could be added to ZCS. Wilson's proposal involved adding an internal memory register consisting of a few binary digits (bits), and augmenting the system's set of possible actions to include actions which set or reset individual bits in the register. Over time, ZCS should adapt to manipulate and exploit internal state in situations where this is appropriate.

In this paper we present results from experiments with ZCS extended to incorporate Wilson's proposed memory system: we refer to the extended system as ZCSM. Our results demonstrate that Wilson's proposals do work, at least for small ($b < 3$) b-bit internal registers.

For completeness, Section 2.1 gives a summary of ZCS, followed by our replication of Wilson's published results in Section 2.2. Section 3 presents our results from extending ZCS to incorporate temporary memory. Results from testing ZCSM in a variety of environments indicated that occasionally an adapted population of classifiers could exhibit severe failures in performance which were both sudden and unpredictable. To explore the cause of these failures, Section 4 describes results from extended testing of ZCS in situations where increasingly long chains of sequential actions are required to reach the reward state: we found that ZCS grew less stable as the chain-length increased. We argue that this is due to two factors: greedy classifier creation and conflicting over-general classifiers. The second factor could be alleviated by basing the fitness of classifiers on their accuracy.

To demonstrate that the effects we identify are not artifacts of the spatially discrete 'grid-world' environments employed in our experiments, Section 5 presents results from using ZCS for animat guidance in environments where the animat's spatial location is continuous and its sensory input is more firmly analogous to visual sensing: we show that similar stability problems occur. Conclusions are drawn in Section 6.

2 ZCS: Overview and Replication

2.1 Overview

The definitive paper on ZCS is (Wilson, 1994b), to which we refer the reader for full details. For the sake of completeness, we include an overview here.

ZCS is viewed as a mechanism which interacts with some environment: detectors supply a binary encoded *sense-vector* which affects the *action* chosen by ZCS: these actions are executed by *effectors* which may alter the state of the environment in some manner.

ZCS maintains a single population, [P], of N condition-action classifiers. The conditions are encoded using the ternary alphabet $\{0, 1, \#\}$ where 0 and 1 are used to match against sensory input, and # acts as a "don't-care" wild-card, allowing generalization. Actions are also encoded using a discrete alphabet. The conditions and actions in the initial population are set randomly: the probability distribution of characters c (i.e. 'alleles') at each locus in the condition is given by: $\Pr(c = \text{'\#'}) = P_{\#}; \Pr(c = \text{'0'}) = \Pr(c = \text{'1'}) = 0.5(1.0 - P_{\#})$. Each classifier has an associated scalar strength, set initially to a value S_0 for all classifiers.

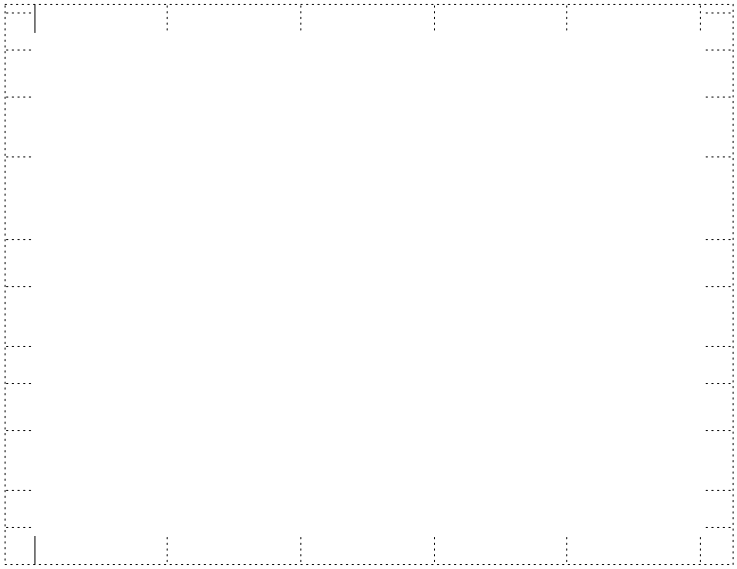
ZCS operates iteratively in discrete time-steps. On each time-step, the system passes through stages of *performance*, *reinforcement*, and *discovery*.

In the performance stage, the current sensory input is compared with the conditions of all classifiers in [P]: a condition matches the input if there is a 1 in the input at each locus where there is a 1 in the condition, and likewise for 0's; a wild-card # in the classifier condition matches either a 0 or a 1 in the input. All

2.2 Replication of Wilson's Results

Wilson (1994b) gives results of ZCS operating in 'woods'-style environments (cf. Wilson, 1985; Cliff & Bullock, 1993). In both, the classifier system is considered as being responsible for guiding an 'animat' (i.e. an artificial creature: (Wilson, 1985, 1987, 1991)) through environments composed of two-dimensional grids of cells. Cells may be occupied by an obstacle (represented as T for 'tree'), by a reward (represented as F for 'food'), or may be blank. The animat can occupy blank cells, and can move into F reward-cells but not into T obstacle-cells. The animat moves in discrete steps: on each step it may move into one of the eight surrounding cells. In all the environments discussed in this paper, toroidal wrap-around occurs if the animat attempts to move off an edge. At each cell, the animat 'senses' the contents of the eight surrounding cells, encoding the contents of each using a 2-bit binary code (i.e.: 11=F; 10=T; 00=blank) to form a 16-bit *sense-vector* against which classifier conditions can be matched.

A single k85, de



Ⓣ

Ⓣ

opportunity to "sit and think"). Wilson describes this extension as the addition of "temporary memory": for consistency, we will use the same terminology, although we acknowledge (and agree with) the comments of Colombetti and Dorigo (1994, pp.250-251), who note that terms such as *memory* and *representation* are often deceptively subjective labels referring to the agent's internal state, and that internal state is a more neutral term.

We refer to ZCS with b bits of memory as ZCSMb: when discussion of ZCSMb is independent of the value of b, we refer simply to ZCSM.

3.2 Implementation

ZCSM was implemented in the manner proposed by Wilson. The classifier condition is divided into two sections: the first is used to match against the sense-vector, as in ZCS; the second is a sequence of b characters from the ternary alphabet $\{0, 1, \#\}$ which is matched against the current settings of the b bits in the memory register. The classifier action is also in two parts: the first part specifies one of the 9 possible external actions (eight movements and one null), encoded as two characters from $\{0, 1, \#\}$. The second part is the internal action, being a sequence of b characters from the same ternary alphabet: a 0 or 1 in the internal action specifies that the corresponding bit in the register should be set to that value; a # indicates that the corresponding bit of the register should be left unaltered.

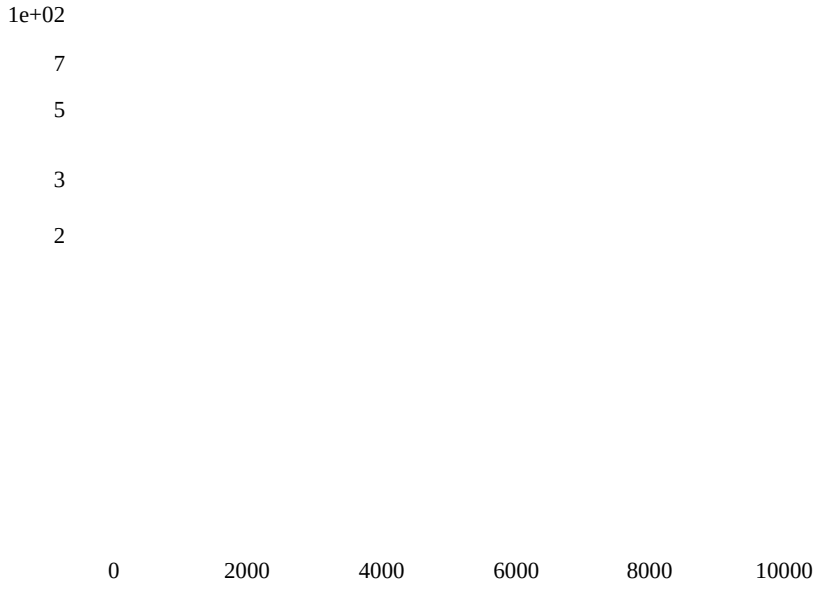
In the formation of the initial random population of classifiers in ZCSM, characters for both conditions and both actions are generated at random, using the same probability distribution as for the generation of the sensory condition in ZCS. At the start of each trial, the register is initialized by setting all bits to zero. In breeding, crossover can occur at any point in the string formed by concatenating the characters representing the sense-condition, memory-condition, internal-action, and external-action; in that order.[4]

Adding the internal register does not require the introduction of any new parameters. In the experiments reported below, all parameter values were the same as those used in the ZCS replication experiments described previously in Section 2.2.

3.3 Results

Although internal state can be an advantage in a variety of situations, our primary interest was to test the ability of ZCSM in dealing with non-Markov woods-style environments. Depending on the structure of the environment, ZCSM should be able to employ the memory mechanisms to disambiguate perceptually aliased global states. That is: if a given sense vector can correspond to one of several global positions with conflicting appropriate actions, it is possible in certain environments for the internal state to be manipulated in such a way that the memory register augments the (aliased) sensory input so that an appropriate action can be selected. This is well illustrated by considering the environment woods101 shown in Figure 6.[5]

There are two distinct cells in woods101 which generate the same sense vector (see Figure 7), but


1e+02
7
5
3
2
0
2000
4000
6000
8000
10000

Starting Cell	Most Likely Trajectory
0_0	F
1_0	0_0 F
2_0	6_0 2_1 0_1 F
3_0	0_0 F
4_0	6_0 2_1 0_1 F
5_0	3_0 0_0 F
6_0	2_1 0_1 F
7_0	5_1 3_0 0_0 F
8_0	6_0 2_1 0_1 F
9_0	7_0 5_1 3_0 0_0 F

Figure 10: Most likely trajectories resulting from the vector field shown in Figure 9. Each line in the table shows a starting cell, followed by the most likely trajectory from that cell through woods101 to the F. The cell labels are those introduced in Figure 6. The subscript to each label shows the contents of the memory register. Note that the memory register is set to 0 at the start of each trial.

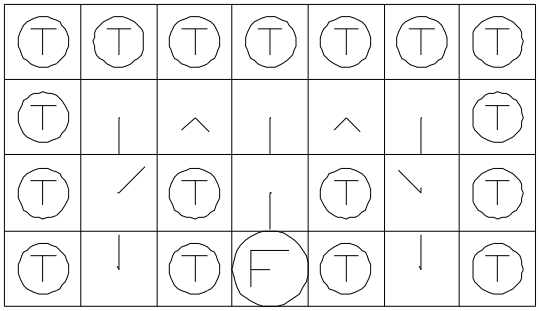
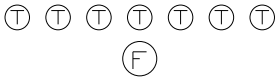
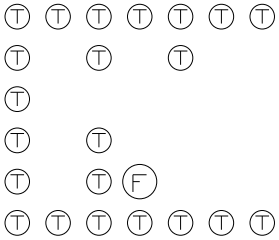
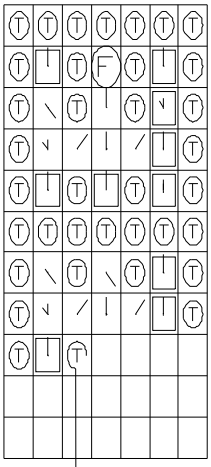


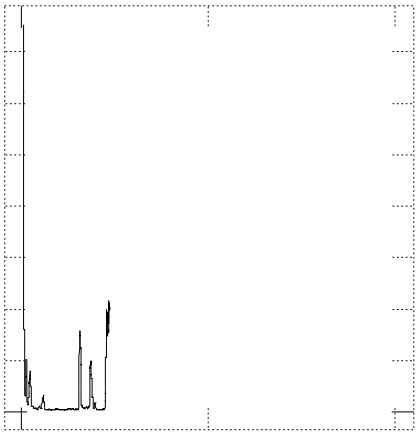
Figure 12: Vector field from ZCSM0 in woods101, after 10000 trials.





the trial started: the circuitous routes ensure that the memory bit is set in such a way that when the aliased cell is reached, an appropriate action is generated. As can also be seen from the trajectories in Figure 18, the most likely trajectories from the aliased cells on the vertical





continue to adapt in light of changes in the environment). Furthermore, if the environment is non-Markov, then the given sense-vector may be an alias for more than one global state, and the weaker action sets may indicate actions which occasionally lead to a reward. As we saw in Section 3.3, insufficiently large internal state-space could lead to the formation of action sets which correspond to stochastic solutions to the problem of aliased inputs, with the relative distribution of strengths over actions for that state being an indication of the probability of reward following from each action.

We use the phrase *Deceptively Lucky Exploration* to refer to the case where a low-strength action is selected for exploration, and circumstances just happen to be

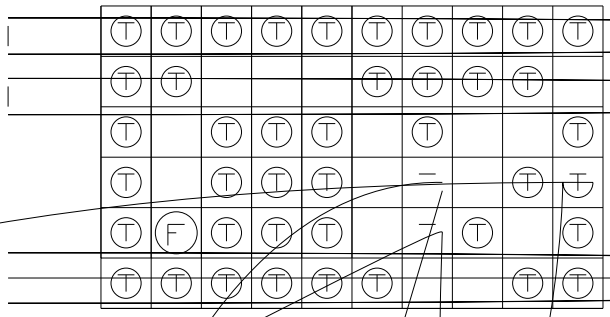


Figure 23: The environment woods14-14.

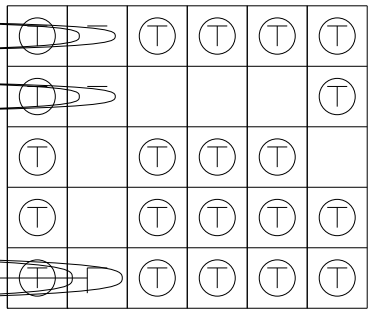


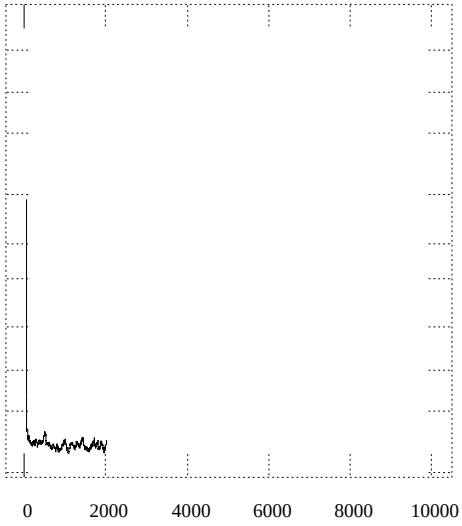
Figure 24: The environment woods14-06.

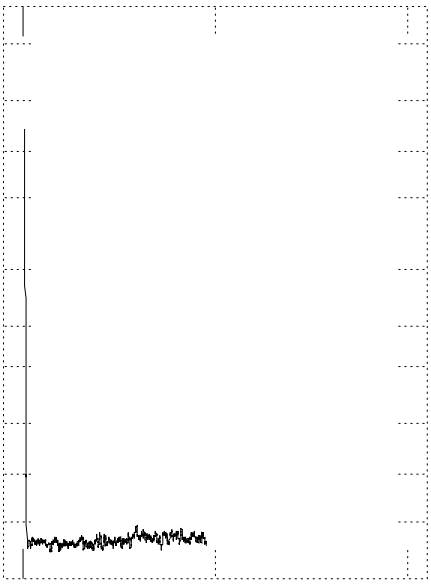
As can be seen from Figure 22, woods14 has 18 blank cells, and at each cell there is a unique
sense-vector, and *only one* action which takes the animat nearer to the
one action in each cell, deceptively lucky exploration cannot occur.

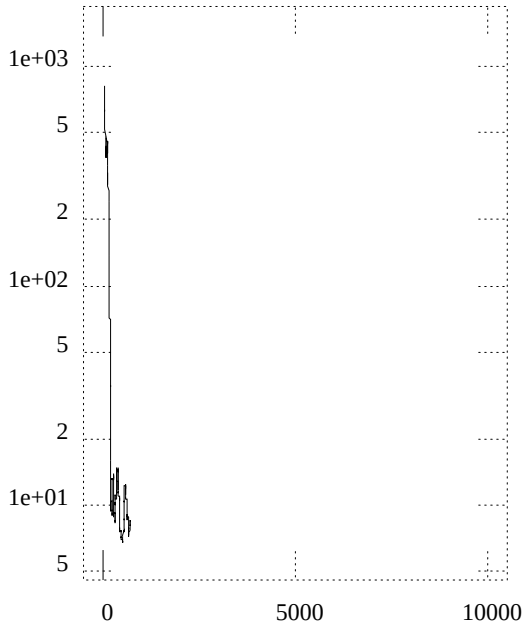
Given its simple nature, we expected that woods14 would represent
problem. We planned to first generate some control data using ZCS o
test ZCSM on the same environment to see whether the addition of m
caused the problem: because woods14 is Markov, ZCS should be capable
classifiers, so the comparison between ZCS and ZCSM would be fair.

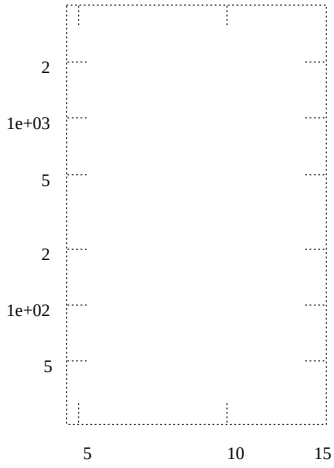
Our intention in working with woods14 was purely to identify whe
major factor underlying the observed instability. For this reason, we did
parameters from the values used in the work reported in Sections 2.2
we found that even ZCS could not learn reliable strategies for woods14
lengths required in this environment are sufficiently long that stable set
be found.

Problems with long chain maintenance have been noted before


0
2000
4000
6000
8000
10000




1e+03
5
2
1e+02
5
2
1e+01
5
0
5000
10000


2
1e+03
5
2
1e+02
5
5
10
15

received. In the case of an action set which moves the animat onto a F, $s = 1$ so the reward received is modulated by a factor of 1.0. For an action set active at cell c

external actions: for the experiments described here, this is 0.125 so the probability of randomly choosing an inappropriate action is 0.875. If an inappropriate action is (randomly) chosen for the classifier created in covering, then the system has to repeatedly perform that action until its strength is sufficiently diminished (by repeated decrements during reinforcement) that covering is re-activated. It is not inconceivable that this process (of covering generating a sole inappropriate action which requires many steps to be 'deselected') could be repeated several times in succession within a particular trial, leading to a very large number of steps to F on that trial.

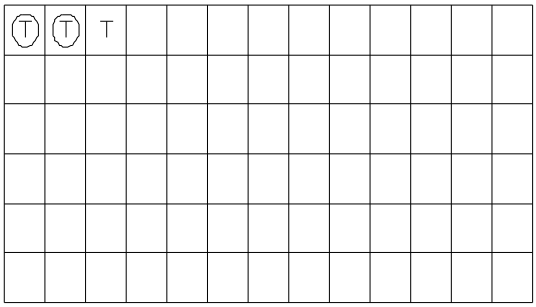
However, once the F has been reached on such a trial in woods14-p, covering *must* have generated the correct action, and so on subsequent trials the problem should be much less likely to occur. Therefore, while greedy classifier generation clearly can account for sudden spontaneous decreases of performance (i.e. one 'bad' trial, taking a large number of steps), it is less clear why the breakdowns in performance are *sustained* over large sequences of trials (because, for the 'bad' trial to have ended, ZCS must have generated classifiers advocating the correct action in *each* cell in the woods14-p environment, so the next trial should not present any problem).

Although it is possible that 'chain reactions' (where inappropriate covering in one distant cell could trigger deletions of weak classifiers at other distant cells, giving rise to further inappropriate covering which causes further deletion of crucial classifiers, and so on) could be the reason for the sustained failures, there is another factor we have identified which appears to play a much stronger role in causing and sustaining the observed failures: this factor is the formation of conflicting over-general classifiers, discussed further in Section 4.2.2. It should be noted that Wilson (1994b, pp.13-15) anticipated the problem of greedy classifier creation in long-chain maintenance, and proposed an extension to ZCS involving a *niche* GA (cf. Booker, 1989) as one potential remedy.

4.2.2 Conflicting Over-General Classifiers

The inclusion of wild-card characters in the classifiers effects a generalization mechanism. Generalization is a powerful feature if there are several global states whose sense-vectors differ in some respects but for which the same action is appropriate: one classifier advocating an appropriate action can match multiple states if it has sufficient wildcards to ignore the differences between the sense-vectors. For example, in the woods environments, if there is a F in a particular cell adjacent to the animat, then a classifier with wildcards at all positions in the sense-vector except for the two bits corresponding to the position of the F can advocate moving in the correct direction (i.e. onto the F, generating a reward) *regardless* of the particular surrounding pattern of Ts and blank all

advocate inappropriate actions.


T

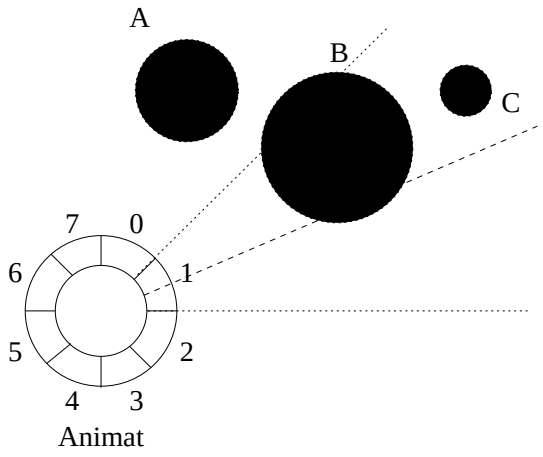


Figure 42: Schematic illustration of the flatland vision system. The circular animat is shown with 8 equal-angle 'visual' sensors occupying its circumference. The dark circles are obstacles in the world. The dashed line indicates the 'optical axis' of sensor 1, and the dotted lines indicate the limits of its angle of acceptance. Obstacle A has no effect on sensor 1 because it is outside the angle of acceptance.. Obstacle B subtends over half the acceptance angle of sensor 1. Obstacle C has no effect on sensor 1 because it is occluded by obstacle B: if B was removed, C may be too distant to subtend a sufficient proportion of the sensor's angle of acceptance to affect it.

prior research in modeling visual localization and navigation in bees and wasps (Cartwright & Collett, 1983, 1987; Collett, 1992), which used similarly minimal models.

In the current FLAVA work, time proceeds in discrete timesteps. On each timestep, the animat makes a movement: the movement is a step of fixed length, whose direction is chosen randomly within some range specified by the classifier's external action. For compatibility with our earlier experiments, we gave the ZCS-FLAVA animat 8 external actions: as with the cellular woods environments, the action specifies one of eight directions of movement, with 45° between successive directions. On each step, a uniform random deviate over [-22.5°,22.5°] is added to the specified movement direction before the step is taken. The net effect is that an animat applying a constant external action will wander in a consistent direction with random drift: a biased "drunkard's walk" over a continuous space.

The animat can move in the two spatial dimensions, but maintains a constant orientation. It cannot move into space occupied by an obstacle. It has a one-bit omnidirectional "collision detector" which outputs 1 if the animat attempts to collide with an obstacle, and 0 otherwise. There is no noise or uncertainty in either the 'visual' sensors or the 'collision detector'. Drawing inspiration from studies of visual navigation in animals, the location of rewards in our FLAVA studies are *not* detectable by the animat's sensors: the animat has to find its way to a reward by reference to 'visual landmarks' alone.

Further realism is introduced by making the length of each random movement-step small in comparison to the size of the world: sequences of small movement steps can be considered as an approximation to spatiotemporally continuous movement. However, this approach raises the problem of the temporally discrete nature of the classifier system: ZCS, like most other classifier systems, operates in discrete timesteps, each composed of one run through the sense-act-reinforce-discovery sequence. If Δt represents the inter-timestep interval, then reducing Δt (to get a better approximation to continuous movement) compresses the timescale over which chains of actions can be supported. To avoid this problem, we use ZCS in a *sensory interrupt* fashion: once an external action is selected, that (noisy) action is applied on each timestep in a pure sense-act loop (i.e. *without* reinforcement or discovery) which is only interrupted when

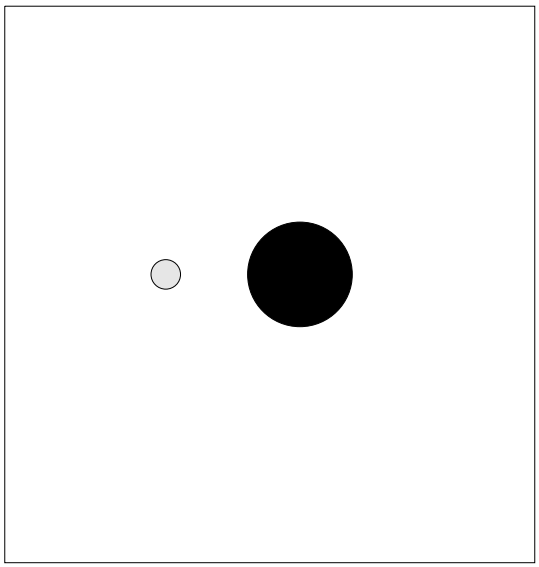


Figure 43: A simple FLAVA environment. The large dark circle is an obstacle; the small light circle denotes the zone where the animat will receive reward (the animat cannot sense this zone). See text for discussion.

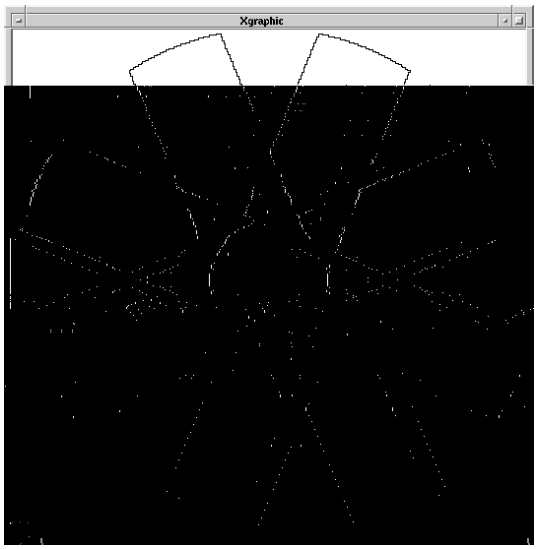


Figure 44: Sensory signature neighborhoods for the environment illustrated in Figure 43. See text for discussion.

``

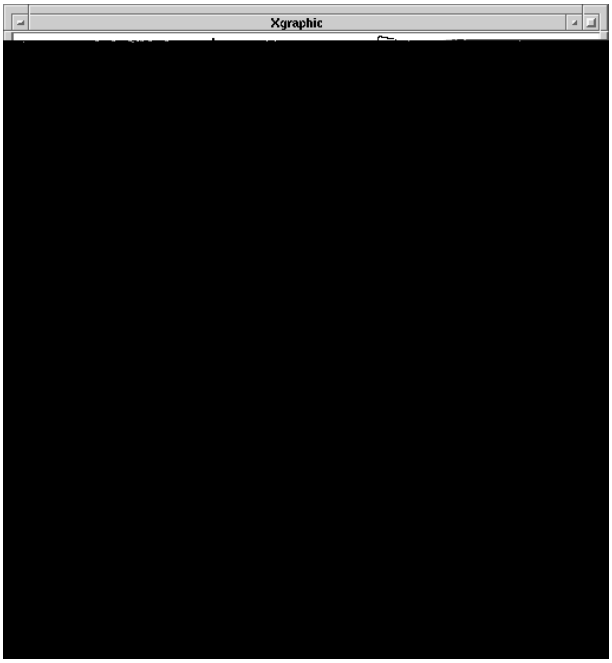


Figure 49: Sensory signature neighborhoods for a FLAVA world with three circular obstacles. See text for discussion.

problems faced by ZCS are not artifacts of the cellular woods environments.

6 Conclusions

Our overall conclusion is that, while ZCS plays an important role in bringing out the similarities between classifier system learning and Q-Learning, its minimalism may cause problems if it is to be used in applications requiring either large amounts of internal state, or the maintenance of long chains of actions. However, these two issues are problematic for most classifier systems, so ZCS is no worse than many other systems; moreover, its minimalism offers clarity in determining the causes of the problems, and in assessing the effects of proposed remedial mechanisms.

To summarize, there are three significant contributions to the literature in this paper.

First, we have demonstrated a reconstruction of ZCS as described by Wilson (1994b) and have shown that Wilson's published results can be replicated. Replication of published results is an important (if unglamorous) task in any scientific field.

Second, we have implemented Wilson's proposal for adding temporary memory to ZCS. Although our results demonstrate that Wilson's proposals work when the number of bits of temporary memory is small, combinatorics indicate that the approach will not scale well as the number of bits required increases. Nevertheless, our demonstration that ZCS could converge on stochastic solutions when there was insufficient internal state to disambiguate sensory aliases implies that there may be applications for which insufficient memory is not an insurmountable problem.

Third, we have demonstrated the limits of performance for ZCS in maintaining long chains of actions. It is informative that even the minimalist ZCS suffers from fragility reported in other, more complex classifier

to be poor.[7] One plausible mechanism is to allow accuracy to play a role in calculating classifier fitness: Wilson's recent work on XCS (Wilson, 1994a) is clearly a promising development in this direction.

Acknowledgements

The authors thank Stewart Wilson for his interest and encouragement over the course of this work, and especially for his comments on earlier drafts of this paper. Thanks also to Jean-Arcady Meyer for his comments on an earlier draft. Thanks to Malcolm McIlhagga and Hilary Tunley for allowing use of their SPARC machines. Financial support for Susi Ross came from a UK SERC/EPSRC MSc studentship grant.

Reference

Booker, L. B. (1989). Triggered rule discovery in classifier systems. In Schaffer, J. D. (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms*, pp. 265–274. Morgan Kaufmann.

Brooks, R. A. (1992). Artificial life and real robots. In Varela, F. J., & Bourgine, P. (Eds.), *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life (ECAL91)*, pp. 3–10 Cambridge MA. M.I.T. Press Bradford Books.

Cartwright, B. A., & Collett, T. S. (1983). Landmark learning in bees: Experiments and models. *Journal of Comparative Physiology*, *151*, 521–543.

Cartwright, B. A., & Collett, T. S. (1987). Landmark maps for honeybees.