# Analysing Breakdowns in Performance in ZCS *

**L. F. González Hernández**
COGS, University of Sussex, Brighton BN1 9QH, U.K.
luisgh@cogs.susx.ac.uk

## Abstract

Wilson's "zeroth-level" classifier system, suggested as a minimalist system that would allow a better understanding of the inner workings of classifier systems, has been shown to suffer performance breakdowns in simple Markovian environments due to its inability to support long chains of actions. Cliff and Ross [1] suggested some possible explanations as to why this may happen. In this paper, their conclusions are re-examined and extended to include covering as the fundamental mechanism that causes the bad performance to be sustained for several trials.

## 1 Introduction

Wilson's "zeroth-level" classifier system (ZCS)[5], was proposed as an attempt to produce a simple system which, though maintaining much of the original framework by Holland, strips it from some of the more complex mechanisms that obscured the understanding of the inner workings of the system. Wilson showed some similarities between his system and Q-Learning([3], [4]). More recently, Cliff and Ross implemented an extension to ZCS suggested by Wilson; they describe ZCS' performance breakdowns in simple Markovian environments for which large reinforcement chains need to be sustained. The problem of large chains in classifier systems has been well known for some time know, [6], but Cliff and Ross showed that it was present in even such a simple system as ZCS.

In this paper I review the causes proposed for the breakdowns in performance in ZCS and suggest why covering, previously thought to have only a partial effect on this phenomenon is in fact central to it.

The paper is organised as follows, section 2.1 briefly describes ZCS to enable the reader to follow the subsequent discussions. Section 3 describes hitherto proposed explanations for the breakdowns as well as introducing a new alternative. Section 4 describes some experiments that were carried out to investigate the validity of the new alternative. A detailed account of the mechanisms at work is presented.

*This paper has been accepted for presentation at the European Conference on Artificial Life, Brighton, 1997

## 2 Classifier Systems

Classifier Systems (CS), originally proposed by Holland [2], are learning systems in which a set of condition-action productions learn to solve a certain control task by use of reinforcement obtained from the environment that is shared among useful productions. A classifier's *strength* reflects the usefulness that a given production has shown in the past and is used as a basis for a competition against other classifiers for the right to determine the action of the agent at any one time. On a greater time scale, an evolutionary process breeds new classifiers allocating a greater chance of reproduction to stronger productions, and replacing weaker individuals in the classifier population.

The complexity inherent to the framework proposed by Holland led to obscurity that prevented a thorough understanding of the inner workings of CS. ZCS was conceived as a simplified framework that would still retain the main characteristics, yet simpler to understand.

### *2.1 ZCS: Zeroth-level Classifier System*

Although ZCS is described in much detail in both Wilson's and Cliff and Ross' papers, a small description is needed in order to more fully understand the arguments. Any interested reader is referred to any of the mentioned references for further information.

ZCS consists of a population of condition-action productions coded in a ternary alphabet {0,1,#}, where # acts as a wild card in the conditions and occurs with some fixed probability. The system has no internal message list and behaves thus in a reactive way. In cycles, the system will undergo one trial in which classifiers matching environmental messages are first grouped in the *match set*. Based on a roulette, an action advocated in the match set is selected so that those classifiers that suggest it, form the *action set*. Once the action is carried out, the credit assignment cycle will update the strength of the classifiers in a manner reminiscent of Q-learning. Each classifier in the action set has its strength decreased by a fraction determined by the learning rate, $\beta$. They also receive a fraction of any immediate reward as well as a discounted part of the strength of the maximally strong action advocated in the next step. Thus an *im-*

*plicit bucket brigade* is carried out in which a discounted fraction of the strongest action in the next time step is passed onto those classifiers in the action set. Finally, matching classifiers that suggest a different action are taxed in order to achieve definiteness in the actions proposed as the experiment progresses.

To cope with cases in which the match set is either empty or contains a weak and therefore unconvincing set of rules (this is determined by requiring that the strength in the matching set be greater than a fraction $\phi$ of the average strength in the population), covering will take the environmental message and generate a new, possibly more general rule, that matches it and proposes a random action. This rule is the inserted back in the population with a strength set to be equal to the average of the population.

## 3 The Breakdowns

In their paper, Cliff and Ross describe some experiments on simple Markovian environments, the **woods14-*p*** set of worlds. It is a set of simple two-dimensional grid worlds. Those worlds have one food **F** which produces a fixed valued reward upon being eaten. There are $p$ empty locations. At any such location, there is only one movement that will lead the animat to the food. Figure 1 shows one such world.

| | | | | | | |
|---|---|---|---|---|---|---|
| **T** | **T** | **T** | **T** | **T** | **T** | **T** |
| **T** | **T** | | | | **T** | **T** |
| **T** | | **T** | **T** | **T** | | **T** |
| **T** | | **T** | **T** | **T** | | **T** |
| **T** | **F** | **T** | **T** | **T** | | **T** |
| **T** | **T** | **T** | **T** | **T** | **T** | |

Figure 1: **woods14-9** simple grid world.

Given the simplicity of the world, one would expect any learning system to score well on it. It turns out, though, that ZCS fails to produce reliable behaviour. Unexpected breakdowns in performance appear even after the animat has learnt to cope with it. The greater $p$, the more often this breakdown is observed. Cliff and Ross suggest two different mechanisms that together are accountable for this behavior, *greedy classifier generation* and *over-generalisation in some classifiers.*

### *3.1 Greedy Classifier Creation*

In a **woods14-*p*** environment, given that the animat has to go through all cells between its starting point and the food, classifiers that match cells closer to the food will be used more often. In addition, Cliff and Ross argue, given that only a discounted reward is passed down the chain, those classifiers matching the environment of less visited cells, are also those that will have smaller reinforcement. Thus, in the extreme case, death is forced upon all those classifiers that apply to some far cell by the genetic algorithm and thus, next time the animat visits that cell, it will be up to the covering mechanism to produce a correct action. Cliff and Ross, suggest that, although this may happen several times in a row, any trial only finishes when reward is obtained, and thus, even though this mechanism may account for some isolated breakdowns, it can't explain the fact that the performance is affected during several simulation cycles after the observed breakdowns.

### *3.2 Over-general Classifiers*

Wild cards in classifier systems enable rules to capture generalisations over certain environmental conditions that suggest some common action. This can lead, though, to classifiers that match different environmental messages for which the adequate actions differ. These classifiers grow strong in highly rewarded locations and are then applied to incorrect situations.

This is especially critical in **woods14-*p*** worlds, Cliff and Ross argue, where cells situated further away from the food tend to be covered by less strong classifiers. The Hamming distances between the environmental messages corresponding to cells close to the reward and those more distant are small. Thus, over-general classifiers that match cells nearer to the food grow stronger compared to those that might be only applicable to less visited cells and tend to impose their actions there too.

Cliff and Ross summarise "*the worse-than-random performance times for the system when initially adapting and when re-adapting after a collapse seem to be due mainly to the large number of steps required to diminish the strength of inappropriate over-general classifiers to such a degree that covering can take effect in the **F**-distant cells.*"

### *3.3 The Rest of the Story*

Although both mechanisms described in [1] do take place, the phenomenon is somewhat more complicated as I will show later. Over-general classifiers do not produce sustained breakdowns in performance. The way over-general classifiers affect the system is by competing with those that though proposing the correct action are less strong. This continues until the strength of both falls below the limit in which covering is invoked. As described above, ZCS invokes covering when there is either no matching classifier in some situation, something which is fairly infrequent given the size of the population of classifiers and a sufficiently high probability of wild cards, and when the total strength of the matching set is smaller that a fraction $\phi$ of the average strength of the population. Thus, an over-general strong classifier that suggests the wrong action, will normally decrease its strength through the bucket brigade mechanism until the total strength of all classifiers in the match set falls below that threshold upon when the covering is invoked. The match set normally will include classifiers that suggest the appropriate action though less strongly, being therefore ignored. Thus the real effect of an over-general classifier is depleting the strength of appropriate ones until covering is fired.

Whatever the cause for it, covering will in **woods14-*9*** produce an incorrect classifier with a probability of $\frac{7}{8}$, and thus, in general, it will take some successive invocations of covering before the correct action is found. This has a double effect. On the one hand, if a classifier suggesting an appropriate action is still in the match set, its associated strength will be further depleted until it disappears. On the other hand, we have that every newly created classifier will be assigned the average strength in the population. Given that it was created through covering, it will be the strongest classifier within the match set. It would be fair to say that the total strength of the match set will almost be that of the newly created classifier. Unsuccessful attempts by the animat to apply it, will decrease its strength until it is about half its value. At this point, the strength in the matching set will be again below half of the average of the population and a new covering process starts. The classifier to be replaced will be selected using a roulette in terms of the reciprocal of the strength. Thus weak classifiers, possibly associated with other food-distant cells will be deleted. The inappropriate classifier though, is still much stronger, about half the average strength of the population. In general, it will therefore be kept. Thus every covering operation disrupts the population even further. The longer it takes for the correct classifier to be found, the more significant the proportion of the population disrupted will be.

We should note that classifiers applying to cells both somewhat closer or further away from the food may be destroyed. The first will cause further covering to be produced within the same trial while the latter will mean that in following trials, the animat will need to re-learn appropriate actions in those cells. Thus the disruption caused by a chain of covering operations will affect several trials while the animat painfully re-learns how to deal with less visited cells. We expect though that the average of trials won't grow to be as big as in the beginning of the experiments given that only distant cells have to be re-learned.

It is important to note that Cliff and Ross do point out that "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" but they seem to overlook the fact that the more successive coverings in one trial, the more disruption in the population and therefore the more trials it takes for the animat to re-learn the knowledge contained in those lost classifiers.

# 4 Experiments

To investigate the process described in 3.3, experiments in various of the **woods14-*p*** were carried out, maintaining the parameters as in [1].

First, the influence of covering in the disruptions in the performance was investigated. To that end, the amount of covering that took place at any one time as well as the associated change in mean steps to food were recorded. By plotting the changes in steps to food against the number of covering operations that provoked those changes, it was expected that this would show a definite tendency indicating that greater disruptions would happen as a consequence of a greater amount of covering. The results in Figure 2 were obtained out of ten different runs in several **woods14-*p*** environments. During the first trials, the animat uses covering to "seed" the classifier population with appropriate conditions. To avoid confusing those trials with any breakdowns in performance, once an adapted behaviour is reached, the first one hundred trials are not shown.

It can be seen that, as expected, there is a definite, almost linear dependency between the amount of covering operations performed and the changes in the average steps to food taken by the animat. The greater the value of $p$, i.e. the more likely it is that food-distant cells are dealt with by weak classifiers, the greater the chance that a large chain of disruptive covering invocations takes place. So, **woods14-06**, which was found by Cliff and Ross to be very stable, only shows a small amount of significant covering taking place in the whole of the ten runs. On the other hand, **woods14-14** and **woods14-18** seem to be quite prone to covering chains which result in more significant disruptions and thus explain why the performance of ZCS in those environments is unsatisfactory.

When a straight line is fitted, all environments show the same slope of 0.34. This might be due to the fact that the bucket brigade reduces the strength of newly covering-generated rules by the same amount in all environments and therefore, the expected time that it takes for some incorrect classifier to lose enough strength so that a further covering operation may take place is more or less the same.

We can also see that large amounts of covering inevitably result in a great disruption in the population. In particular, we see that when covering is invoked 40 times, which, given the parameters in ZCS means that we expect 10% of the population to be replaced, the dis-

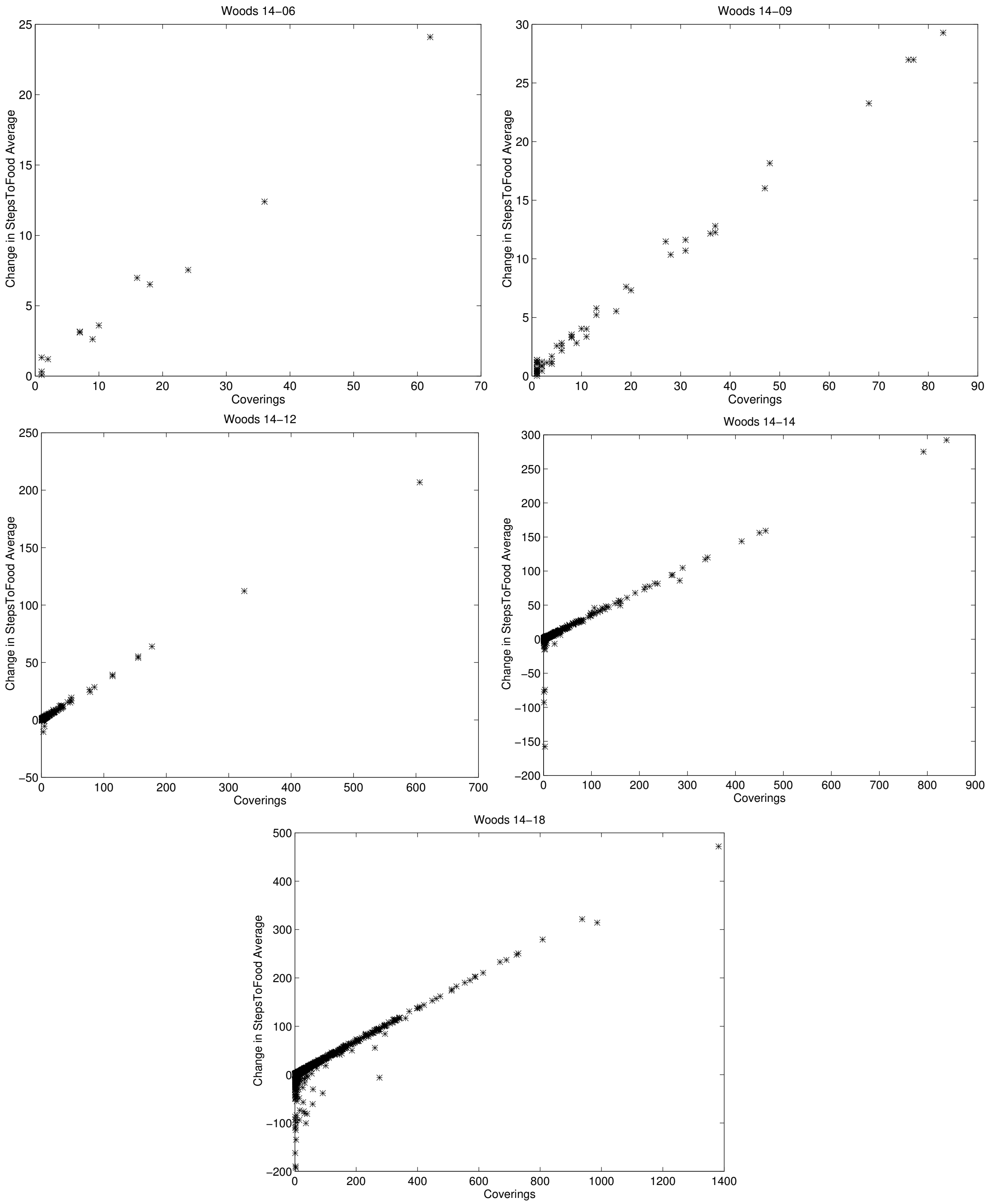


Figure 2: Coverings vs. changes in steps to food in **woods14-**$p$

ruption causes some 15 steps more until the animat obtains the reward. Further, Figure 3, shows one run in **woods14-12** in which we can see that the duration of the breakdowns is more significant the greater the disruption.

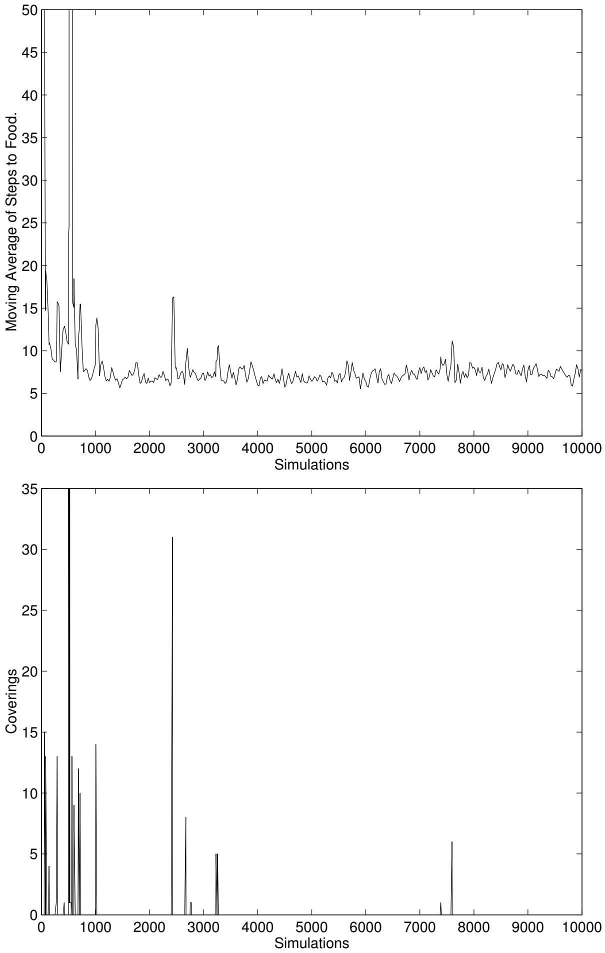


Figure 3: One run of woods14-12 in which we can see how big *chains* of invocation of the covering operations cause disruptions that last for several simulations.

### *4.1 Discussion of a typical Run*

In this section I will describe the specific mechanisms by which the chains of covering are fired, outlined above in section 3.3. The specific examples used are taken from a run in **woods14-15**, shown in Figure 4. This figure also shows the coordinates used in the explanations.

| X \ Y | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | T | T | T | T | T | T | T | T | T | T |
| **1** | T | T |  |  |  | T | T | T | T |  |
| **2** | T |  | T | T | T |  | T |  |  | T |
| **3** | T |  | T | T | T |  | T |  | T | T |
| **4** | T | F | T | T | T |  | T | T |  | T |
| **5** | T | T | T | T | T | T |  |  | T | T |

Figure 4: Coordinates used for **woods14-15**

As for the conditions in the rules, the encoding used represents an empty space as `00`, a tree `T` as `10` and the food as `11`. The sensory strings are formed by concatenating the sensory information in the following order:

```
1 2 3
8 * 4
7 6 5
```

so that

```
T T _
_ * T
T T _
```

is represented as:

1010001000101000

.

First we have the case in which covering generates a classifier, because the total strength in the matching set is less than a fraction $\phi$ of the average of the population. As we can see in Figure 5, the system is in the cell `(1,9)`, i.e. 15 steps away from the food. The appropriate action as seen from Figure 4 would be to go to the southwest. This is suggested by rule `214`. Unfortunately, its strength is not enough because it is far from the reward and covering has introduced a new fairly strong rule, `119`. This rule clearly dominates the set. The action set will therefore only include it and the strength of all the rules in the matching set will decrease until, seven steps later, by chance the appropriate action is selected. Although, in this case, the system is able to continue, the match set for the cell is still inappropriate and will cause some disruption next time the animat lands on it.

```
>Environment: (1,9) ->[1010101010100010]<

MatchSet:
 35:< [10#01#1#1010##10]->[ N]  1.41 >
 87:< [##1###10##100010]->[NW]  2.32 >
119:< [10#0##1#1##00###]->[ E] 16.99 >
181:< [10###0#01#1000#0]->[ W]  1.19 >
214:< [#010101##0100010]->[SW]  2.55 >

Actionset:
119:< [10#0##1#1##00###]->[ E] 16.99 >

...

>Environment: (1,9) ->[1010101010100010]<

MatchSet:
 35:< [10#01#1#1010##10]->[ N]  0.75 >
 87:< [##1###10##100010]->[NW]  2.40 >
119:< [10#0##1#1##00###]->[ E]  8.03 >
181:< [10###0#01#1000#0]->[ W]  0.57 >
214:< [#010101##0100010]->[SW]  2.51 >
```

Figure 5: Covering is fired by an insufficiently strong match set. The rule, `119` is assigned the average strength of the population and thus is stronger than the rest of the members of the match set. This includes a rule, `214` which is only second strongest and does suggest the appropriate action.

The second example illustrates the case in which the disruption is fired by an over-general classifier that is being used in a cell for which the action it suggests is inappropriate. Rule `363` also matches the environmental conditions of the cell $c_5$ in `(1,4)`. Thus it is much stronger than the rest of the match set. Even though the appropriate action is the strongest alternative action in the match set, its strength is smaller than of the rules advocating to go west. We can see that the strength of those over-general rules is depleted by the bucket brigade mechanism until they are no longer significant. At the same time, rule `353` which did suggest the correct action in the first match set, has also lost much of its strength so that by the last step shown in this example, it is actually replaced, through covering, by a new rule with the wrong action associated to it. Thus a more serious disruption through successive coverings is caused.

## 5 Limiting the Amount of Disruption caused by Covering

In order to reduce the disruptions caused by covering, two different mechanisms were tried. On the one hand, the strength assigned to a newly generated classifier was reduced so that instead of being that of the average strength in the population, it would just be barely enough to successfully become the matching set, next time around it matches the environmental message. The idea is that reinforcement will strengthen messages if they turn out to suggest a suitable action but will weaken non suitable productions so that the system doesn't disrupt other weak classifiers unnecessarily. The second technique evaluated was to put an arbitrary but reduced limit to the amount of classifiers that can be affected by covering to some percentage of the population. This would be activated once the system had already converged to some more or less adapted behavior.

### 5.1 *Limiting the strength of Classifiers created through Covering*

In detail, this mechanism is actually very simple. We know that covering will be fired if the strength in the matching set is bellow a fraction of the average strength in the population. We assign each newly generated classifier a strenth equal to

$$S_{|C|} = \frac{\phi \bar{S}}{1-\beta} \tag{1}$$

where $S_{|C|}$ is the strength of a classifier generated through covering, $\bar{S}$ is the average strength in the population and $\phi$ and $\beta$ are respectively the minimum fraction of the average strength of the population required in the matching set and the learning rate. In this way, after a reinforcement cycle is completed, the strength is just $\phi\bar{S}$, the threshold for it to still be a valid matching set on its own next time around.

### 5.2 *Limiting the amount of Classifiers deleted through Covering*

This mechanism would limit the amount of classifiers that covering can replace to a 2% of the population so that repeated covering would not destroy valuable but weak productions. Once the simulation has converged[1], every trial will only be allowed to replace up to a 2% of the classifiers. Any further covering invocations will replace one of the classifiers within that reduced group.

[1] This was set arbitrarily at 1000 trials in the case of the environment studied in detail. This value was deemed appropriate to the environment but is nevertheless an arbitrary threshold.

```
>Environment: (3,7) ->[1000001000101010]<

MatchSet:
158:< [100#0##0#0#0#010]->[ N]  2.80 >
164:< [100#00100##0#01#]->[ W]  6.09 >
263:< [1#00##100010###0]->[ S]  0.06 >
353:< [100000100##010##]->[SE]  4.68 >
363:< [10#0#0100#1010#0]->[ W] 21.45 >
381:< [10000#1#####1##0]->[ S]  0.03 >

Actionset:
164:< [100#00100##0#01#]->[ W]  6.09 >
363:< [10#0#0100#1010#0]->[ W] 21.45 >

.....

MatchSet:
158:< [100#0##0#0#0#010]->[ N]  2.45 >
164:< [100#00100##0#01#]->[ W]  1.18 >
353:< [100000100##010##]->[SE]  0.04 >
363:< [10#0#0100#1010#0]->[ W]  1.19 >
381:< [#000001#0010#0##]->[SW]  4.14 >

Actionset:
381:< [#000001#0010#0##]->[SW]  4.14 >

>Environment: (3,7) ->[1000001000101010]<

MatchSet:
158:< [100#0##0#0#0#010]->[ N]  2.21 >
164:< [100#00100##0#01#]->[ W]  1.32 >
353:< [10000##00#101010]->[ W] 16.78 >
363:< [10#0#0100#1010#0]->[ W]  1.33 >
381:< [#000001#0010#0##]->[SW]  3.31 >

Actionset:
164:< [100#00100##0#01#]->[ W]  1.32 >
353:< [10000##00#101010]->[ W] 16.78 >
363:< [10#0#0100#1010#0]->[ W]  1.33 >
```

Figure 6: An over-general classifier, `363`, competes against `353` which, though much weaker, proposes the correct action in this situation. Several steps later, the strength of both rules has decreased so that the over-general classifier is no longer as relevant but enough so that in the end, `353` is substituted by a randomly generated rule as a result of covering.

This way, it was hoped, an unsuccessful candidate classifier would be much more likely to be selected for replacement and the overall disruption would be limited.

## 6 Empirical Study

To investigate the effects of the different mechanisms, three further sets of experiments were carried out on **woods14-*14***. For control purposes, the results were compared to that of the traditional ZCS in the same environment.

The objective here was to determine the influence of any of the above mechanisms, or the combination of both together, on the *duration* and the *magnitude* of the disruptions, ie. on the capability of the system for recovery from a breakdown. To study the durations, we studied those periods in which the system's moving average steps to food would go over a certain *limit* or threshold. Given a "perfect" rule base the average steps in this environment would be 7.5.

Figure 7 shows the result of the experiments. The number of times that a disruption of a certain length has happened within ten runs is plotted against the length of the disruptions and the threshold or *limit*, such that any trial for which the average steps to food is greater than that limit is considered to be part of a disruption.

We can see that considering as disruptions, those scores higher than 14, the traditional ZCS would still have disruptions of up to 400 and certainly many between 100 and 200 steps. We also see that the strategy of limiting covering to only a 2% of the population is far too simplistic and does not produce the hoped improvements. In fact what we see is that convergence is affected and there are a greater number of long lasting disruptions. In fact, as the limit increases, the duration of disruption decrease and thus the performance also increases. This shows that the system is incapable of improving the performance through learning once it has more or less converged, as it did for the traditional ZCS. In other words, there seems to be a trade off between allowing covering to affect a greater part of the population and the capacity of the system to improve at latter stages of the learning process. Overall, even with an increased lower bound, there is almost no significant gain over the traditional ZCS.

Limiting the initial strength after covering, on the other hand seems to produce a much stronger difference on the behavior of the population. Even for lower limits the durations of the longest breakdowns are reduced compared to those found in ZCS. Increasing the limit further reduces them.

Finally, a set of experiments in which a combination of both mechanisms was tried out, turns out to produce a much worse result than that originally in ZCS. Again, we can see that there is a tendency of having very long disruptions in which the actual steps that the system

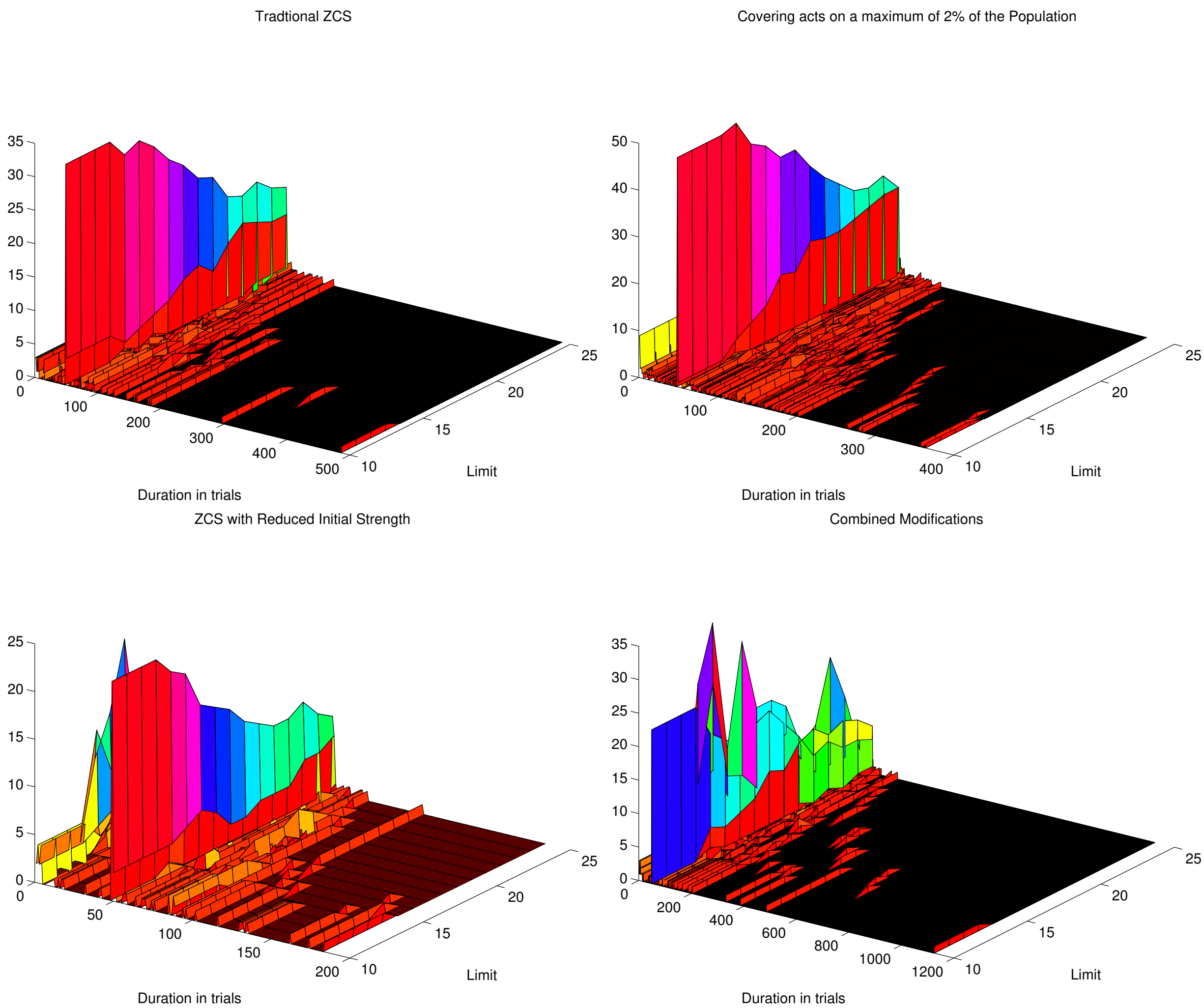


Figure 7: Attempts at dealing with coverings disruptions in **woods14-*14***

takes to find the food on average are not very far from the limit we are setting and thus, it seems, we find again a trade off between the amount of disruption that we allow covering to cause and the speed of convergence in the learning process.

It seems that the strategy of limiting the amount of covering to a 2% of the population is too simplistic. We are allowing covering to randomly select some classifiers and delete them according to the inverse of the strength. It seems that we need a more powerful heuristic to select which classifiers to replace and maybe even how many to allow.

## 7 Conclusions

In this paper, the causes of the poor performance of ZCS in simple Markovian environments are investigated. Previous attempts at explaining this behaviour are described. It seems that although those mechanisms described in [1] are present in the system, they are only one of the causes of a more deeper disruption. This is mainly caused by chains of covering invocations that force death upon a significant proportion of the classifiers in the population thus making the re-learning process to take place more slowly.

Some attempts at solving the situation are proposed and investigated. In particular a simple strategy for limiting the amount of classifiers replaced proved to be too simplistic but shows a trade off between the percentage of the population that we let covering change and the speed at which the system learns refined behaviours. Limiting the initial strength of classifiers produced through covering, on the other hand, seems to produce better results.

## 8 Acknowledgements

The author wishes to thank L. Morales Rueda and Dr. D. Cliff, as well as the anonymous reviewers, for their useful comments. This work has been partially supported by a postgraduate study grant from the Government of the Canary Islands[2].

## References

[1] D. Cliff and S. Ross. Adding temporary memory to ZCS. *Adaptive Behvavior*, 3(2):101–150, 1995.

[2] J. H. Holland. Processing and processors for schemata. In E. L. Jacks, editor, *Associative information processing*, pages 127–146. American Elsevier, New York, 1971.

[3] C.J.C.H. Watkins. *Learning from Delayed Rewards.* PhD thesis, Cambridge University, Cambridge, U.K., 1989.

[4] C.J.C.H Watkins and P. Dayan. Technical note: Q-learning. *Machine Learning*, 8:279–292, 1992.

[5] S. E. Wilson. ZCS: A zeroth level classifier system. *Evolutionary Computation*, 2(1):1–18, 1994.

[6] S. W. Wilson and D. E. Goldberg. A critical review of classifier systems. In J. D. Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 244–255, George Mason University, 1989. Morgan Kaufmann.

[2]Convocatoria 1995, B.O.C. N° 127, 2/10.

## 9 Appendix

This appendix reproduces the means and standard deviation of the traditional ZCS compared with that of each of the proposed mechanisms to reduce the disruptions. The "limit" refers to the threshold for considering some number of steps as a disruption as described in the main section of the paper (Section 6).

| | Traditional ZCS | | Restricted Cover. | | t-test |
|---|---|---|---|---|---|
| Limit | Mean | Std.Dev. | Mean | Std.Dev. | |
| 10 | 60.79 | 62.38 | 66.96 | 65.34 | -1.27 |
| 11 | 60.79 | 62.38 | 66.96 | 65.34 | -1.27 |
| 12 | 60.79 | 62.38 | 66.96 | 65.34 | -1.27 |
| 13 | 60.79 | 62.38 | 66.96 | 65.34 | -1.27 |
| 14 | 43.37 | 53.49 | 45.96 | 52.46 | -0.73 |
| 15 | 44.26 | 49.39 | 40.58 | 43.95 | 1.13 |
| 16 | 47.52 | 39.81 | 36.63 | 40.25 | 4.12 |
| 17 | 40.73 | 38.62 | 37.28 | 38.77 | 1.25 |
| 18 | 41.22 | 37.69 | 39.44 | 34.18 | 0.62 |
| 19 | 39.60 | 34.23 | 36.72 | 33.08 | 1.09 |
| 20 | 38.96 | 31.18 | 36.39 | 30.32 | 1.02 |
| 21 | 38.44 | 29.29 | 34.89 | 29.29 | 1.47 |
| 22 | 34.63 | 29.04 | 33.77 | 28.97 | 0.35 |
| 23 | 35.54 | 28.77 | 37.88 | 28.25 | -0.88 |
| 24 | 34.36 | 29.39 | 37.85 | 27.09 | -1.25 |
| 25 | 36.34 | 29.64 | 37.11 | 25.94 | -0.27 |

| | Traditional ZCS | | Reduced Strength | | t-test |
|---|---|---|---|---|---|
| Limit | Mean | Std.Dev. | Mean | Std.Dev. | |
| 10 | 60.79 | 62.38 | 54.51 | 42.47 | 0.90 |
| 11 | 60.79 | 62.38 | 54.51 | 42.47 | 0.90 |
| 12 | 60.79 | 62.38 | 54.51 | 42.47 | 0.90 |
| 13 | 60.79 | 62.38 | 54.51 | 42.47 | 0.90 |
| 14 | 43.37 | 53.49 | 37.26 | 37.27 | 1.18 |
| 15 | 44.26 | 49.39 | 31.60 | 35.03 | 2.67 |
| 16 | 47.52 | 39.81 | 27.61 | 30.42 | 5.19 |
| 17 | 40.73 | 38.62 | 31.50 | 28.64 | 2.17 |
| 18 | 41.22 | 37.69 | 32.92 | 26.92 | 1.86 |
| 19 | 39.60 | 34.23 | 30.72 | 27.77 | 2.13 |
| 20 | 38.96 | 31.18 | 36.72 | 28.46 | 0.51 |
| 21 | 38.44 | 29.29 | 40.59 | 28.67 | -0.48 |
| 22 | 34.63 | 29.04 | 37.93 | 29.15 | -0.74 |
| 23 | 35.54 | 28.77 | 39.32 | 27.36 | -0.81 |
| 24 | 34.36 | 29.39 | 43.67 | 24.14 | -1.82 |
| 25 | 36.34 | 29.64 | 42.34 | 25.64 | -1.15 |

| | Traditional ZCS | | Combined | | t-test |
|---|---|---|---|---|---|
| Limit | Mean | Std.Dev. | Mean | Std.Dev. | |
| 10 | 60.79 | 62.38 | 77.46 | 133.64 | -2.65 |
| 11 | 60.79 | 62.38 | 77.46 | 133.64 | -2.65 |
| 12 | 60.79 | 62.38 | 77.46 | 133.64 | -2.65 |
| 13 | 60.79 | 62.38 | 77.46 | 133.64 | -2.65 |
| 14 | 43.37 | 53.49 | 45.89 | 83.20 | -0.58 |
| 15 | 44.26 | 49.39 | 33.42 | 68.04 | 2.95 |
| 16 | 47.52 | 39.81 | 43.24 | 74.09 | 1.19 |
| 17 | 40.73 | 38.62 | 37.85 | 62.23 | 0.82 |
| 18 | 41.22 | 37.69 | 40.13 | 55.05 | 0.29 |
| 19 | 39.60 | 34.23 | 37.10 | 54.45 | 0.73 |
| 20 | 38.96 | 31.18 | 34.19 | 42.05 | 1.50 |
| 21 | 38.44 | 29.29 | 40.01 | 42.25 | -0.46 |
| 22 | 34.63 | 29.04 | 32.39 | 38.22 | 0.69 |
| 23 | 35.54 | 28.77 | 27.40 | 35.64 | 2.58 |
| 24 | 34.36 | 29.39 | 30.10 | 30.42 | 1.20 |
| 25 | 36.34 | 29.64 | 27.75 | 29.71 | 2.35 |

This are the results of the T-student test statistic to check whether the mean is significantly different in the Traditional ZCS and the other. Given the degrees of freedom, the statistics have to be greater than $t_{\frac{\alpha}{2}}$, where for $\alpha = 0.05$, $t = 1.645$, and for $\alpha = 0.025$, $t = 1.960$.

In order to analyse this results, there are to things that need to be taken into account. Firstly, peformances where the number of steps goes over the 14 require that the animat is producing at least some inadequate movement. So, the greater the *limit*, the more real disruptions and less pure randomly long performances are included in the statistic. Secondly, to understand why the mean can increase with the limit, consider the situation when the current limit is just enough so that a great number of very small peaks are included when working out the statistics. This will lower the mean as there will be a great number of small peaks. By increasing the limit slightly, all those peaks are removed and only big significant disruptions are left over, thus the mean increases.

Looking at the t-student statistics, there is no clear veredict as to whether there is a significant statistical difference between the traditional ZCS and the solutions attempted. For some cases, specially for a low limit in the combined solution, the performance is actually *worse* with the solutions proposed. But for higher limits, if there is some significant difference, it points to a better performance under the solutions proposed[3]. The results are inconclusive but encouraging.

[3] The only exception is the mean found for a limit of 24 in the case where the newly generated classifiers are assigned a reduced strength.