New Crossover Operator for Evolutionary Rule Discovery in XCS

Sergio Morales-Ortigosa, Albert Orriols-Puig, and Ester Bernadó-Mansilla
Grup de Recerca en Sistemes Intel·ligents
Enginyeria i Arquitectura La Salle
Universitat Ramon Llull
Quatre Camins 2, 08022, Barcelona (Spain)

Abstract

XCS is a learning classifier system that combines a reinforcement learning scheme with evolutionary algorithms to evolve rule sets on-line by means of the interaction with an environment. Usually, research conducted on XCS has mainly focused on the analysis and improvement of the reinforcement learning component, overlooking the evolutionary discovery process to some extent. Recently, the first efforts towards analyzing and designing new operators for the evolutionary algorithm have been done. The selection pressure produced by different selection schemes has been studied and the rule representation of XCS has been extended to adapt evolution strategies as the discovery component of the system. This paper continues on the analysis of the evolutionary algorithms in the on-line architecture by analyzing the role of the crossover operator in the original XCS and XCS based on evolution strategies. A new recombination operator, inspired by the BLX crossover operator in real-coded genetic algorithms, is designed for XCS. The new recombination operator is experimentally compared with the traditional crossover operator of XCS on a collection of real-life classification problems. The results show the competence of the new operator, providing the best results, on average, on the tested domains.

1. Introduction

Learning Classifier Systems (LCSs) [11] are complex machine learning methods that combine reinforcement learning techniques for rule evaluation with genetic algorithms for rule discovery to evolve a distributed set of solutions[1] on-line. Recently, XCS [19, 20], by far the most influential LCSs, has received an special amount of attention since it has been shown to be able to solve a wide variety of learning problems such as classification problems, reinforcement learning problems, unsupervised learning problems, and function approximation problems. In the present work we focus on classification problems, but the conclusions could also be extended to other types of tasks.

Research conducted on XCS during the last few years has been especially focused on three aspects: (i) designing new reinforcement learning mechanisms to obtain better rule evaluations [4], (ii) proposing modifications to the architecture to deal with new types of problems [22], and (iii) adapting new representations to deal with real-life problems [15]. Nonetheless, the study of the evolutionary discovery process in XCS, and especially, the role of the different genetic operators, has received little attention. Only few analyses of selection operators [6, 16], reproduction based on linkage extraction [5], and self-adaptive mutation [12], all of them for problems with binary attributes, can be found in the literature. Nonetheless, research on designing new crossover and mutation operators is very scarce. Oppositely, in the field of continuous optimization, many analysis of the effect of different operators have been performed, which resulted in the design of new enriched genetic operators [9].

The present work is a follow-up of the work in [14] in which the authors proposed to analyze the effect of the different genetic operators in XCS. For this purpose, a new discovery process based on evolution strategies (ESs) [17] was designed. Then, XCS with both types of evolutionary schemes were analyzed, especially focusing on the role of mutation. The analysis was crucial for both incrementing our understanding of how mutation works and improving the performance in real-life problems. We now continue this work by studying the crossover operator in XCS. We propose a new crossover scheme, addressed as BLX crossover, which is inspired by the BLX crossover operator of RCGA [9] but adapted to deal with the interval-based representation of XCS. The performance obtained when using the new crossover operator in XCS with both ESs and GAs is tested on a large collection of real-world problems.

The remainder of this paper is organized as follows. Section 2 gives a brief description of XCS and its extension to

[1] Usually, solutions are represented as interval-based rules

evolution strategies. Section 3 describes in detail the new BLX crossover operator and Sect. 4 provides experimental methodology used in the experimentation. Section 5 presents and discusses the results. Finally, Section 6 summarizes, concludes, and discusses future work lines.

2. The XCS Classifier System

XCS [19] is a Michigan-style LCS that combines reinforcement learning techniques [18] with genetic algorithms [10] to learn a distributed set of sub solutions online. In [14], the component discovery of the system, driven by a genetic algorithm, was replaced with an evolution strategy. As follows, we provide a brief description of XCS, and then, we explain the extension to evolution strategies. For sake of clarity, we refer to XCS driven by genetic algorithms as XCS-GA and to XCS driven by evolution strategies as XCS-ES. For an algorithmic description the reader is referred to [7].

2.1. Knowledge Representation

XCS-GA evolves a population [P] of classifiers that consist of a production rule that takes the following form: **if** $x_1 \in [l_1, u_1] \wedge \ldots x_\ell \in [l_\ell, u_\ell]$ **then** *class*. That is, each variable of the condition x is represented by an interval $[\ell_i, u_i]^\ell$ (ℓ is the input length). Then, a rule matches an input instance $e = (e_1, e_2, \ldots, e_\ell)$ if $\forall_i \ell_i \leq e_i \leq u_i$.

Each classifier has four main parameters: (i) the payoff prediction p, an estimate of the reward that the system will receive if the class of the rule is selected as output, (ii) the prediction error ϵ, which estimates the error of the payoff prediction, (iii) the fitness F, which is computed as and inverse function of the prediction error, and (iv) the action set size as, an estimate of the average size of the action sets in which the classifier has participated.

2.2. Learning Interaction

At each learning iteration, XCS-GA receives a new training example $e = (e_1, e_2, \ldots, e_\ell)$ and the system creates a *match set* [M], which consists of all classifiers in [P] whose condition matches e. The next step depends on whether the system is in exploration (training) mode or in exploitation (test) mode. In exploration mode, the system randomly chooses one of the possible classes and builds the action set [A] with all the classifiers in [M] that advocate the selected class. The parameters of all the classifiers in [A] are updated according to a generalized version of Q-learning (see [19]). In exploitation mode, the classifiers in [M] vote, according to their fitness, for the class they predict. The most voted class is selected as output.

2.3. Discovery Component

XCS-GA applies a steady-state niche genetic algorithm (GA) to discover new promising rules. The GA is triggered on [A] when the average time since the last application of the GA to the classifiers in [A] exceeds a certain threshold θ_{GA}. Then, the system selects two parents from [A]. So far, two selection schemes have been studied: *proportionate selection* [19], in which each classifier in [A] has a probability proportional to its fitness to be chosen; and *tournament selection* [6], in which tournaments are held among a set of randomly selected classifiers, and the best classifier of the tournament is chosen as a parent. Then, the parents are crossed and mutated with probabilities χ and μ respectively. Crossover shuffles the condition of the two parents by cutting the chromosomes by two points. Mutation decides whether each variable has to be changed; in this case, it adds a random amount to the lower or to the upper bound of the variable interval. Each offspring is introduced in the population, removing a low-fit classifier if the population is full. The deletion probability of a classifier is proportional to the size of the action sets where the classifier has participated and inversely proportional to its fitness [13]. This biases the search toward highly fit classifiers, and at the same time balances the classifiers' allocation in the different action sets.

2.4. From GA to ES in XCS

In XCS-ES [14], the classifier representation is enriched with a vector of strategy parameters $s = (s_1, s_2, \ldots, s_\ell)$, which are used to adapt the intervals of the variables of the rule's condition (the rule's condition is referred to as *object parameters* in ESs terms). The selection, crossover, and mutation operators are also re-defined according to the new representation. Three selection schemes are considered by XCS-ES: proportionate selection [19], tournament selection [6], and truncation selection [17]. Two kind of crossover operators are considered: (i) discrete/dominant recombination and (ii) intermediate recombination. Discrete recombination produces a new rule where each variable and strategy parameter is selected from one of the parents. Intermediate recombination calculates the center of mass of the parents. Finally, mutation is redefined as follows. First, the intervals of each rule variable x_i are mutated as $x_i = x_i + z$, where $z = (s_1 N_1(0,1), s_2 N_2(0,1), \ldots, N_\ell(0,1))$ are independent random samples from Gaussian normal distribution).

3. New BLX Crossover

In [14], the authors experimentally showed that the exploratory capabilities of Gaussian mutation enabled XCS-

ES to achieve better accuracy ratios than XCS-GA when only the mutation operator was considered. This was because XCS-ES mutation was able to guide the search more accurately. That is, XCS-ES mutation was able to not only perform local search but also to innovate, i.e., explore new regions of the feature space (in GAs, innovation is usually provided by the crossover operator). Inspired by this later aspect and based on the BLX-α operator of RCGAs [9], here we design a new crossover operator for XCS, which we address as BLX crossover. This new operator is meant to be more sensitive in combining the information of the parents, searching a balance between local search and exploration of new regions of the feature space. The operator is defined for both XCS-GA and XCS-ES. As follows, the new operator is described in detail for each case.

3.1. BLX Crossover for XCS-GA

Given two parent classifiers $p_1 = (a_1^{p_1}, a_2^{p_1}, \ldots, a_\ell^{p_1})$ and $p_2 = (a_1^{p_2}, a_2^{p_2}, \ldots, a_\ell^{p_2})$, where each variable is represented by the interval of feasible values $a_i^{p_i} = [\ell_{a_i}^{p_i}, u_{a_i}^{p_i}]$ the BLX crossover operator generates the interval for each variable i the two offspring classifiers o_1 and o_2 as follows. First, a random value α ranging in [0, 0.5] is generated. Then, the minimum lower bound c_{min} and the maximum upper bound c_{max} of the two parents is computed, i.e., $c_{min}^i = min(\ell_{a_i}^{p_1}, \ell_{a_i}^{p_2})$ and $c_{max}^i = max(u_{a_i}^{p_1}, u_{a_i}^{p_2})$. The distance between c_{max}^i and c_{min}^i is calculated: $I = c_{max}^i - c_{min}^i$. Finally, I is used to generate the lower bound and the upper bound of the variable i of offspring o_1 and o_2, that is,

$$\ell_{a_i}^{o_1} = c_{min} + \alpha \cdot I \cdot rand(\{-1, 1\}) \tag{1}$$
$$\ell_{a_i}^{o_2} = c_{min} + (1-\alpha) \cdot I \cdot rand(\{-1, 1\}) \tag{2}$$
$$u_{a_i}^{o_1} = c_{max} + \alpha \cdot I \cdot rand(\{-1, 1\}) \tag{3}$$
$$u_{a_i}^{o_2} = c_{max} + (1-\alpha) \cdot I \cdot rand(\{-1, 1\}) \tag{4}$$

where $rand(\{-1, 1\})$ returns either -1 or 1 with the same probability. Note that a low value of α means that each offspring will be very similar to one of the parents, performing a type of local search; oppositely, a large value of α will result in different offspring, probably exploring new regions of the feature space. For this reason, we consider that this type of mutation balances local search and innovation.

3.2. BLX Crossover for XCS-ES

In XCS-ES, a vector of strategy parameters is added to the original classifier representation of XCS-GA. Therefore, BLX crossover takes two parents represented by the pair condition r_i and strategy parameters s_i, i.e., $p_1 = (r_1, s_1)$ and $p_2 = (r_2, s_2)$, where $r_i = (a_1^{p_i}, a_2^{p_i}, \ldots, a_\ell^{p_i})$ and $s_i = (\sigma_1^{p_i}, \sigma_2^{p_i}, \ldots, \sigma_\ell^{p_i})$, and generates two children by crossing both the information contained in the condition and the information of the strategy parameters. The conditions r_i are crossed according to the scheme proposed in the previous section. Then, for each child o_i, a new vector of strategy parameters $s_i = (\sigma_1^{o_i}, \sigma_2^{o_i}, \ldots, \sigma_\ell^{o_i})$ is generated. $\sigma_j^{o_i}$ is a random number ranging in $[c_{min}^i - I_i\alpha, c_{max}^i + I_i\alpha]$, where $c_{min}^i = min(\sigma_i^{p_1}, \sigma_i^{p_2})$, $c_{max}^i = max(\sigma_i^{p_1}, \sigma_i^{p_2})$, and $I_i = c_{min}^i, c_{max}^i$.

4. Experimental Methodology

For the study we used a collection of 12 real-life data sets: *balance-scale* (bal), *bupa* (bpa), *glass* (gls), *heart-s* (h-s), *iris* (irs), *pima* (pim), *tao* (tao), *thyroid* (thy), *vehicle* (veh), *Wisconsin breast-cancer data base* (wbcd), *Wisconsin diagnose breast-cancer* (wdbc), and *wine* (wne). All the data sets where extracted from the UCI repository [1], except for tao, which was extracted from a local repository [2]. The different configurations of XCS-GA and XCS-ES were ran on these data sets and the quality of the results was compared in terms of the performance (test accuracy). To obtain reliable estimates of this metric we used a ten-fold cross-validation procedure. XCS was configured as follows (see [7] for notation details): $iter.$ = 100,000, N = 6400, θ_{GA} = 50, χ = 0.8, μ = 0.04, $r0$ = 0.6, $m0$ = 0.1.

We statistically analyzed the performance of each learner following the procedure pointed out in [8]. We first applied the multi-comparison Friedman test to contrast the null hypothesis that all the learning algorithms performed the same on average. If the null hypothesis was rejected, the post-hoc Bonferroni-Dunn test was used. Furthermore, we complemented the statistical analysis performing pairwise comparisons by means of the non-parametric Wilcoxon signed-ranks test.

5. Experimental Results

Our first concern was to compare the performance of XCS-GA and XCS-ES with the BLX crossover operator and the different selection schemes defined for XCS; besides, we aimed at analyzing whether the new crossover operator permitted to improve the performance obtained with original XCS as defined in [19]. For this purpose, Table 1 provides the test performance of XCS-GA and XCS-ES with the new BLX crossover operator (see from the 2nd to the 5th column of the table). As we are interested in analyzing the interaction of the different genetic operators, we ran the experiments with the two most used selection schemes in XCS: proportionate selection (ps) [19] and tournament selection (ts) [6]. Moreover, we also used truncation selection for XCS-ES, since it is a selection operator widely used in the ESs realm. Lastly, we add the results obtained with the original definition of XCS-GA [19, 21], that is, XCS-GA with proportionate selection and two-point crossover, with the aim of comparing the improvements provided by the new crossover operator with respect to the original XCS (6th column of the table). The last two rows of the table supply the average rank of each learning technique and its position in the ranking.

The empirical results show that the three best ranked methods in the comparison correspond to configurations that use BLX crossover. More specifically, XCS-GA with the both types of selection and XCS-ES with tournament selection outperformed the original XCS scheme.This highlights the suitability of the new crossover operator. Note, moreover, that all the schemes of XCS-GA with BLX crossover have a better average rank than the schemes of XCS-ES with BLX crossover. This indicates that the GA schemes benefited more than the ES schemes from the new BLX crossover operator.

We statistically compared the test performance of the six learning methods. The multi-comparison Friedman's test permitted to reject the null hypothesis that all the learners were statistically equivalent at alpha = 0.10. Nonetheless, the post-hoc Bonferroni test could not detect any significant difference between the best ranked learner, i.e., XCS-GA with tournament selection and any other learning technique. Therefore, we applied a pairwise analysis by means of the Wilcoxon signed-ranks test at $\alpha = 0.10$ to detect further differences. It is well known that pairwise comparisons increase the risk of rejecting null hypotheses that are actually true. Herein, we assume this risk with the aim of providing more information about the excellence of the different methods. The pairwise analysis detected that (i) XCS-GA with tournament selection generated models that were significantly more accurate than those created by XCS-ES with proportionate and truncation selection and (ii) XCS-ES with tournament selection outperformed XCS-ES with proportionate selection. No further significant differences were identified by the statistical analysis.

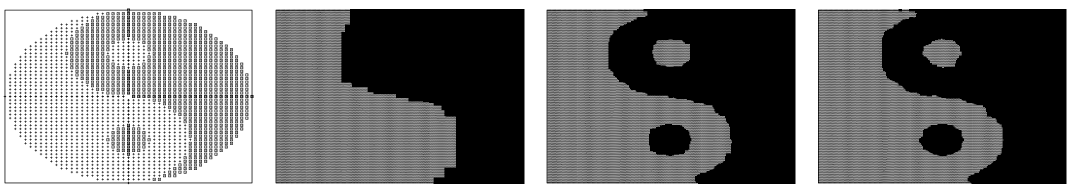
We further studied the behavior of the new crossover operator by analyzing the differences in each particular training data set with the aim of providing a glimpse of under which problem characteristics BLX crossover performed the best. We identified that the new operator yielded excellent results in dense problems, i.e., problems with a large number of instances with respect to the number of variables. Notice, for example, the excellent results obtained by all the configurations using BLX in the *bal* and the tao problems with respect to XCS-GA with two point crossover. To our knowledge, these are, by far, the best results obtained by XCS with these two problems [2, 3]. We hypothesized that this was because the new operator was able to fit complex decision boundaries more accurately since it has a less disruptive behavior than the two point crossover operator. To confirm this hypothesis, we did the following experiment. We ran different configurations of XCS-ES on the tao problem (see the domain in figure 1(a)) and plotted the decision boundaries of the evolved rules sets. The decision boundaries where plotted by testing the resulting rule set with a dense test set that contained instances equally distributed around the feature space and depicting with different colors these instances depending on the class predicted by the system. We selected the tao problem for this analysis since it consists of two variables, and so, the decision boundaries can be easily illustrated. Besides, tao has curved decision boundaries that pose a big challenge for interval-based LCSs [2]. Figure 1(b) shows the decision boundaries obtained with XCS-GA with two point crossover. Figures 1(c) and 1(d) show the same information but for XCS-GA and XCS-ES with BLX crossover. Note that XCS-GA with two point crossover evolved models that obviated the two inner concepts of the tao problem; moreover, the decision boundary defined by the system presented an abrupt shape, concentring a big amount of the test error. Oppositely, when BLX crossover was used, XCS-GA and XCS-ES were able to define very accurate class boundaries and discover the two inner concepts of the problem. It is worth noting that, this specificity-driven pressure that seems to be present in the BLX crossover operator does not go in detriment of the test accuracy in problems where the training instances are more sparse, as shown in the general results presented in Table 1.

Finally, we also detected that the BLX crossover operator prevented the system from over-fitting in complex problems. That is, we identified that XCS-GA with two-point crossover tended to over-fit the training instances, i.e., create rules that cover few instances, with the aim of maximizing the training accuracy in complex problems. To show this, Figure 2 shows the evolution of the training and the test performance achieved by XCS-GA with two-point crossover and the new BLX crossover in the *bal* problem. Note that, with two-point crossover, the training accuracy increased during the 100,000 learning iterations, but the test accuracy started to decrease at about 10,000 iterations. This indicates that XCS-GA with two-point crossover was over-fitting the training instances in this particular problem. On the other hand, the results denote that the new BLX crossover operator prevented XCS from over-fitting. Notice that the test accuracy remained oscillating around 86%. This behavior was observed in other domains such as the tao problem (the figures are not shown due to space limitations).

The overall results presented in the section showed the suitability of the new crossover operator with respect to two point crossover, the original recombination operator used in XCS with interval-based representation. We showed the BLX crossover provided the best average behavior in com-

Table 1. Comparison of test accuracy achieved by XCS-GA and XCS-ES with BLX crossover and proportionate selection (ps), tournament selection (ts), and truncation selection (tr). The last column provides the results of the original definition of XCS, i.e., XCS-GA with two point crossover (tp) and proportionate selection. The last two rows supply the average rank of each learner and its position in the ranking.

Data set	XCS-GA ps	XCS-ES ps	XCS-GA ts	XCS-ES ts	XCS-ES tr	XCS-GA tp
bal	86.29 (1)	85.39 (5)	85.44 (4)	85.55 (2.5)	85.55 (2.5)	83.20 (6)
bpa	69.18 (1)	65.89 (6)	68.11 (3.5)	68.11 (3.5)	66.76 (5)	68.21 (2)
gls	69.94 (5)	70.25 (4)	71.65 (2)	71.49 (3)	69.16 (6)	72.12 (1)
h-s	39.51 (4)	38.64 (6)	38.76 (5)	39.88 (3)	43.21 (2)	46.91 (1)
irs	95.11 (4)	95.11 (4)	95.33 (1.5)	95.11 (4)	94.89 (6)	95.33 (1.5)
pim	75.22 (1)	74.43 (4)	74.65 (2.5)	73.83 (5)	74.65 (2.5)	72.53 (6)
tao	96.80 (1)	96.77 (2)	96.68 (3)	96.61 (4)	96.57 (5)	91.22 (6)
thy	97.36 (1)	96.90 (3)	97.05 (2)	96.43 (5)	96.74 (4)	95.81 (6)
veh	69.34 (4)	69.23 (6)	69.74 (3)	70.21 (2)	69.31 (5)	71.79 (1)
wbcd	95.66 (5)	96.28 (3)	96.33 (1.5)	95.99 (4)	96.33 (1.5)	94.85 (6)
wdbc	91.04 (5)	91.33 (3)	91.62 (2)	91.97 (1)	90.63 (6)	91.09 (4)
wne	95.88 (4)	96.07 (3)	96.25 (2)	96.82 (1)	95.13 (6)	95.50 (5)
Rank	*3.00*	*4.08*	*2.67*	*3.17*	*4.29*	*3.79*
Pos	*2*	*5*	*1*	*3*	*6*	*4*



(a) Domain (b) XCS-GA 2pt (c) XCS-GA BLX (d) XCS-ES BLX

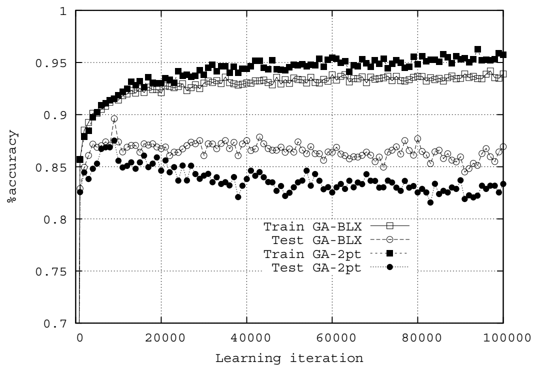
Figure 1. Decision boundaries obtained with original scheme of XCS-GA with (b) two point crossover, (c) XCS-GA with BLX crossover, and (d) XCS-ES with BLX crossover in the (a) tao problem.

bination with XCS-GA and XCS-ES; moreover, we also illustrated the advantages provided by this scheme in problems with a large number of instances per dimension and complex boundaries.

6. Summary, Conclusions, and Further Work

In this paper, we analyzed the genetic discovery in the XCS classifier system. The present work is a follow-up of the work presented in [14], in which a new genetic procedure based on evolution strategies was introduced into XCS, analyzing in detail the effect of different types of mutation. We now considered the last genetic operator: crossover. We designed a new crossover operator inspired by one of the most competitive crossover schemes in RCGA: the BLX crossover operator. We experimentally showed the advantages of the new operator and compared it with two-point crossover.

The observations provided in this work suppose the first steps toward a better understanding of how the genetic operators influence the search in XCS, an aspect that has been overlooked for a long period. The results clearly showed that XCS could benefit from new recombination schemes that exploit the characteristics of each particular problem. For example, we noticed the improvement that the new operator provided in the *bal* and *tao* problems; on the other hand, this improvement was not present in other problems such as *h-s*, in which the original XCS scheme provided the best results. This supports the idea that more research needs to be conducted on designing new operators that are more efficient for a set of problem characteristics. Moreover, it also makes evident that it is crucial to characterize the learning problems and to decide which set of operators should be used depending on each particular problem.



(a) bal

Figure 2. Evolution of the training and test accuracy of XCS-GA with two-point crossover and BLX crossover in the bal problem.

Therefore, as further work, we will study different strategies to extract characteristics from the training data sets and link these characteristics to the type of operators used during search with the aim of designing hyper-heuristics that enable the system to self-tune its operators depending on the apparent complexity of each particular problem.

Acknowledgements

The authors would like to thank the *Ministerio de Educación y Ciencia* for its support under project TIN2005-08386-C05-04 and *Generalitat de Catalunya* for its support under grants 2005FI-00252 and 2005SGR-00302.

References

[1] A. Asuncion and D. J. Newman. *UCI Machine Learning Repository: [http://www.ics.uci.edu/∼mlearn/ MLRepository.html]*. University of California, 2007.

[2] E. Bernadó-Mansilla and J. Garrell. Accuracy-based Learning Classifier Systems: Models, analysis and applications to classification tasks. *Evolutionary Computation*, 11(3):209–238, 2003.

[3] M. V. Butz. *Rule-based evolutionary online learning systems: A principled approach to LCS analysis and design*, volume 109 of *Studies in Fuzziness and Soft Computing*. Springer, 2006.

[4] M. V. Butz, D. E. Goldberg, and P. L. Lanzi. Gradient descent methods in learning classifier systems: Improving XCS performance in multistep problems. *IEEE-TEC*, 9(5):452–473, 2005.

[5] M. V. Butz, M. Pelikan, X. Llorà, and D. E. Goldberg. Automated global structure extraction for effective local building block processing. *Evolutionary Computation Journal*, 14(3):345–380, 2006.

[6] M. V. Butz, K. Sastry, and D. E. Goldberg. Strong, stable, and reliable fitness pressure in XCS due to tournament selection. *GPEM*, 6(1):53–77, 2005.

[7] M. V. Butz and S. W. Wilson. An algorithmic description of XCS. In *Proc. IWLCS*, volume 1996, pages 253–272. Springer, 2001.

[8] J. Demsar. Statistical comparisons of classifiers over multiple data sets. *JMLR*, 7:1–30, 2006.

[9] F. Herrera, M. Lozano, and J. L. Verdegay. Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis. *Artificial Intelligence Review*, 12(4):265–319, 1998.

[10] J. H. Holland. *Adaptation in natural and artificial systems*. The University of Michigan Press, 1975.

[11] J. H. Holland. Adaptation. In R. Rosen and F. Snell, editors, *Progress in Theoretical Biology*, volume 4, pages 263–293. New York: Academic Press, 1976.

[12] J. Hurst and L. Bull. A self-adaptive XCS. In *IWLCS'01: Revised Papers from the 4th International Workshop on Advances in Learning Classifier Systems*, pages 57–73, London, UK, 2002. Springer-Verlag.

[13] T. Kovacs. Deletion schemes for classifier systems. In *GECCO'99: Proceedings of the 1999 Genetic and Evolutionary Computation Conference*, pages 329–336. Morgan Kaufmann, 1999.

[14] S. Morales-Ortigosa, A. Orriols-Puig, and E. Bernadó-Mansilla. Can evolution strategies improve learning guidance in XCS? design and comparison to genetic algorithm based xcs. In *Frontiers in Artificial Intelligence and Applications. Also available at http://www.salle.url.edu/∼aorriols/CCIA08.pdf*. IOS Press, submitted.

[15] A. Orriols-Puig, J. Casillas, and E. Bernadó-Mansilla. Fuzzy-UCS: a Michigan-style learning fuzzy-classifier system for supervised learning. *IEEE-TEC*, in press.

[16] A. Orriols-Puig, K. Sastry, P. Lanzi, D. Goldberg, and E. Bernadó-Mansilla. Modeling selection pressure in XCS for proportionate and tournament selection. In *GECCO'07*, volume 2, pages 1846–1853. ACM Press, 2007.

[17] I. Rechenberg. *Evolution Strategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*. Frommann-Holzboog, 1973.

[18] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press, 1998.

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

[20] S. W. Wilson. Generalization in the XCS classifier system. In *3rd Annual Conf. on Genetic Programming*, pages 665–674. Morgan Kaufmann, 1998.

[21] S. W. Wilson. Get real! XCS with continuous-valued inputs. In *Learning Classifier Systems. From Foundations to Applications*, pages 209–219, Berlin, 2000. Springer-Verlag.

[22] S. W. Wilson. Classifiers that approximate functions. *Journal of Natural Computing*, 1(2):211–234, 2002.