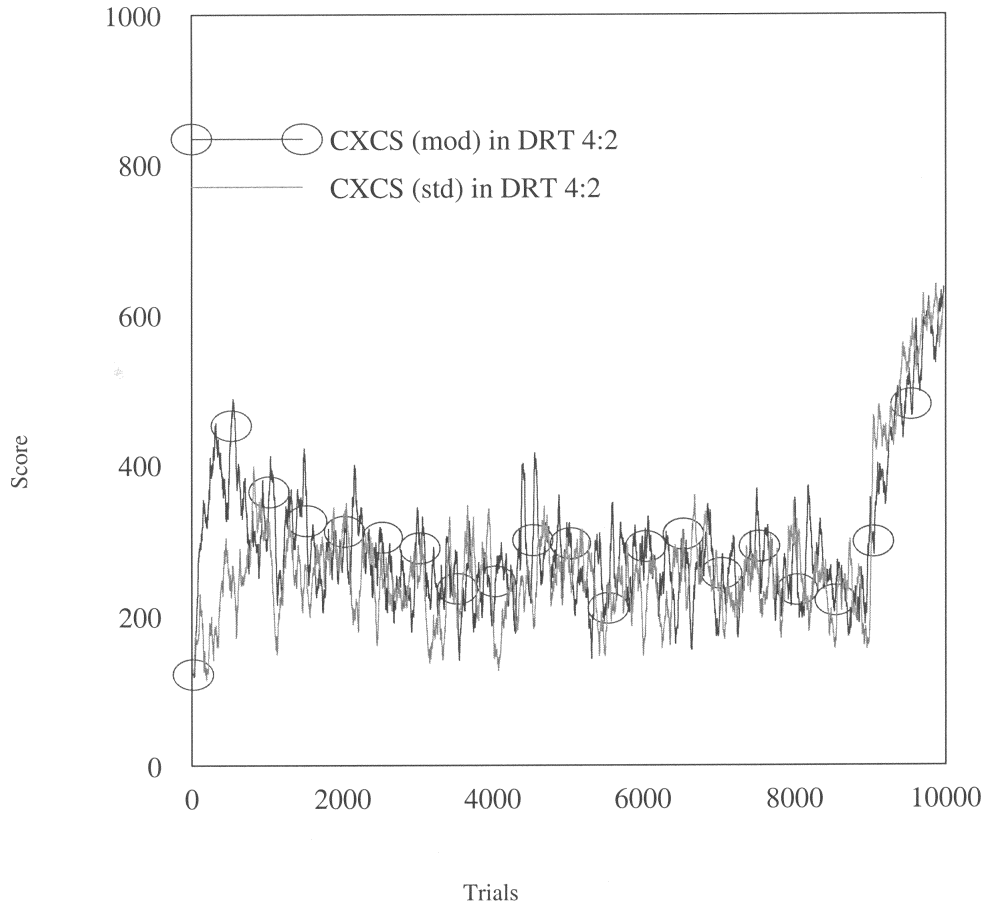

Figure 15. Performance in DRT 4:2.

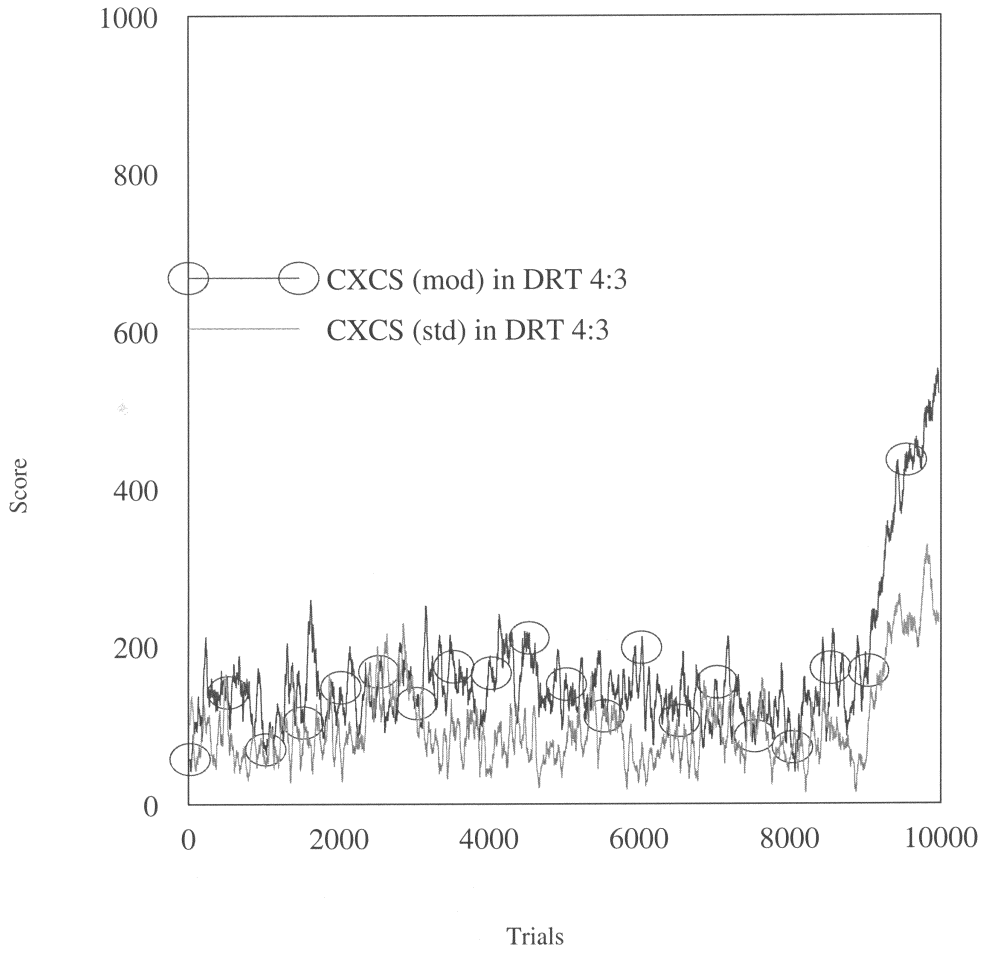

Figure 16. Performance in DRT 4:3.

References

1. Axlerod, R. (1987). The evolution of strategies in the iterated prisoner's dilemma. In L. Davies (Ed.), *Genetic algorithms and simulated annealing* (pp. 32–41). San Mateo, CA: Morgan Kaufmann.
2. Booker, L. B. (1985). Improving the performance of genetic algorithms in classifier systems. In J. J. Grefenstette (Ed.), *Proceedings of the First International Conference on Genetic Algorithms and Their Applications* (pp. 80–92). Hillsdale, NJ: Erlbaum.
3. Booker, L. B. (1989). Triggered rule discovery in classifier systems. In J. D. Schaffer (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms* (pp. 265–274). San Mateo, CA: Morgan Kaufmann.
4. Bull, L. (1997). Evolutionary computing in multi-agent environments: Partners. In T. Back (Ed.), *Proceedings of the Seventh International Conference on Genetic Algorithms* (pp. 370–377). San Mateo, CA: Morgan Kaufmann.
5. Bull, L. (1999). On the evolution of multicellularity and eusociality. *Artificial Life, 5*, 1–15.
6. Bull, L., & Fogarty, T. C. (1996). Artificial symbiogenesis. *Artificial Life, 2*, 269–292.
7. Cobb, H. G., & Grefenstette, J. J. (1991). Learning the persistence of actions in reactive control rules. In *Proceedings of the Eighth International Machine Learning Workshop* (pp. 293–297). San Mateo, CA: Morgan Kaufmann.
8. Frey, P. W., & Slate, D. J. (1991). Letter recognition using Holland-style adaptive classifiers. *Machine Learning, 6*, 161–182.
9. Grefenstette, J. J. (1987). Multilevel credit assignment in a genetic learning system. In J. J. Grefenstette (Ed.), *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms* (pp. 202–209). Hillsdale, NJ: Erlbaum.
10. Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor: University of Michigan Press.
11. Holland, J. H. (1990). Concerning the emergence of tag-mediated lookahead in classifier systems. In S. Forrest (Ed.), *Emergent computation* (pp. 188–201). Cambridge, MA: MIT Press.
12. Holland, J. H. (1995). *Hidden order*. Reading, MA: Addison-Wesley.
13. Holland, J. H., & Reitman, J. S. (1978). Cognitive systems based on adaptive algorithms. In D. A. Waterman, & F. Hayes Roth (Eds.), *Pattern-directed inference systems* (pp. 313–329). New York: Academic Press.
14. Ikegami, T., & Kaneko, K. (1990). Genetic fusion. *Physical Review Letters, 65*, 3352–3355.
15. Khakhina, L. N. (Ed.). (1992). *Concepts of symbiogenesis: History of symbiogenesis as an evolutionary mechanism*. New Haven, CT: Yale University Press.
16. Lindgren, K., & Nordhal, M. G. (1994). Cooperation and community structure in artificial ecosystems. *Artificial Life, 1*(1), 15–38.
17. Margulis, L. (1970). *Origin of eukaryotic cells*. New Haven, CT: Yale University Press.
18. Maynard Smith, J., & Szathmary, E. (1993). The origin of chromosomes 1: Selection for linkage. *Theoretical Biology, 164*, 437–466.
19. Maynard Smith, J., & Szathmary, E. (1995). *The major transitions in evolution*. Oxford: Freeman.
20. Riolo, R. (1990). Lookahead planning and latent learning in a classifier system. In J. Meyer & S. W. Wilson (Eds.), *From animals to animats* (pp. 316–326). Cambridge, MA: MIT Press.
21. Smith, R. E. (1994). Memory exploitation in learning classifier systems. *Evolutionary Computation, 2*, 199–220.
22. Stolzmann, W. (2000). An introduction to anticipatory classifier systems. In P. L. Lanzi, W. Stolzmann, & S. W. Wilson (Eds.), *Learning classifier systems—from foundations to applications* (pp. 175–194). Berlin: Springer.

23. Tomlinson, A., & Bull, L. (1999). A zeroth-level corporate classifier system. In A. S. Wu (Ed.), *Proceedings of the 1999 Genetic and Evolutionary Computation Conference Workshop Program* (pp. 306–313). Orlando, FL: Gecco.

24. Wilson, S. W. (1985). Knowledge growth in an artificial animal. In J. J. Grefenstette (Ed.), *Proceedings of an International Conference on Genetic Algorithms and their Applications* (pp. 16–23). Hillsdale, NJ: Erlbaum.

25. Wilson, S. W. (1994). ZCS: A zeroth level classifier system. *Evolutionary Computation, 2*, 1–18.

26. Wilson, S. W. (1995). Classifier fitness based on accuracy. *Evolutionary Computation, 3*, 149–176.

27. Wilson, S. W. (1998). Generalization in the XCS classifier system. In J. R. Koza, W. Banzhaf, K. Chellapilla, K. Deb, M. Dorigo, D. B. Fogel, M. H. Garzon, D. E. Goldberg, H. Iba, & R. L. Riolo (Eds.), *Proceedings of the Third Annual Genetic Programming Conference* (pp. 665–674). San Mateo, CA: Morgan Kaufmann.

28. Wilson, S. W., & Goldberg, D. E. (1989). A critical review of classifier systems. In J. D. Schaffer (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms* (pp. 244–255). San Mateo, CA: Morgan Kaufmann.