Analyzing the Evolutionary Pressures in XCS

Martin V. Butz
Department of Cognitive Psychology
University of Würzburg
Würzburg, 97070, Germany
butz@psychologie.uni-wuerzburg.de

Martin Pelikan
Illinois Genetic Algorithms Laboratory
University of Illinois at Urbana-Champaign
Urbana-Champaign, IL, USA
pelikan@illigal.ge.uiuc.edu

Abstract

After an increasing interest in learning classifier systems and the XCS classifier system in particular, this paper locates and analyzes the distinct evolutionary pressures in XCS. Combining several of the pressures, an equation is derived that validates the generalization hypothesis which was stated by Wilson (1995). A detailed experimental study of the equation exhibits its applicability in predicting the change in specificity in XCS as well as reveals several other specificity influences.

1 INTRODUCTION

The accuracy based fitness approach in XCS (Wilson, 1995) results in a learning classifier system (LCS) that evolves not only classifiers for best actions, but a complete *payoff map* of the problem. This means that the system evolves an internal representation that can determine the quality of each possible action in each possible state of the encountered environment. Several studies showed that the payoff map in XCS is compact, complete, and accurate.

The purpose of this paper is to clarify and analyze the evolutionary pressures in XCS. The combination of several pressures results in a formula that predicts the change in specificity in the population. This formula validates the generalization hypothesis (Wilson, 1995), which was experimentally investigated in Kovacs (1997). Providing experimental evidence, the formula proves its applicability in an over-general population as well as an accurate one.

The paper starts with an overview over XCS with all involved processes relevant for the paper. Next, the evolutionary pressures in XCS are first analyzed separately and then in interaction. Section 5 provides experimental validation of the claimed pressures, interactions, and parameter dependencies. Finally, a conclusion is provided.

2 XCS OVERVIEW

The XCS classifier system, as it is explained and used herein, incorporates the basics published by Wilson (1995) and the further enhancements in Wilson (1998) and Kovacs (1999). An algorithmic description of the used system can be found in Butz and Wilson (2001). This section gives an overview of XCS emphasizing the formulas and methods important in the remainder of the paper. For further details the interested reader should refer to the cited literature.

As all LCSs the XCS interacts with an environment. The environment provides situations or problem instances σ coded as binary strings of length L (i.e. $\sigma \in \{0,1\}^L$). Furthermore, actions $\alpha \in \alpha_1, ..., \alpha_n$ are executable in the environment. Finally, the environment provides a scalar reward $\rho \in \Re$ reflecting the correctness or quality of the last applied action.

As all LCSs, XCS consists of a population $[P]$ of classifiers which is of fixed length N. The structure of a classifier in XCS is as follows. The *condition part* C specifies where the classifier is applicable. It is coded as a string over the ternary alphabet $\{0, 1, \#\}$ of length L (i.e. $C \in \{0, 1, \#\}^L$). The *action/classification part* A specifies the action/classification of the classifier. It can specify any action executable in the environment ($A \in \alpha_1, ..., \alpha_n$). The *reward prediction* p estimates the payoff encountered after the execution of the specified action. The *prediction error* ϵ estimates the current error of p and is essentially used for the accuracy and resulting fitness determination. The *fitness* F is a measure of the accuracy of p with respect to all competing classifiers. The *experience* exp counts how often the parameters of the classifier were updated. The *time stamp* ts stores the time when last the classifier was in a set where a GA was applied. The *action set size* estimate as approximates the average size of the action sets the classifier belongs to. The *numerosity* num reflects how many micro-classifiers (usual classifiers) this *macroclassifier* represents. This notation is only important for efficiency purposes.

At the beginning of an experiment the population of XCS is usually empty. Sometimes though, the population is initialized with randomly generated classifiers. Each attribute in the condition of such classifiers is set to a #-symbol (a "don't care"-symbol) with a probability $p_{\#}$ and to zero or one (chosen randomly) otherwise. The action is chosen randomly among all possible actions.

A learning cycle at time step t starts with the perception of the actual problem $\sigma(t)$ and the consequent formation of the match set $[M]$. If less than θ_{mna} actions are represented in $[M]$, *covering* occurs. In covering, a matching classifier is created similar to the procedure when initializing the population. Next, an action a is selected randomly with a probability of p_{explr} and deterministic otherwise. Out of $[M]$ an action set $[A]$ is formed consisting of all classifiers that specify action a. The action is executed in the environment and a reward $\rho(t)$ is perceived. With respect to the perceived reward (and the maximal reward prediction in the successive match set in multi-step problems), the reward prediction p, the error measure ϵ, and the action set size estimate as of all classifiers are updated using the Widrow-Hoff delta rule (Widrow & Hoff, 1960).

$$p_{cl} = p_{cl} + \beta * (\rho - p_{cl}) \tag{1}$$

$$\epsilon_{cl} = \epsilon_{cl} + \beta * (|\rho - p_{cl}| - \epsilon_{cl}) \tag{2}$$

$$as_{cl} = as_{cl} + \beta * (\sum_{c \in [A]} num_c - as_{cl}) \tag{3}$$

Parameter $\beta \in (0, 1)$ denotes the learning rate. If the experience of a classifier is still less than $1/\beta$, p, ϵ, and as are updated with the MAM technique ("moyenne adaptive modifiée") which sets the values to the averaged actual values encountered so far. The fitness is updated in three steps.

$$\kappa_{cl} = \begin{cases} 1 & \text{if } \epsilon_{cl} < \epsilon_0 \\ \alpha * (\epsilon_{cl}/\epsilon_0)^{-\nu} & otherwise \end{cases} \tag{4}$$

$$\kappa'_{cl} = \frac{\kappa_{cl} * num_{cl}}{\sum_{c \in [A]} \kappa_c * num_c}$$

$$F_{cl} = F_{cl} + \beta * (\kappa'_{cl} - F_{cl}) \tag{5}$$

First, the accuracy κ is calculated according to the current prediction error ϵ. Next, the relative accuracy κ' is calculated with respect to the current action set. Finally, the fitness is updated according to κ'. Note that the fitness is calculated in terms of *macroclassifiers* while the value of all other measures specifies the micro-classifier value. After all updates and the increase of the experience counter exp of each classifier, a GA may be applied.

The GA is only applied if the average time in the action set $[A]$ since the last GA application, recorded by the time stamp ts, is greater than the threshold θ_{GA}. If a GA is applied two classifiers are selected in $[A]$ for reproduction using a roulette wheel selection with respect to the fitness of the classifiers in $[A]$. Next, the classifiers are reproduced and the children undergo mutation and crossover. In mutation, each attribute in C of each classifier is changed with a probability μ. Two mutation types are investigated herein. In *niche mutation* classifiers are only mutated in such a way that they still match the current state after mutation (i.e. #-symbols are mutated to the current corresponding input value and 1s and 0s are mutated to #-symbols. In *free mutation* an attribute is mutated to the two other possibilities equal probable. Regardless of the mutation type, the action is mutated to any other possible action with a probability μ. For crossover, two-point crossover is applied with a probability χ. The parents stay in the population and compete with their offspring. The classifiers are inserted applying a *subsumption deletion* method. If a classifier cl exists in the population that is more general in the condition part, experienced (i.e. $exp_{cl} < \theta_{sub}$), and accurate (i.e. $\epsilon_{cl} < \epsilon_0$), then the offspring classifier is not inserted but the numerosity of the subsumer cl is increased. Finally, if the number of micro-classifiers in the population exceeds the maximal population size N, excess classifiers are deleted. A classifier is chosen for deletion with roulette wheel selection proportional to its action set size estimate as. Further, if a classifier is sufficiently experienced ($exp > \theta_{del}$) and significantly less accurate than the average fitness in the population ($f < \delta * \sum_{cl \in [P]} f_{cl} / \sum_{cl \in [P]} num_{cl}$), the probability of being selected for deletion is further increased. Note that the GA is consequently divided into a reproduction process and a separate deletion process.

Finally, *action set subsumption* may be applied. This method searches in each action set $[A]$ for the classifier that is (1) accurate, (2) experienced, and (3) most general among the ones that satisfy (1) and (2). If such a classifier exists, it subsumes all classifiers in [A] that are more specific (i.e. specify proper subsets in the condition). The more specific classifiers are deleted and the numerosity of the subsumer is increased accordingly.

3 DISTINCT PRESSURES

Although the description above explains the functioning of the system, it does not become clear why it is any good. To reveal the strength of XCS, this section analyzes the distinct evolutionary pressures separately. Section 4 reveals the interactions between the pressures.

3.1 FITNESS PRESSURE

The parameter update of prediction p, prediction error ϵ and action set size estimate as represented in formulas 1, 2, and 3 assures that the values are an average over all encountered states so far with emphasis on the recently encountered states. The fitness of a classifier is derived from its relative accuracy in $[A]$. It represents the proportional accuracy with respect to all other classifiers in $[A]$. Thus, the selection pressure is a pressure towards accurate classifiers in each environmental niche. The existence and amount of pressure towards accurate classifiers is highly dependent on problem and parameter settings. Note, that in the case when all classifiers in an action set $[A]$ are accurate or similarly inaccurate, the fitness does not directly distinguish the classifiers anymore. In this case the selection process is similar to a random selection in $[A]$. However, an experimental validation in section 5 shows that the noise in the values of unexperienced classifiers can influence the selection process.

3.2 SET PRESSURE

With respect to the population the application of reproduction in the action set results in another pressure. This pressure towards generality was stated by Wilson (1995) in his *generality hypothesis* and was later refined to an *optimality hypothesis* and further experimentally investigated by Kovacs (1997). The basic idea is that classifiers that are more often part of an action set are more often part of the GA and consequently reproduced more often as long as they are as accurate as more specific classifiers. Thus, reproduction in action sets causes an intrinsic pressure towards generality. The amount of expected (lower) specificity of classifiers in an action set is determined by the following formula:

$$s([A]) = \frac{\sum_{k=0}^{L} \binom{L}{k} (\frac{s([P])}{2})^k (1-s([P]))^{L-k} * \frac{k}{L}}{\sum_{k=0}^{L} \binom{L}{k} (\frac{s([P])}{2})^k (1-s([P]))^{L-k}} \qquad (6)$$

Where s denotes the average proportion of specific values in the conditions of the classifiers in the referred set. Considering a specificity of $[P]$ in the population the formula determines the resulting specificity in the action set assuming a binomial specificity distribution in the population. This assumption is certainly valid in the beginning of an experiment if the population is initialized with respect to $p_{\#}$. In this case the average specificity will be $1 - p_{\#}$. It can be observed that k in the numerator and n in the denominator cause the specificity in $[A]$ to be smaller than the specificity in $[P]$. This confirms the proposition of the additional generality pressure mentioned above since the selection takes place in the action set, while deletion takes place in the more specific population. Without fitness pressure, the formula provides an estimate of the difference in specificity of selected classifiers and deleted classifiers as long as a binomial distribution is present. The above equation is enhanced in section 4.1 and experimentally validated in section 5.

3.3 MUTATION PRESSURE

In LCSs mutation appears to have a stronger impact. Generally, a random mutation process causes a tendency towards an equal number of symbols in a population. Thus, applying random mutations in a population of individuals or in particular classifiers, the result will be a population with an approximately equal proportion of zeros and ones or essentially 0, 1, and # in a classifier system. The *free mutation* described above pushes towards a distribution of $1:2$ general:specific while *niche mutation* pushes towards $1:1$. The average change in specificity between the parental classifier $s(cl(t))$ and the mutated offspring classifier $(s(cl(t+1))$ for the niche mutation case can be written as

$$\begin{aligned}\Delta_{mn} = s(cl(t+1)) - s(cl(t)) = \\ s(cl(t))*(1-\mu) + (1-s(cl(t))*\mu - s(cl(t)) = \\ \mu(1-2s(cl(t)))\end{aligned} \qquad (7)$$

and for the free mutation case as

$$\begin{aligned}\Delta_{mf} = s(cl(t+1)) - s(cl(t)) = \\ s(cl(t))*(1-\mu/2) + (1-s(cl(t))*\mu - s(cl(t)) = \\ 0.5\mu(2-3s(cl(t))).\end{aligned} \qquad (8)$$

Thus, mutation alone pushes the population towards a specificity of 0.5 and 0.66 applying niche mutation and free mutation, respectively. The strength of the pressure depends on the mutation type, the frequency of the GA application (influenced by the parameter θ_{ga}), and the probability μ of mutating an attribute.

3.4 DELETION PRESSURE

Due to its proportionate selection method with respect to the action set size estimate as and possibly the fitness F of a classifier, the deletion pressure is difficult to formalize. Generally, the selection of deletion classifiers in the population does not result in any set pressure as encountered in the selection method. Thus, without any bias the average specificity of deleted classifiers is equal to the average specificity in the population $s([P])$.

Due to the bias towards selecting classifiers that occupy larger action sets, deletion stresses an equal distribution of classifiers in each environmental niche. The further bias towards low-fit classifiers was investigated and optimized by Kovacs (1999) in that a low fitness is only considered if the classifier has a sufficient *experience* assuring that the classifier is indeed inaccurate.

3.5 SUBSUMPTION PRESSURE

The final pressure in XCS is the pressure induced by the subsumption deletion method. Due to the experience and accuracy requirement it is assured that subsumption only applies to accurate classifiers. Once accurate classifiers are found subsumption deletion pushes towards maximal *syntactic* generality in difference to the set pressure above which only pushes towards generality if generality also assures a higher applicability rate. *GA subsumption deletion* hinders the insertion of more specific classifiers once an accurate, more general one evolved. *Action set subsumption* is much stronger since an accurate, more general classifier actually absorbs all more specific classifiers regardless if it already existed or was just generated.

To summarize, the subsumption pressure is an additional pressure towards accurate, maximally general classifiers (i.e. classifiers that are still accurate and in the mean time as general as possible). Subsumption only applies once accurate classifiers are found.

4 PRESSURE INTERACTION

After we analyzed the various evolutionary pressures separately in the last section, this section puts the pressures together and analyzes their interaction. First, the interaction of set, mutation, and deletion pressure is formulated. Next, the effect of subsumption pressure is discussed. Finally, we provide a visualization of the interaction of all the pressures. The theoretical analyzes are experimentally validated in section 5.

4.1 SPECIFICITY PRESSURE

When analyzing the interaction of set, mutation, and deletion pressure described above, we realize that all three pressures influence the average specificity in the population. Thus, the three pressures together result in a *specificity pressure* in the population.

Due to the problem dependence of the fitness pressure, we cannot formulate the pressure and consequently need to assume a similar fitness of all classifiers in our analysis. As it will be shown in section 5, this assumption holds when all classifiers are accurate and nearly holds when all are similarly inaccurate. The addition of subsumption pressure is discussed in section 4.2.

Despite the fitness irrelevance assumption, deletion pressure is also dependent on the action set size estimate *as* of a classifier. In accordance with Kovacs's insight in the relatively small influence of this dependence (Kovacs, 1999), we assume a random deletion from the population in our formulation. Thus, as stated above, a deletion results on average in the deletion of a classifier with a specificity equal to the specificity of the population $s([P])$. The generation of an offspring, on the other hand, results in the insertion of a classifier with an average specificity of $s([A]) + \Delta_{fm}$ or $s([A]) + \Delta_{nm}$ dependent on the type of mutation used. Putting the observations together, we can now calculate the average specificity of the resulting population after one learning cycle:

$$s([P(t+1)]) = s([P(t)]) + f_{ga} * \frac{2 * (s([A]) + \Delta_m - s([P(t)]))}{N} \quad (9)$$

The parameter f_{ga} denotes the frequency of the GA application in XCS. The formula adds to the current specificity in the population $s([P(t)])$ the expected change in specificity calculated by the difference between the specificity of the two reproduced and mutated classifiers $s([A]) - \Delta_m$ and $s([P(t)])$. Although the frequency f_{ga} is written as a constant in the equation, it is actually dependent on $s([P(t)])$ as well as the specificity distribution in the population. Thus, f_{ga} can generally not be written as a constant. By setting θ_{ga} to one, it is possible, though, to force f_{ga} to be one since the average time since the last GA application in an action set (not generated by covering) will always be at least one.

4.2 ADDING SUBSUMPTION

Although we revealed the cause and existence of the set pressure that pushes the population towards more general classifiers once all are accurate, we also showed that this pressure is somehow limited. Equation 9 shows that without subsumption the convergence of the population towards accurate, maximally general classifiers is not assured. Essentially, if the specificity of the accurate, maximally general classifiers in a problem is lower than the value of the converged equation 9, then the population will not completely converge to those classifiers. Another reason for a lack of convergence can be that the set pressure is not present at all. This can happen, if XCS encounters only a subspace of all possible examples in the universe $\{0,1\}^L$.

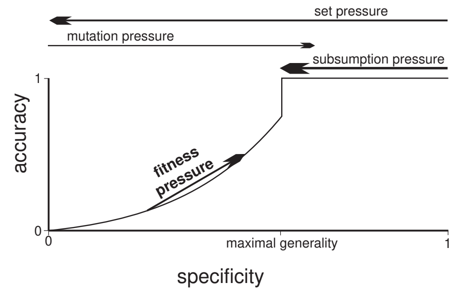

Figure 1: Together, the evolutionary pressures lead the population towards the accurate, maximally general classifiers.

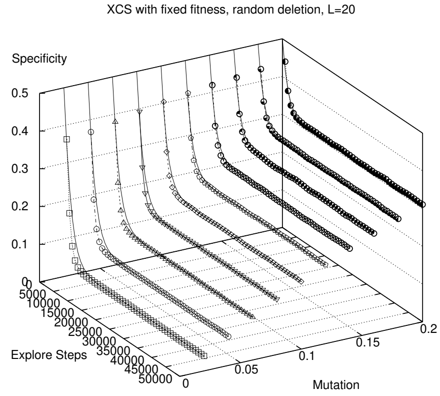

Figure 2: Eliminating the fitness influence, the specificity in XCS behaves exactly like the theory.

In this case, the subsumption pressure results in a further convergence to the intended accurate, maximally general classifiers.

4.3 ALL PRESSURES

Finally, fitness can influence several other pressures as mentioned above. Generally, the fitness pushes from the over-general side towards accuracy as long as the environment provides helpful, layered payoff or the consistency of predicting an outcome is biased in over-general classifiers as revealed in Butz, Kovacs, Lanzi, and Wilson (2001). Thus, in terms of specificity fitness results in a pressure towards the specificity of maximally general classifiers from the over-general side. The pressures are visualized in figure 1. While set and mutation pressure (free mutation is visualized) are accuracy independent, subsumption and fitness pressure are guided by accuracy. Due to its distinct influences, deletion pressure is not visualized.

5 EXPERIMENTAL VALIDATION

In order to validate the proposed pressures and the specificity behavior formulated in equation 9 we apply XCS to boolean strings of length $L = 20$ with different settings. The following figures show runs with varying mutation from 0.02 to 0.20. In each plot the solid line denotes the formula and the dotted lines represent the XCS runs. All curves are averaged over 20 experiments. If not stated differently, the population is initially filled up with randomly generated classifiers with don't care probability $p_{\#} = 0.5$. Niche mutation is applied. The remaining parameters are set as follows: $N = 2000$, $\beta = 0.2$, $\alpha = 1$, $\epsilon_0 = 0.001$, $\nu = 5$, $\gamma = 0.95$, $\theta_{ga} = 1$, $\chi = 0.8$, $\theta_{del} = 20$, $\delta = 0.1$, $\theta_{sub} = \infty$, and $p_{explr} = 1$.

5.1 FIXED FITNESS

In order to validate the different assumptions in the theory, we start by examining runs where the fitness influence is eliminated. That is, each time a classifier is updated, its fitness is not updated as usual but is simply set to its numerosity (completely eliminating the fitness influence in selection). The same is done in covering. Moreover, we eliminated the distinct deletion pressure influences by deleting classifiers proportionally to their numerosity num regardless of their value of as or F. This section investigates the influence caused by the two mutation types. Moreover, we investigate the influence of the GA threshold θ_{ga} as well as the influence of an initialization of the population.

With the restrictions, the runs exactly match the theory as shown in figure 2. The initial specificity of 0.5 drops off quickly in the beginning due to the strong *set pressure*. However, soon the effect of mutation becomes visible and the specificity in the population converges as predicted. The higher the mutation rate μ, the stronger the influence of mutation pressure, which is manifested in the higher convergence value in the curves with higher μ.

Although the mutation pressure becomes visible in the variation of μ, figure 3 further reveals the influence caused by mutation. As formulated in equation 8 the mutation pressure is slightly higher when applying the free mutation type. When directly comparing figure 2 and figure 3 one can observe that the higher the parameter μ, the higher the difference in the mutation pressure.

As stated earlier, we set the GA threshold θ_{ga} to one to assure a GA frequency f_{ga} of one. When altering θ_{ga} setting it to the common value of 25 figure 4 reveals what has been suspected before. For equation 9 to exactly match the specificity change, f_{ga} cannot be denoted by a constant but is actually dependent on

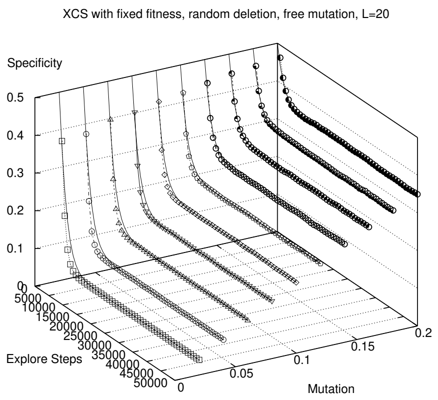

Figure 3: As predicted is the pressure caused by free mutation higher than the one by niche mutation.

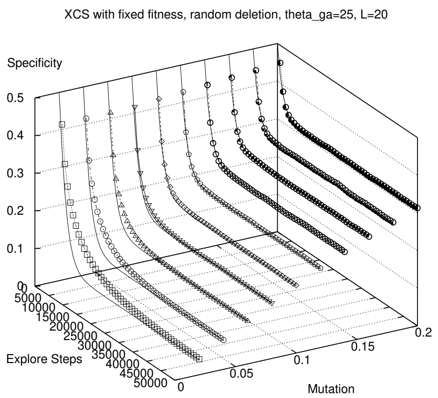

Figure 4: When increasing the threshold θ_{ga} the GA frequency and consequently the specificity pressure decreases once the specificity drops.

the current specificity in the population. Once the specificity in the population has dropped, the action set sizes increase since more classifiers match a specific state. Consequently, more classifiers take part in a GA application, more time stamps ts are updated, the average time since the last GA application in the population and in the action sets drops, and finally the GA frequency drops which is observable in the graph. However, as predicted by equation 9, despite its dependence on the actual specificity, f_{ga} does not influence the convergence value.

Although we decided to initially fill up the population with classifiers to assure a perfect binomial specificity distribution in the beginning of the run, this appeared not to be necessary as shown in figure 5. The figure shows runs in which the population is initially empty.

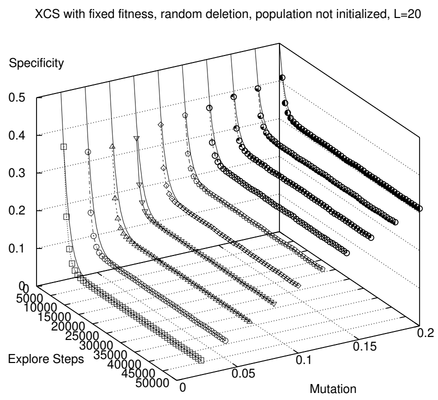

Figure 5: Without initializing the population, the generality drops slightly faster in the beginning.

The only effect observable is that the specificity drops off slightly faster in the beginning of a run. Since the population does initially not contain 2000 classifiers, the generality pressure is stronger which is also expressed in equation 9.

5.2 CONSTANT FUNCTION

While the fitness influence was intentionally eliminated above, this and the next section are dedicated to determine the actual fitness influence. This section applies XCS to a constant boolean function which always returns a reward of 1000. The result is that all classifiers are accurate since the prediction error will be zero. However, an influence could be possible due to the fitness determination according to the relative accuracy.

Figure 6 exhibits that this influence can be neglected when the random deletion method is applied as before. It shows that the assumption of a binomial distribution indeed holds later in the run or is at least not too harsh since the specificity exactly behaves as predicted.

When applying the usual deletion method in XCS, however, the behavior of the specificity changes. Figure 7 shows that the slope of the curves decreases. In the end, though, the convergence value is reached what is predicted by the theory. Since the observable influence can only be caused by the bias of the deletion method towards deleting classifiers in larger niches, the reason for this behavior follows. Since the specificity drops, the action set size increases as noted before. Thus, since more general classifiers are more often present in action sets, their action set size estimate as is more sensitive to the change in the action set size and consequently, it is larger in more general classifiers while specificity drops. Eventually, all as

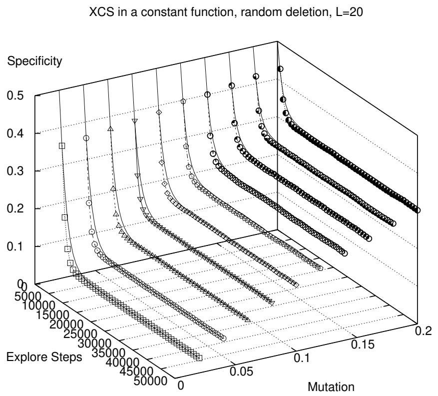

Figure 6: When applied to a constant function, the changing specificity still matches the proposed theory.

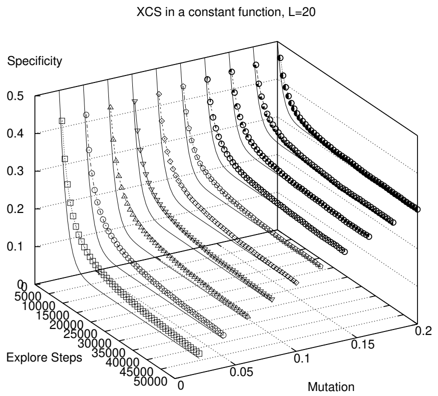

Figure 7: As expected, the effect caused by the deletion method is minor.

values will have adjusted to the change and the predicted convergence value is achieved. This proposition is further validated by the fact that the difference in the runs and the theory are smaller and become equal faster with a higher mutation rate μ since the specificity slope is not as steep as in the curves with lower μ values.

5.3 RANDOM FUNCTION

While the fitness influence remained small in the case of a constant function, much more noise is introduced when applying XCS to a random function. The final two curves reveal the behavior of XCS in a random boolean function that randomly returns rewards of 1000 and 0.

Figure 8 exhibits that in the case of a random function the fitness influences the specificity slope as well as the convergence value. The convergence takes longer and,

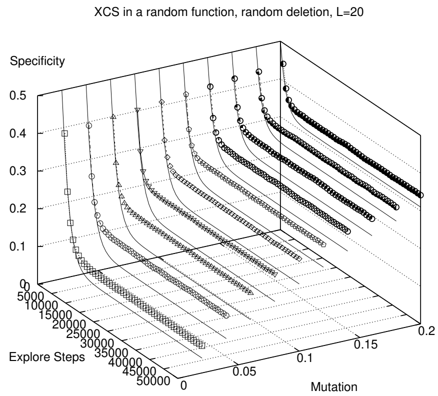

Figure 8: Applied to a random function, the specificity stays on a higher level due to the stronger noise in more specific classifiers.

moreover, the convergence value stays larger. Since again random deletion is applied, the fitness pressure is the only possible influence. Although we don't have a proof in hand we believe the following. Since the encountered rewards are 0 and 1000, the reward prediction of the classifiers fluctuates around 500 and consequently the prediction error around 500 as well. As in the case of the more sensitive action set size estimates above, here the sensitivity manifests in the prediction error ϵ. More specific classifiers have a less sensitive ϵ and consequently a higher variance in the ϵ values. This by itself does not cause any bias, however, since the accuracy calculation expressed in equation 4 scales the prediction error to the power ν, which is set to the usual value five, the higher variance causes an on average higher fitness.

When applying the normal deletion method, the deletion method causes a further increase in specificity as shown in figure 9. Since the specificities do not converge to those in figure 8, the cause must lie in the bias towards deleting experienced, low-fit classifiers. Interestingly, this pressure is strongest when μ is set to approximately 0.1. A more detailed analysis showed that this is the case because the variance in the fitness values is high as well as the more general classifiers are sufficiently experienced so that their possible low fitness value may be considered during deletion. When the specificity drops further due to a lower μ, the variance in fitness decreases significantly and the effect diminishes. On the other hand, when increasing μ further, the average experience in the population decreases under the crucial value of $\theta_{ga} = 20$ and consequently, the additional bias towards low-fit classifiers applies decreasingly often.

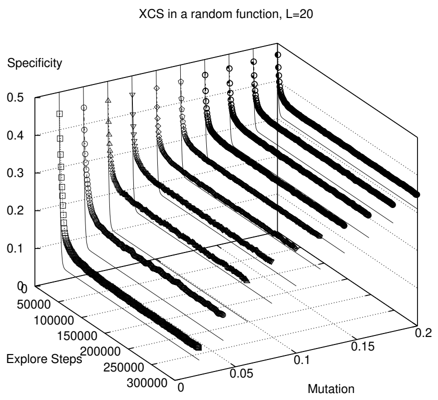

Figure 9: The fitness biased deletion method further influences the specificity.

6 SUMMARY AND CONCLUSION

This paper has investigated various evolutionary pressures in XCS. By analyzing the pressures separately and next investigating their interactions, we were able to derive a formula that can predict the specificity change in the population of XCS. While this change has been hypothesized long ago, we are now able to confirm the hypothesis mathematically and use the derived formula to explain and predict the behavior of the population in XCS. Although the fitness influence is not incorporated in the formula we showed that the formula is applicable in the case of all accurate and all similarly inaccurate. Essentially, the formula can predict how the specificity in a population will evolve once accurate but possibly over-specific classifiers are found. Moreover, it can also predict how the population evolves if there are only inaccurate classifiers and the fitness pressure towards accuracy from the over-general side is very weak.

A final important insight is that regardless of the initial specificity introduced by the *don't care probability* $P_{\#}$, we now know how the specificity changes and to which value it converges. Future research should use this insight and proceed to control the changes in specificity where necessary.

ACKNOWLEDGMENTS

This work was sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant F49620-00-1-0163. Research funding for this work was also provided by the National Science Foundation under grant DMI-9908252. Support was also provided by a grant from the U. S. Army Research Laboratory under the Federated Laboratory Program, Cooperative Agreement DAAL01-96-2-0003. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research, the National Science Foundation, the U. S. Army, or the U. S. Government.

References

Butz, M. V., Kovacs, T., Lanzi, P. L., & Wilson, S. W. (2001). How XCS evolves accurate classifiers. *Submitted to the Genetic and Evolutionary Computation Conference (GECCO-2001).*

Butz, M. V., & Wilson, S. W. (2001). An algorithmic description of XCS. *In Lanzi, P. L., Stolzmann, W. and Wilson, S. W. (Eds.), Advances in Learning Classifier Systems, LNAI 1996*. to appear.

Kovacs, T. (1997). XCS Classifier System Reliably Evolves Accurate, Complete, and Minimal Representations for Boolean Functions. In Roy, Chawdhry, & Pant (Eds.), *Soft Computing in Engineering Design and Manufacturing* (pp. 59–68). Springer-Verlag, London.

Kovacs, T. (1999). Deletion schemes for classifier systems. In Banzhaf, W., Daida, J., Eiben, A. E., Garzon, M. H., Honavar, V., Jakiela, M., & Smith, R. E. (Eds.), *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99)* (pp. 329–336). San Francisco, CA: Morgan Kaufmann.

Widrow, B., & Hoff, M. (1960). Adaptive switching circuits. *Western Electronic Show and Convention*, *4*, 96–104.

Wilson, S. W. (1995). Classifier fitness based on accuracy. *Evolutionary Computation*, *3*(2), 149–175.

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