A Zeroth Level Corporate Classifier System

Andy Tomlinson
Faculty of Computer Studies and Matematics
Intelligent Computer System Centre
University of the West of England
Bristol BS16 1QY, U.K.
blob @ics.uwe.ac.uk

Larry Bull
Faculty of Computer Studies and Matematics
Intelligent Computer System Centre
University of the West of England
Bristol BS16 1QY, U.K.
larry@ics.uwe.ac.uk

Abstract

It has long been recognised that increased co-operation amongst the classifiers in a Michigan-style classifier system may resolve some of the established difficulties associated with the design. One approach to this was proposed by Wilson and Goldberg - the “corporate” classifier system. In this paper we implement the “corporate” classifier system design, within Wilson’s ZCS, in such a way that it complies with their theoretical proposals. In the resultant system , a zeroth-level corporate classifier system, all classifiers initially stand alone but during the course of evolution, a mutation-type operator is used to couple together classifiers by means of structural links. Linked classifiers are considered to represent a corporation, and are treated as a unit by the discovery mechanism of the system. This is achieved by the use of a macro-level evolutionary operator called "corporate crossover". In this design the production system remains oblivious to corporations and operates as ZCS. A technique referred to as concept analysis is introduced which is used to clarify the effects of such rule associations, as implemented here, within a Michigan-style classifier system.

1 INTRODUCTION

We have previously presented a corporate classifier system (Tomlinson and Bull, 1998, 1999) which links individual rules in the population to form rule-chains or corporations which represent temporal chains of inference and map aspects of multiple time-step tasks. Such rule structures have been shown to overcome sensory ambiguities by the use of contextual links formed between rules in consecutive match sets. Both the discovery component and the performance component were responsive to the presence of corporations.

Within the performance component, corporations are able to take “persistent” (Tomlinson and Bull, 1998) control over a number of time-steps, i.e. once a corporate rule wins the auction and while each successive rule in the chain matches each successive stimulus, the currently active rule in the corporation inherits control and automatically determines the system action on that time-step (after Smith, 1992).

This mechanism is contrary to the proposals of Wilson and Goldberg (1989) which state that the performance component should not be influenced by corporations. In this paper we investigate the feasibility of Wilson and Goldberg’s proposals regarding corporate classifier systems which have as yet only been considered theoretically.

The concept of rule corporations is derived originally from the biological phenomenon of symbiosis with the aim of encouraging stronger co-operation between the rules comprising the system and thus to eliminate unwanted parasites and improve efficiency.

ZCS, a Zeroth Level Classifier System, was presented by Wilson (1994) as a simplified Michigan style classifier system (Holland, Holyoak, Nisbett, & Thagard, 1986). The aim of this model was to gain a clearer insight into the characteristics of the basic components of the system. ZCS was tested in two environments, Woods1 and Woods7. ZCS and the Woods environments are described in section three, and section four explains what modifications must be made to ZCS to convert it into a zeroth-level corporate classifier system (ZCCS). Results are presented which compare the performance of ZCCS to that of ZCS in the Woods environments.

HCS, a hierarchically structured classifier system, (Shu & Schaeffer,1991) has many similarities with the concept of a corporate classifier system. Individual classifiers within the system are for some purposes grouped into family units. Increased co-operation within the system is encouraged by the imposed inter-dependencies. HCS was tested on a series of Boolean functions. Section five describes HCS and goes on to compare performances of ZCS and ZCCS on one of the Boolean functions used by Shu and Schaeffer.

The results of testing indicate that ZCCS is able to learn the

Boolean function faster than ZCS. In section six rulebase analysis is used to illustrate how ZCCS learns faster than ZCS. Section seven demonstrates that the improvements gained by using ZCCS can be obtained with ZCS once system parameters are optimised. Results are presented to confirm this theory.

2 CORPORATIONS

The corporate classifier system (CCS) was first introduced by Wilson and Goldberg (1989) as a theoretical approach to alleviating the co-operator/competitor dilemma in the Michigan-style classifier system, discussed in the same paper. The Pittsburgh-style classifier system (Smith, 1980) maintains a population not of rules, but of entire rule-bases. These are evaluated as a unit and so the rules within the rule-bases form natural co-dependencies under the genetic algorithm (GA)(Holland, 1975) , e.g. (Grefenstette, 1987). It was felt that a similar effect may be achievable in a Michigan style classifier system if rules could link to form co-operative clusters for the purpose of reproduction.

These rule clusters or corporations can only be reproduced or deleted as a unit, and are formed by a mutation type operator. The performance and reinforcement components of the system behave as they do in a standard system; it is the discovery component that is effected by the presence of corporations.

For reproduction, the fitness of a corporation is dependent on the strengths of its members (possibly the average strength). If average strength is used to determine the fitness of a corporation then this may be sufficient to encourage corporate linkage as increased stability is generally advantageous and any bucket brigade transaction within a corporation would leave its overall fitness unchanged.

3. ZCS: AN OVERVIEW

3.1 DESCRIPTION OF ZCS

ZCS, like most classifier systems, periodically receives a binary encoded "message" from the environment via sensory detectors. The system determines a response based on this input and feeds a message to the system's effectors which perform the indicated action. A suitable training environment will reward desired behaviour by providing some scalar reinforcement. Internally the system cycles through a sequence of performance, reinforcement and discovery on each time-step of the training period. It is during the performance phase of the cycle that input is received and actions are executed.

The ZCS rule-base consists of a population of N condition/action rules or classifiers. The rule condition is a string of characters from the ternary alphabet {0,1,#}, where # acts as a wildcard allowing generalisation. The action is represented by an integer value and both conditions and actions are initialised randomly prior to test runs. Also associated with each classifier is a strength scalar which acts as an indication of the perceived utility of that rule within the system. This strength of each rule is initialised to a predetermined value termed S_0.

On receipt of a message, the rule-base is scanned and any rule whose condition matches the message at each non-# position is tagged as a member of the current match set [M]. An action is selected from those advocated by the rules comprising [M]. In ZCS this is performed by a simple roulette wheel selection policy. Once an action has been selected, all rules in [M] that advocate this action are tagged as members of the action set [A] and the system effectors execute the action.

Reinforcement in ZCS consists of redistributing strength between subsequent action sets during training. A fixed fraction β of the strength of each member of [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 pre-determined discount factor. 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 is imposed on the members of [M] that do not belong to [A] on each time-step in order to encourage exploitation of the stronger classifiers.

ZCS employs two discovery mechanisms, a panmictic GA and a covering operator. On each time-step there is a probability $\boldsymbol{p}$ of GA invocation. When called, the GA uses roulette wheel selection to determine two parent rules based on strength. These rules are copied with mutation (with probability μ) and crossover (single point, with probability χ). In this implementation of ZCS the GA produces only a single offspring rule on each invocation, the rule replacement rate $\boldsymbol{p}$ however remains consistent with Wilson’s original experiments (i.e. $\boldsymbol{p} = 0.25$). The parents donate a third of their strengths to the offspring if crossover is employed, or half if not. The new rule replaces an existing member of the population which is chosen using roulette wheel selection based on the reciprocals of rule strengths.

If on some time-step, [M] is empty or has a combined strength of less than half of the population average, then a covering operator is invoked. A new rule is created with a condition which matches the environmental message and a randomly selected action. The rule is then made less specific by the inclusion of #'s with a probability per allele governed by the current specificity of the population. The new rule is given a strength equal to the population average and inserted into the population overwriting a rule selected for deletion as before.

Full details of ZCS can be found in (Wilson, 1994).

3.2 THE WOODS ENVIRONMENT

Woods 1 is a two dimensional rectilinear grid of dimensions 5 x 5. 16 cells are blank, 8 contain rocks and one contains food. The system is viewed as an animat (Wilson 1985) traversing this map in search of food. It 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 they are occupied by rocks. 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.

On each time-step the animat receives a message from the environment which describes the surrounding 8 cells. The message is encoded as a 16-bit binary string with two bits representing each of the 8 cells. A blank cell is represented by 00, food (F) by 11 and rocks(O) by 10 (01 has no meaning). The message is ordered with the cell directly above the animat represented by the first bit-pair, and then proceeding 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 performance would balance out to about 27 steps per trial. Optimum performance in Woods 1 is 1.7 steps. Wilson's animat improved from random performance to about 3 steps/trial over 10,000 trials with 400 rules.

Woods 7 is a similar yet somewhat more demanding environment than Woods 1. Like Woods 1, it is a toroidal grid but of size 58 x18 cells. 57 cells evenly scattered around the map are occupied by food. Each of these cells has rocks positioned randomly in two of the 8 surrounding cells. The rest of the map is blank. Unlike Woods 1, this is a non-Markovian environment. In fact, given the animat's sensory abilities there is only so much that the system can learn in Woods 7. Wilson (1994) 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. Wilson's animat reaches food in about 5, after about 4,000 trials in Woods 7.

4. CORPORATE CLASSIFIER SYSTEMS

4.1 HOW ARE CORPORATIONS IMPLEMENTED ?

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 up with one other rule.)

If corporations are viewed as chains of rules, it is reasonable to assume that 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 initialised as inactive but when two rules are selected for joining, then one of each rules links ("link forward" for one rule, "link back" for the other) will be set to reference the other rule. This linkage is used to encourage associations between rules through the formation of inter-dependent rule chains.

In addition to this each rule also contains a "corporate size" parameter and a "corporate I.D." parameter included to facilitate subsequent processing. Initially size is set to 1 and corporate I.D. is left inactive. Within corporations, all rules will hold the same values for size and corporate I.D, and these are set during the formation of the corporation, either through "corporate joining" or through the action of crossover by the GA.

Here "coupling" occurs panmicticly with random probability on each time-step, in the same manner as the GA. An initial coupling probability of 0.1 (once every ten time-steps on average) was decided on. The optimum rate is likely to be dependent on such factors as population size, GA activity and the nature of the task to be learned.

Rules are selected for coupling randomly from the population. 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.

Based on the proposals of Wilson & Goldberg (1989) 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.

For previous work on genetic linkage mechanisms see for example Ikegami and Kaneko (1990), Kargupta (1996).

4.2 DISCOVERY COMPONENT MODIFICATIONS

Rule replacement is based on the reciprocal of rule fitnesses, not strengths. If a corporate rule is selected for deletion then the corporation is first disbanded, then the rule is tagged for deletion. These are the only modifications required by the covering operator, however the GA alterations require further attention.

The crossover site is selected as usual 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 1/3 of the strength of each parent if crossover is employed (or 1/2 of the parent's strength if it is not).

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, 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. The final modification is to the mutation operator. Mutation is now extended to all members of the new corporation rather than just the new rule derived from crossover (i.e. rule 8 in the example).

4.3 ZCS AS A ZEROTH LEVEL CORPORATE CLASSIFIER SYSTEM

The basic ZCS model was modified to act as a ZCCS. Modifications were implemented as described in the last section and all other system parameters were maintained as in Wilson's original experiments, that is:

Population size, Ps = 400
Initial rule strength, S_0= 20.0
Learning rate, β = 0.2
Discount factor for BBA = 0.71
Proportion of rule's strength deducted as tax = 0.1
Average No. of new rules from GA per time step, $\boldsymbol{p}$ = 0.25
Probability of crossover, χ = 0.5
Probability of mutation, μ = 0.002
Reward from environment, R = 1000

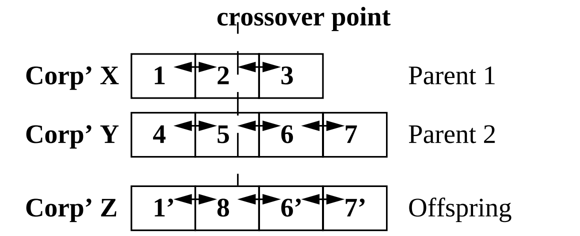

Figure 1: Corporate Crossover

The ZCCS was tested in the same environments as Wilson's original 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. The ZCCS results are presented below for comparison with the ZCS results.

4.4 RESULTS WITH ZCCS

Standard performance/trials plots comparing ZCS and ZCCS in the Woods environments do not clearly illustrate performance differences so plots of "on-line" performance have been included instead. On-line performance is here defined as:

$$P = \sum_{t=1}^{T} (Steps_t)/T$$

where $Steps_t$ = Steps taken at time t and T = Current generation. The resultant plot is representative of the system's learning rate.

In Woods1 ZCCS exhibits a slightly improved learning rate compared to ZCS (Figure 2) but in the more demanding Woods 7 environment ZCCS learns at a reduced rate (Figure 3). 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 the run.

These findings indicate that although corporations do not appear to offer any benefit in Woods 7 they do provide some benefit in Woods 1 (discussed later). ZCCS has also been investigated using a previously presented Boolean logic function.

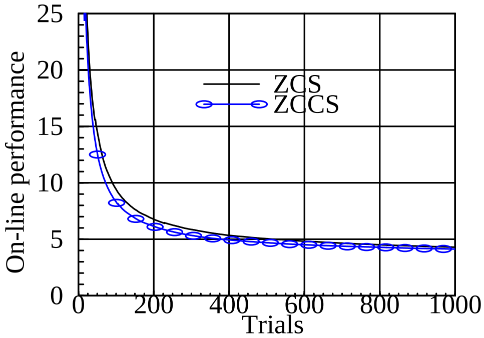

Figure 2: ZCS v ZCCS in Woods 1

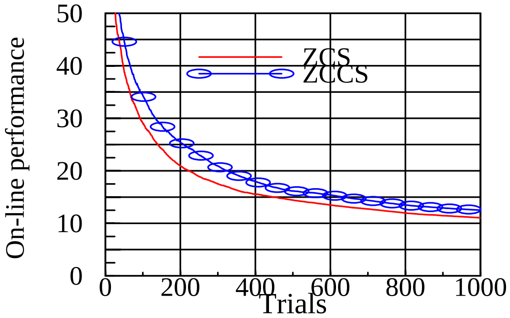

Figure 3: ZCS v ZCCS in Woods 7

5. HCS AND BOOLEAN FUNCTIONS

5.1 HCS: A HIERARCHICALLY STRUCTURED CLASSIFIER SYSTEM

In the closely related work of Shu and Schaeffer (1991) a hierarchically structured classifier system (HCS) in which classifiers are grouped into families by means of structural ties was presented. The aim was to encourage co-adaptation among classifiers. The production system works at the classifier level but most genetic operations are performed at the family level. Within the system, family size is constant and tests were performed to determine the effect of structuring 1, 2, 3 and 4 member families. The system was proposed as a first step towards a hybrid Pitt/Michigan-style system.

The system employs two forms of crossover. In the first, two families are selected and each member of the first family is crossed with the corresponding member of the second in the conventional MCS manner. In the second, two families are again selected but this time crossover swaps family members over between the two. Mutation is not employed in this model and the system exhibits a static structure with family units being established when the rule-base is created.

The bid in HCS is as follows:

$$Bid = \frac{k \times u_c \times u_f \times sp}{2^{l - sp}}$$

where:

k = constant, u_c = utility of classifier, u_f = utility of family, *sp* = specificity and *l* = message length.

5.2 ZCS, ZCCS AND HCS: A COMPARISON

The system was tested on Boolean functions, such as the 5-bit function F2:
$(x_0, x_1, x_2, x_3, x_4) = (x_0 \& x_1)$ OR $(x_2 \& x_3 \& x_4)$.
Performances of HCS with families of 1, 2, 3 and 4 classifiers were compared using the following measure of on-line performance:

$$P = \sum_{t=1}^{T} (C_t - I_t)/T$$

where:
C_t = No. correct responses at generation t, I_t = No. incorrect responses at generation t and *T*= Current generation.

Results showed that HCS on-line performance was improved by the use of family sizes greater than one which suggests that the presence of hierarchies leads to better system organisation.

ZCS and ZCCS were tested on F2 and results are presented below, in the same format as Shu and Schaeffer's results (Figure 4).

On F2 the number of corporations climbed to 35 in the first 500 trials but dropped to 22 after 1000 trials. This figure continued to drop and at the end of the run there were under 10 corporations present on average. This suggests that the benefits of forming corporations are more apparent in the early part of the run but once the task has been learnt their usefulness declines.

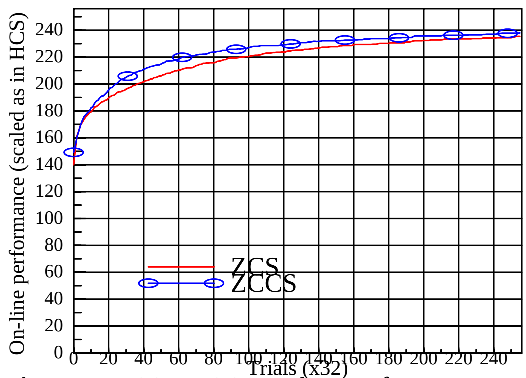

Figure 4: ZCS v ZCCS on-line performance on F2

Figure 4 indicates that ZCCS learns faster than ZCS however standard performance plots (not included) indicate that in the long run performance on F2 is the same. Similar results were found for the other Boolean functions of Shu and Schaeffer.

5.3 DISCUSSION

When comparing these results with those of Shu and Schaeffer (Table 1) it is apparent that ZCS outperforms HCS acting as a standard classifier system (i.e. when family size is set to 1). There may be a number of explanations for this. First, HCS does not employ a mutation operator and this will compromise the discovery abilities of the system. Also, HCS uses a different bid mechanism to ZCS. When ZCS and ZCCS were tested using the HCS bid mechanism there was a noticable degradation in performance (graph not included). One explanation for this is that by incorporating family utilities into the bid, it is possible that the relative range of bids ([strongest bid - weakest bid]/ average bid value) is reduced. This could result in a reduction in discrimination by the production system and this in turn leads to increased uncertainty in decision making. Furthermore specificity is included in the bid and the merits of this approach are uncertain (e.g. Wilson, 1995). Finally, the second of the two crossover mechanisms will have no effect on the population at all if family size is set to 1 and so, in this case GA activity is down to a minimum and this will reduce the system's rate of learning further.

Table 1: Comparison of ZCS, ZCCS and HCS on F2.

Generation (x32)	ZCS	ZCCS	HCS [1]	HCS [4]
60	212	222	194	238
120	227	234	196	239
240	235	238	-	-

When considering ZCCS performance compared to HCS (Table 1) it can be seen that ZCCS reaches almost the same level of performance as the 4-member family version of HCS (its best configuration). HCS seems to learn faster than ZCCS at the beginning of the run. This could be attributed to the fact that HCS is seeded with families at the outset whilst in the current version of ZCCS the initial population consists solely of single rules and corporations are introduced during the course of evolution.

When considering ZCCS performance compared to ZCS, it is reasonable to assume that in the case of a classifier system, an increased learning rate may be indicative of an accelerated ability to organise the existing rule-base, or organise knowledge gained so far. This appears to be the main benefit offered by the corporate classifier system. The question is, how does it achieve this? The most significant difference between ZCS and ZCCS is the nature of the crossover operation. In ZCCS, when a corporate classifier is selected for reproduction, it is possible that more than one rule will be reproduced since fellow corporate members may also be copied (section 4.2). Due to the nature of selection, corporations formed of high utility classifiers will be reproduced more frequently. This suggests that more high utility single classifiers will on average be reproduced per time-step in ZCCS than in ZCS.

6. CONCEPT ANALYSIS

To test this theory an analysis was performed on the system's rule-base. When considering a function of the nature of F2 we can assume that it can be mapped by 20 good rule templates or concepts (for F2 the minimal number of covering concepts is actually 8). During testing, the best 20 different rules in the rule-base at the end of each of ten trials are stored and from these, a "good concept set" is created containing the 20 best overall concepts for that problem as depicted by the system (Table 2).

On ten subsequent trials on the same problem, the rule-base is regularly monitored and the strength of each rule matching a good concept is summed and this value is then divided by the total population strength to give an indication of "good concept set relative utility" (GCS-RU) within the rule-base at that time. When this data is plotted over time it illustrates the growth of GCS-RU over the course of evolution. The idea of concept analysis is derived from the use of macro-classifiers in XCS (Wilson, 1995).

Table 2: Good concept set for F2.

No.	Cond.	Act.	No.	Cond.	Act.
0	11###	1	10	111#0	1
1	10#10	0	11	0#01#	0
2	##111	1	12	#01#0	0
3	01#0#	0	13	10#0#	0
4	#010#	0	14	110##	1
5	0####	0	15	0#0#1	0
6	100##	0	16	0#1#0	0
7	100#1	0	17	#00##	0
8	0###0	0	18	011#0	0
9	1010#	0	19	1110#	1

With rules comprised of a 5-bit condition from a ternary alphabet and a 1-bit action from a binary alphabet there are 486 possible concepts. The population size is set to 240 (as Shu and Schaeffer, 1991) so if good concepts are present in the initial population it is unlikely that they will be duplicated as the rule-base can at best contain only half of all possible concepts.

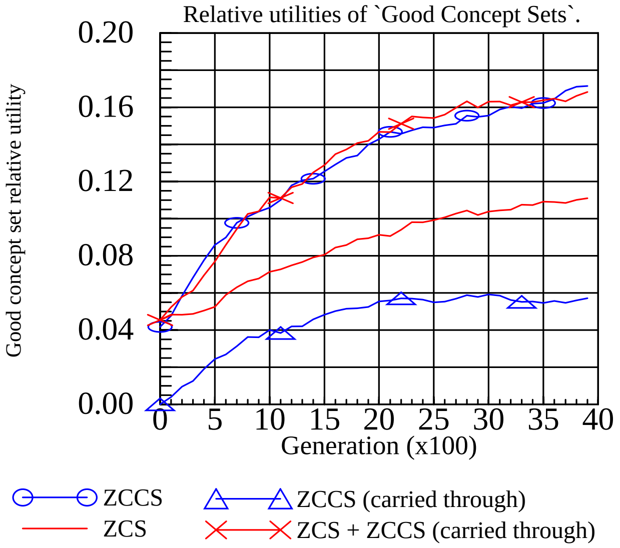

Figure 5: ZCS v ZCCS on F2

The average initial relative utility of a concept can be calculated as: 50% chance of 1/240 = 0.004, and 50% chance of 0, (i.e. the concept is not present in the initial population). For 20 concepts then, the initial relative utility of the set is on average: (10 x 0.004) + (10 x 0) = 0.04.

The GCS-RU should initially equal this value and if the set is reasonable then its strength should grow during evolution. The plot of GCS-RU for ZCS and ZCCS on F2 illustrates these predictions (Figure 5). Further it can be seen that ZCCS establishes these good concepts far more efficiently than ZCS. ZCS improves GCS-RU from 0.04 to 0.11 (a utility growth of 0.07 above average) and ZCCS reaches a GCS-RU of 0.17 (a utility growth of 0.13). Viewed in this way, ZCCS seems to have converged on the ideal concept set almost twice as efficiently as ZCS, however this analysis technique only provides an estimate of rate of convergence or organisation. It can be further shown that the increased rate of convergence is achieved by corporate crossover.

All corporate rules that are reproduced by the GA but which have not actually been selected as parents are tagged as "carried through". When concept analysis is performed, a sub-total is maintained of the strengths of good concepts which have been tagged as carried through. This is also plotted on the GCS-RU graph.

On the same graph it can be seen that the plot of the summation of the ZCCS GCS-RU data and the ZCCS "carried through" GCS-RU data correlates quite closely with the GCS-RU plot of ZCCS, suggesting quite strongly that the accelerated convergence originates from the corporate reproduction.

7. OPTIMIZING SYSTEM PARAMETERS

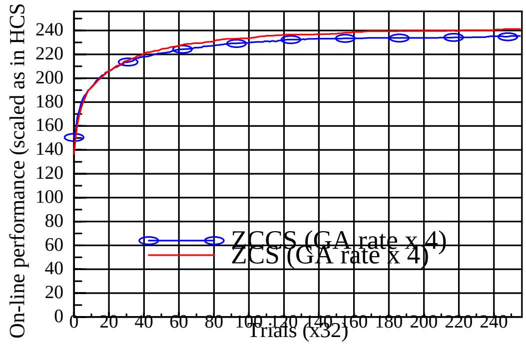

Figure 6: ZCS v ZCCS on-line performance on F2.

If the conclusions drawn from the concept analysis are correct then it seems that in the basic ZCCS model, improvements gained are due simply to increased GA activity. If this is so, then it should be possible to achieve similar improvements by simply increasing the GA activity.

Further tests of ZCS and ZCCS were performed on F2 with the GA rate increased from 0.25, first to 0.5 and then to 1. Plots of on-line performance of ZCS and ZCCS with GA rate = 1 are presented below (Figure 6), along with plots of GCS-RU for ZCS and ZCCS with GA rate = 0.25, 0.5 and 1. (Figure 7).

As the GA rate is increased, so the difference between ZCS and ZCCS results decreases. When the GA rate is set optimally (i.e. a rate of 1 for the F2 task) the use of corporations can degrade performance. This may explain the results of ZCCS in Woods 7. The GA rate of ZCS is already optimised/maximised for Woods 7 and increased GA activity (in the form of corporate crossover) simply leads to premature convergence of the systems rule-base on a subset of the necessary solution set. Woods 1 can be tackled more forthrightly and tolerate increased genetic search. This explains why marginal improvement is observed in learning rate when corporations are used with ZCS in Woods 1. (Although the number of possible stimulus remains constant for Woods environments, the number of presented stimulus is greatly reduced in Woods 1 compared to Woods 7.)

When ZCS (GA rate =1) performance is compared to HCS best performance (family size = 4) on F2, it can be seen that ZCS is able to match the best configuration of HCS, while the single rule version of HCS simply exhibits poor performance in comparison. It now seems most likely that the presented improvements in HCS performance due to the family structures are only possible due to the fact that the basic system has not been thoroughly optimised for the task of solving Boolean functions. More specifically if basic GA activity is increased then system performance would improve, but less, if any gain would be acquired from the inclusion of family structures.

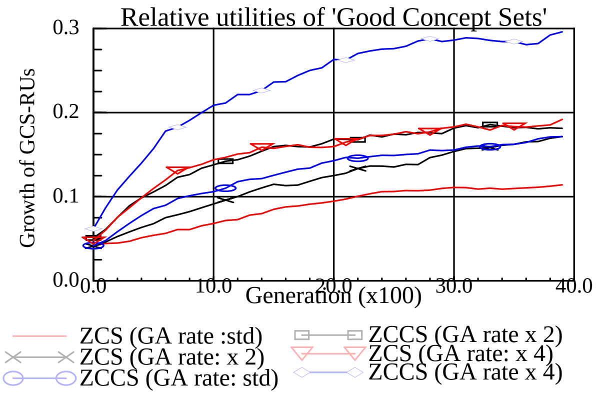

Figure 7: Concepts of ZCS v ZCCS on F2

8. Conclusions

These experiments have shown that it is possible to implement a corporate classifier system as proposed by Wilson and Goldberg (1989). The system used for these experiments can be considered merely a template design kept as minimal as possible, in keeping with the ZCS philosophy.

It is apparent that misleading conclusions may be reached due to a failure to appreciate variations between systems under test, in the rate that the genetic algorithm injects new rules into the population. When these variations are nullified a more insightful comparison can be made between the relative merits of different test models.

Results presented here indicate that no benefit is gained by the presented linkage mechanisms if the system GA rate has been appropriately optimized. There are many ways in which the discussed rule-linkage mechanisms can be enhanced to achieve more directed gains (e.g. Smith's leader/follower corporation (1992)). A design based on such an approach has been implemented and shown to offer significant benefits in certain classes of problems (Tomlinson and Bull, 1998, 1999).

References

Goldberg, D., Kargupta, H., and Harik, G. (1993) “Rapid, accurate optimisation of difficult problems using fast messy genetic algorithms.” In Forrest, S. (Ed.) Proceedings of the Fifth International Conference on Genetic Algorithms, (pp. 56-64), Morgan Kaufmann.

Grefenstette, J. J. (1987) "Multilevel credit assignment in a genetic learning system." In Grefenstette, J.J. (Ed.) Genetic Algorithms and their Applications: Proceedings of the Second International Conference on Genetic Algorithms (pp. 202-209). Lawrence Erlbaum Assoc.

Holland, J. H. (1975) "Adaptation in Natural and Artificial Systems." Univ. of Michigan Press, Ann Arbor.

Holland, J. H., Holyoak, K. J., Nisbett, R. E. & Thagard, P.R. (1986) "Induction: Processes of Inference,Learning and Discovery." MIT Press.

Ikegami, T. & Kaneko, K. (1990) "Genetic Fusion." Physical Review Letters, Vol 65, No. 26 (pp. 3352-3355). The American Physical Society.

Kargupta, H. (1996) “The gene expression messy genetic algorithm.” In Proceedings of the Third IEEE International Conference on Evolutionary Computing. (pp. 814-819) IEEE Press.

Shu, L. & Schaeffer, J. (1991) "HCS: Adding Hierarchies to Classifier Systems." In Belew, R.K. and Booker, L.B.(Ed.) Proceedings of the Fourth International Conference on Genetic Algorithms (pp. 339 - 345), Morgan Kaufmann.

Smith, R. E. (1992) "Memory exploitation in learning classifier systems." Evolutionary Computation, 2(3): 199-220.

Smith, S. (1980) "A learning system based on genetic algorithms." Ph.D. Dissertation (Computer Science), University of Pittsburgh.

Tomlinson, A. & Bull, L. (1998) “A Corporate Classifier System.” In Eiben, A.E., Back, T., Schoenauer, M. & Schwefel, H.P.(Eds.) *Parallel Problem Solving from Nature, PPSN V* (pp. 550-559), Springer.

Tomlinson, A. & Bull, L. (1999) “On Corporate Classifier Systems: Increasing the Benefits From Rule Linkage.” in Banzhaf, W., Daida, J., Eiben, A. E., Garzon, M. H., Honavar, V., Jakiela, M. & Smith, R. E.(Eds) *GECCO-99: Proceedings of the Genetic and Evolutionary Computation Conference*, Morgan Kaufmann.

Wilson, S. W. & Goldberg, D. E. (1989) "A critical review of classifier systems." In Schaffer, J. D. (Ed.) Proceedings of the Third International Conference on Genetic Algorithms, (pp. 244-255), Morgan Kaufmann.

Wilson, S. W. (1985) "Knowledge growth in an artificial animal." In Grefenstette, J. J. (Ed). Proceedings of an International Conference on Genetic Algorithms and their Applications (pp.16-23), Lawrence Erlbaum Associates.

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

Wilson, S.W. (1995) "Classifier Fitness Based on Accuracy." Evolutionary Computation, 3(2): 149-176.