Towards Increasing Learning Speed and Robustness of XCSF: Experimenting With Larger Offspring Set Sizes

Patrick Stalph
Department of Computer Science
University of Würzburg, Germany
Patrick.Stalph@stud-mail.uni-wuerzburg.de

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

ABSTRACT

The XCS classifier system has been successfully applied to various problem domains including datamining, boolean classifications, and function approximation. In all these applications just two classifiers were reproduced in a match or action set, given a time-recency threshold was met in the set. In this paper, we investigate the effect of selecting more than two classifiers for reproduction in XCSF. We either increase the number of selected classifiers or select a number of classifiers relative to the current match set size. In the functions investigated, both approaches showed a highly significant increase in initial learning speed. Also, in less challenging approximation tasks, the final accuracy reached is not affected by the approach. However, in harder functions, learning may stall due to over-reproductions of inaccurate, ill-estimated classifiers. Thus, we propose an adaptive offspring size rate that may depend on the current reliability of classifier parameter estimates. First results with a fixed offspring set size decrement show promising results. Future work is needed to speed-up XCS's learning progress and adjust its learning speed to the perceived problem difficulty.

Categories and Subject Descriptors

F.1.1 [**Models of Computation**]: Self-modifying machines

General Terms

Algorithms, Performance.

Keywords

Learning Classifier Systems, Reproduction, Selection Pressure, XCSF.

1. INTRODUCTION

Learning classifier systems were introduced over thirty years ago [8] as cognitive systems. Over all these years, it has been clear that there is a strong interaction between parameter estimations—be it by traditional bucket brigade techniques [9], the Widrow-Hoff rule [12, 13], or by recursive least squares and related techniques [10, 7]—and the genetic algorithm, whose successful identification and propagation of better classifiers depends on the appropriateness of these estimates. Various control parameters have been used to balance genetic reproduction with the reliability of the parameter estimation, but to the best of our knowledge, there is no study that addresses the estimation problem explicitly and directly.

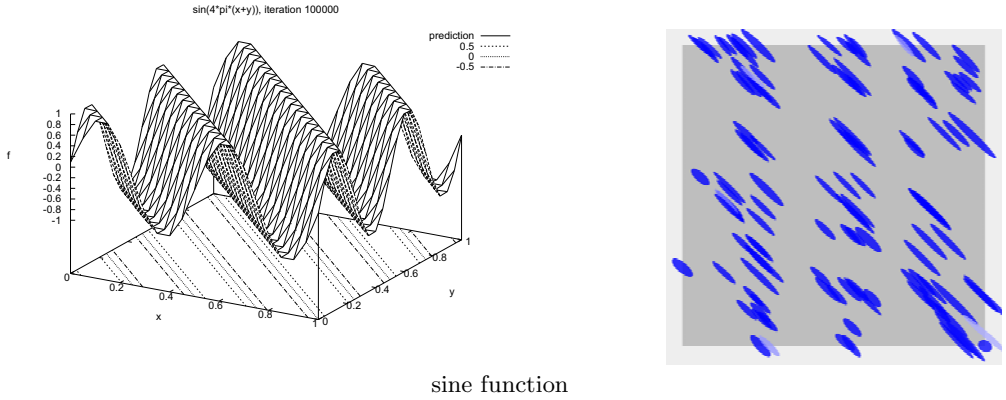
In the XCS classifier system [13], reproduction takes place by means of a steady-state, niched GA. Reproductions are activated in current action sets (or match sets in function approximation problems as well as in the original XCS paper). Upon reproduction, two offspring classifiers are generated, which are mutated and recombined with certain probabilities. Reproduction is balanced by the θ_{GA} threshold. It specifies that GA reproduction is activated only if the average time of the last GA activation in the set lies longer in the past than θ_{GA}. It has been shown that the threshold can delay learning speed but it also prevents forgetting and overgeneralization in the case of unbalanced data sets [11]. Nonetheless, the reproduction of two offspring seems to be rather arbitrary—except for the fact that two offspring classifiers are needed for simple recombination mechanisms. Thus, this study investigates the effect of increasing the number of offspring classifiers generated upon GA invocation. We further focus our study on the real-valued domains and thus the XCSF system [14, 15]. Besides, we use the rotating hyperellipsoidal representation for the evolving classifier condition structures [5].

This paper is structured as follows. Since we assume a general knowledge about XCS, we immediately start investigating performance of XCSF on various test problems and with various offspring sizes. Next, we discuss the results and provide some theoretical considerations. Finally, we propose a road-map for further studying the observed effects and adapting the offspring sizes according to the perceived problem difficulty and learning progress as well as on the estimated reliability of available classifier estimates.

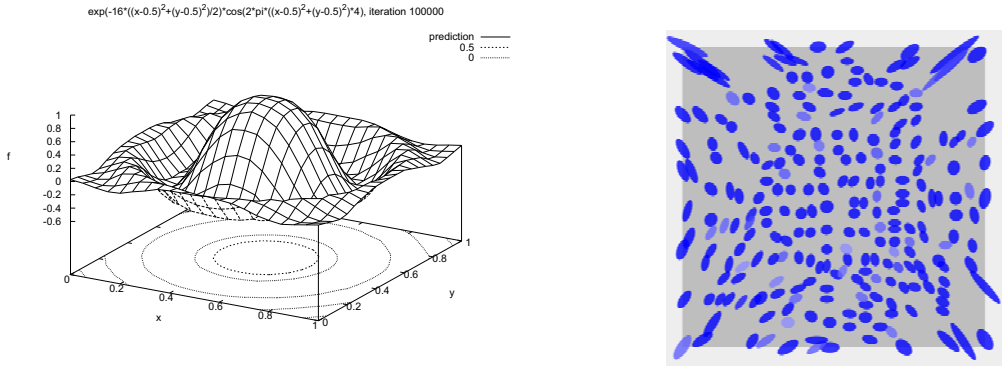
2. INCREASED OFFSPRING SIZES

To study the effect of increased offspring set sizes, we chose four increasingly challenging functions, each with rather dis-

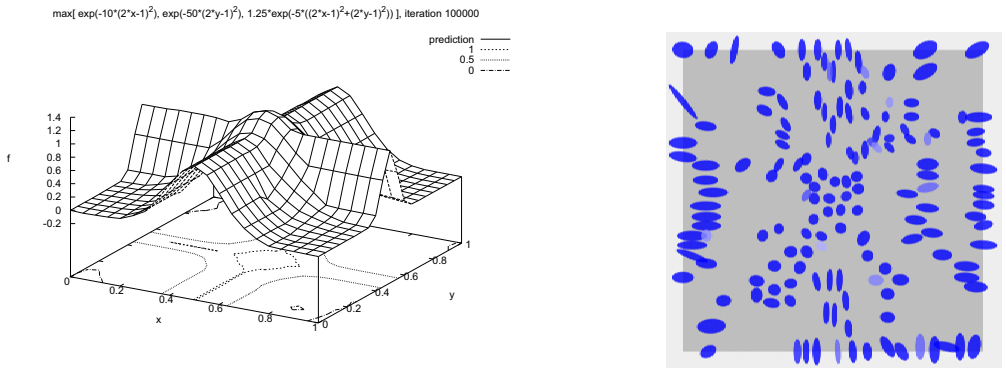
GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.



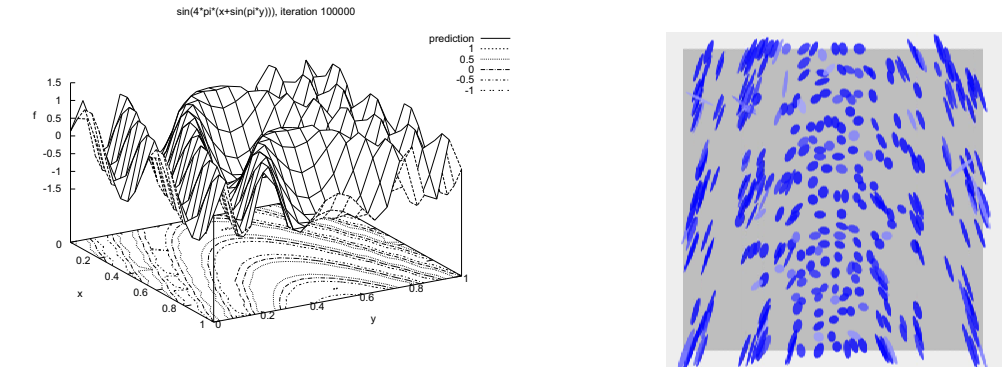
sine function



radial sine function

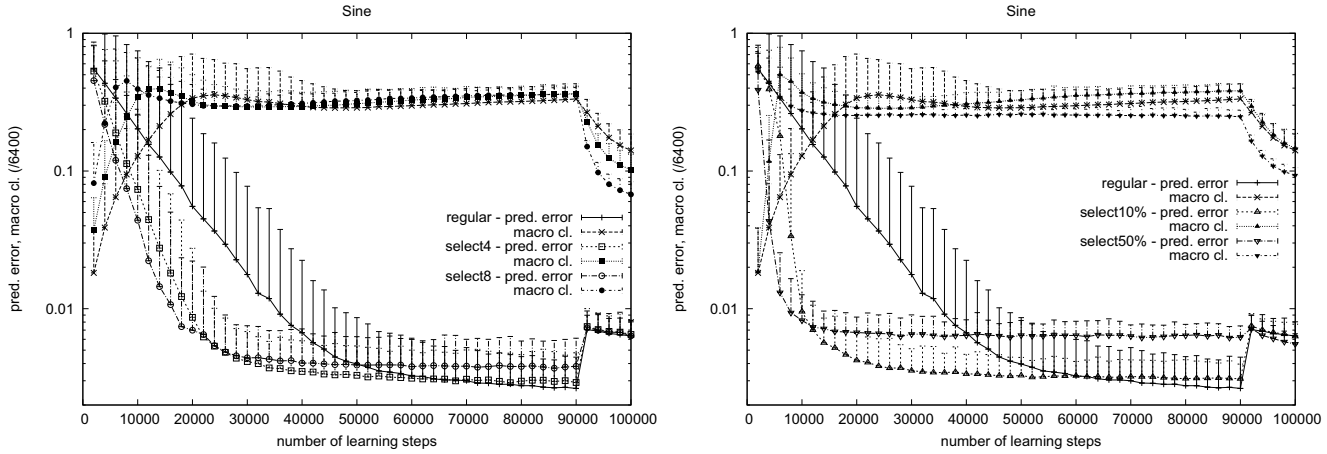


crossed ridge function

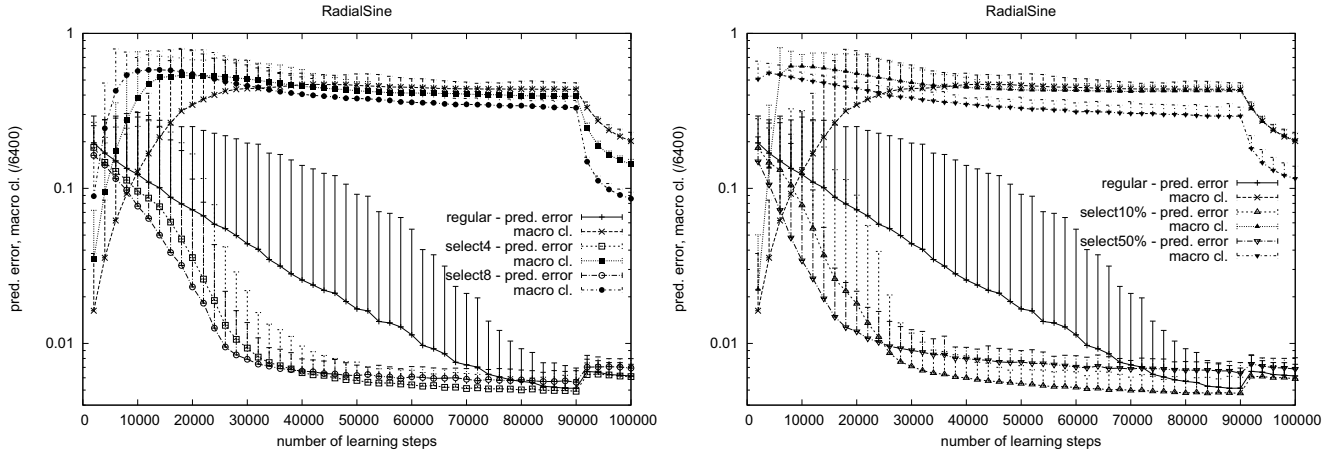


sine-in-sine function

Figure 1: Final function approximations and population distributions after compaction. The conditions visualized are 20% of the actual size.



(a) sine function



(b) radial sine function

Figure 2: Different selection strengths with fixed (left hand side) or match-set-size relative (right hand side) offspring sizes can speed-up learning significantly but potentially increase the final error level reached.

tinct regularities:

$$f_1(x,y) = \sin(4\pi(x+y)) \quad (1)$$

$$f_2(x,y) = \exp^{-16\sum_i (x-.5)^2 \times cos(8\pi \sum_i (x-.5)^2)} \quad (2)$$

$$f_3(x,y) = \max\{e^{-10(2x-1)^2}; e^{-50(2y-1)^2}; 1.25e^{-5((2x-1)^2+(2y-1)^2)}\} \quad (3)$$

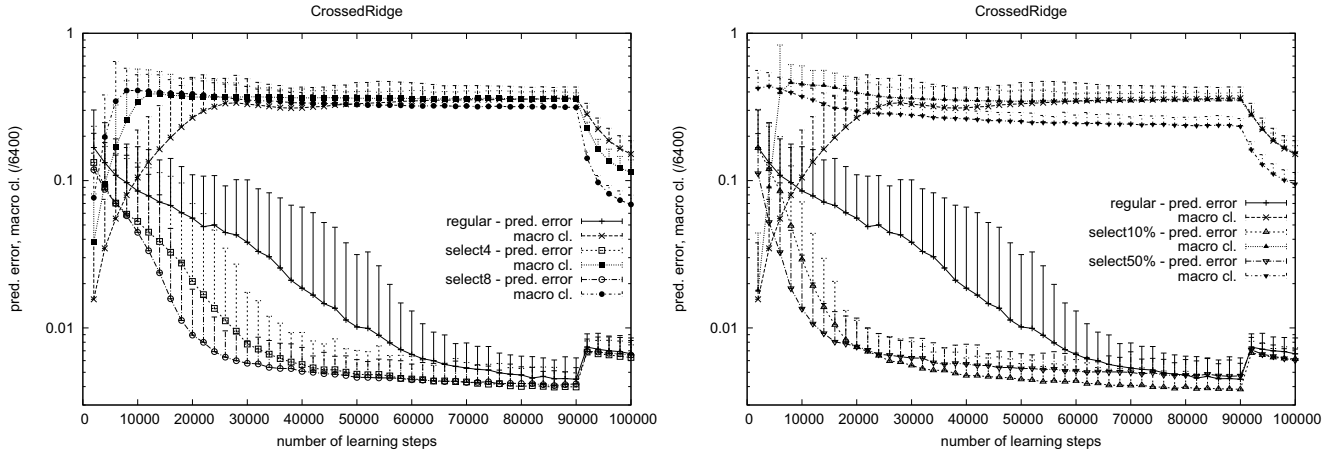
$$f_4(x,y) = \sin(2\pi(x+\sin(\pi y))) \quad (4)$$

Function f_1 has been used in various studies [5] and has a diagonal regularity. It requires the evolution of stretched hyperellipsoids that are rotated by 45°. Function f_2 is a radial sine function that requires a somewhat circular distribution of classifiers. Function f_3 is a crossed ridge function, for which it has been shown that XCSF performs competitively when compared to deterministic machine learning techniques [6]. Finally, function f_4 twists two sine functions so that it becomes very hard for the evolutionary algorithm to receive enough signal from the parameter estimates in order to structure to the problem space effectively for an accurate function value approximation.

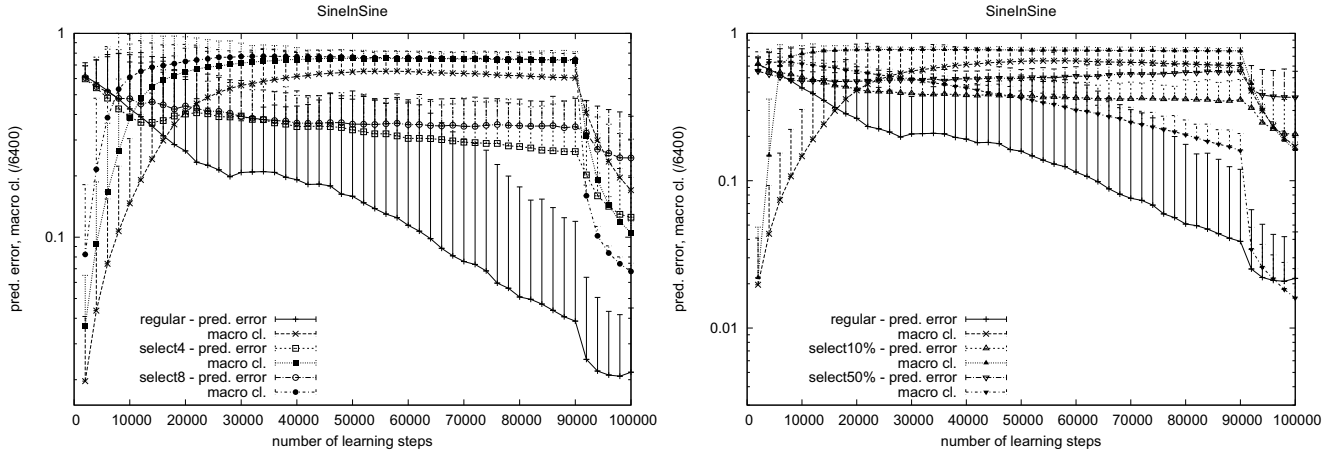
Figure 1 shows the final approximation reached by XCSF with compaction [6] after 100k learning iterations and with population sizes of $N = 6400$.[1] The graphs on the left hand-side show the actual function structures and generally confirm that XCSF is able to learn accurate approximations for all four functions. The contours at the bottom of the graphs show the general structure of the gradients on the surface. The graphs on the right show the classifier conditions of the XCSF final populations after 100k learning iterations. It can be seen that the classifiers orient themselves reflecting the function contours. On irrelevant dimension (such as the diagonal in f_1) XCSF exhibits typical genetic drift.

Performance was compared to the standard setting of two offspring classifiers for settings in which four and eight classifiers (with replacement) were selected for reproduction and 10% and 50% of the match set size classifiers (with replacement) were selected for reproduction. Learning progress is

[1]Other parameters were set to the following values: $\beta = .1$, $\eta = .5$, $\alpha = 1$, $\varepsilon_0 = .01$, $\nu = 5$, $\theta_{GA} = 50$, $\chi = 1.0$, $\mu = .05$, $r_0 = 1$, $\theta_{del} = 20$, $\delta = 0.1$, $\theta_{sub} = 20$. Compaction was started after $90k$ learning iterations. All experiments in this paper are averaged over 20 experiments.



(a) crossed ridge function



(b) sine-in-sine function

Figure 3: While in the crossed ridge function larger offspring sizes mainly speed-up learning, in the challenging sine-in-sine function, larger offspring sizes can strongly affect the final error level reached.

shown in Figure 2 for functions f_1 and f_2. It can be seen that in both cases standard XCSF with two offspring classifiers learns significantly slower than settings with a larger number of offspring classifiers. The number of distinct classifiers in the population, on the other hand, show that initially larger offspring sizes increase the population sizes much faster, indicating that initially higher diversity due to larger offspring sets yields faster learning. However, towards the end of the run, standard XCSF actually reaches a slightly lower error than the settings with larger offspring sets. This effect is the more pronounced the larger the offspring set. In the radial sine function, the effect is not as strong and learning takes longer for the standard XCSF settings. However, if the runs had been extended further, it is likely that standard XCSF would outperform all other offspring set sizes used. Similar observations can also be made in the crossed ridge function (Figure 3a). However, in Figure 3b, which shows performance in the sine-in-sine function f_4, we can see that a larger offspring set size can dramatically degrade performance. Since it is rather hard to detect the local regularities in f_4, larger offspring set sizes can prevent successful learning. While a selection of four offspring classifiers as well as a selection of a size of 10% of the match set size still shows slight error decreases, larger offspring sizes completely stall learning despite large and diverse population sizes. It appears that the larger offspring sizes prevent the population from identifying relevant structures and thus prevent the accurate function approximation.

3. THEORETICAL CONSIDERATIONS

What is really the effect of increasing the number of offspring generated upon GA invocation? The results indicate that initially, faster learning can be induced. However, later on, learning potentially stalls.

Previously, learning in XCS was characterized as an interactive learning process in which several evolutionary pressures [4] cause the learning progress: (1) A fitness pressure is induced since more accurate classifiers are selected for reproduction than those being deleted. (2) A set pressure, which causes an intrinsic generalization pressure, is induced since on average more general classifiers are selected for reproduction than those being deleted. (3) Mutation pressure causes diversification of classifier conditions. (4) Subsumption pressure causes convergence to maximally accu-

rate, general classifiers, if found. Larger offspring set sizes increase fitness pressure, since more classifiers are selected based on the current fitness estimates, however, larger offspring set sizes also increase the set pressure and mutation pressure, since the reproduced classifiers still origin in action sets and are all independently mutated, while excess classifiers are still deleted from the population as a whole. Thus, the balance between fitness, set, and mutation pressures is not directly affected by the increased offspring set sizes because the increase equally affects all pressure influences.

Another analysis estimated the reproductive opportunities a superior classifier might have before being deleted [3]. Moreover, a niche support bound was derived [2], which characterizes the probability that a classifier is sustained in the population, given that it represents an important problem niche for the final solution. Both of these bounds assume that the accuracy of the classifier is accurately estimated. However, the larger the offspring set size is the faster XCSF generates new classifiers and deletes older ones. Thus, the average classifier age in the population is smaller and thus the average number of learning iterations a classifier stays in the population is smaller. This has the effect that the number of iterations available to a classifier for reproductive success is smaller. And since the number of iterations are smaller, the GA has to work with classifier parameter estimates that are less reliable since they underwent less updates on average. Thus, larger offspring set sizes induce larger noise in the selection process. This appears to be the main reason why population sizes become larger initially. Moreover, as long as the fitness pressure still leads into the right direction since the parameter estimates have enough signal, learning proceeds faster. This latter reason stands also in relation to the estimated learning speed of XCS, approximated elsewhere [1]. Since reproductions of more accurate classifiers are increased learning speed increases as long as the more accurate classifiers are detected.

Due to this reasoning, however, it can also be expected that learning can stall prematurely. This should be the case when the signal-to-noise ratio is not sufficiently high to identify more accurate classifiers and consequently have a reliable fitness pressure. That is, the smaller the signal of a more accurate classifier relative to the noise that disturbs the signal, the harder it will be to identify the more accurate classifiers. In XCS terms, that is, when the time necessary to identify a more accurate classifier as actually more accurate is on average larger than the time until the deletion of that more accurate classifier, then the XCS learning progress can be expected to stall. Signal-to-noise ratios depend on the problem at hand, the space partitioning of the classifier, and the used linear approximation techniques (here we use recursive least squares). Thus, it is hard to specify the exact ratio in general and future research is needed to derive mathematical bounds on this problem. Nonetheless, the considerations explain the general observations in the considered functions: The more complex the function, the lower the signal-to-noise ratio in the function and thus the more problematic larger offspring sets become - until also the traditional two offspring classifiers are too fast to yield effective learning progress.

To control the signal-to-noise problem, consequently, it appears to be important to balance reproduction rates and offspring set sizes problem-dependently. A similar suggestion is made elsewhere for the control of parameter θ_{GA} [11]. Thus, we proceed now to an approach that decreases the offspring set size over a learning experiment to get the best of both worlds: Fast initial learning speeds and maximally accurate final solution representations.

4. ADAPTING OFFSPRING SET SIZES

As a first approach to determine if it can be useful to use large initial offspring set sizes and to decrease those size during the run, we linearly annealed the offspring set size from 10% offspring set size to 2 over the $100k$ learning iterations. Figure 4 shows the resulting performance in all four functions comparing the linearly reduced offspring set size with fixed two offspring classifiers and fixed 10% offspring. In graphs 4a,b,c we can see that the annealing technique reaches maximum accuracy. Particularly in graph 4a we can see that the performance stalling is overcome and an error level is reached that is similar to the one reached with that traditional XCS setting. Performance in function f_4 shows that the error still stays on a high level initially but it starts decreasing further when compared to a 10% offspring set size later on in the run. Although it does not reach the error level of the two offspring setting, performance is expected to further decrease when more learning interactions are executed.

Thus, the results show that a linear reduction of offspring set sizes can have positive effects and the final error level reached while still allowing faster initial learning speeds. However, the results also suggest that a fixed linear scheme is not necessarily optimal and its success is highly problem-dependent. Thus, in the future we intend to make offspring set sizes dependent on a current estimate of the signal-to-noise ratios.

5. CONCLUSIONS

This paper has shown that a fixed offspring set size does not necessarily yield the best learning speed that the XCSF classifier system can achieve. Larger offspring set sizes can strongly increase the initial learning speed but do not necessarily reach maximum accuracy. Adaptive offspring set sizes, if scheduled appropriately, can get the best of both worlds in yielding high initial learning speed and low final error. The results however also suggest that a simple adaptation scheme is not generally applicable. Furthermore, the theoretical considerations suggest that the signal-to-noise estimates could be used to control the GA offspring schedule and the offspring set sizes. Given the signal-to-noise ratio is large, a larger set of offspring should be generated.

Another consideration that needs to be taken into account in such an offspring generation scheme, however, is the fact that problem domains may be strongly unbalanced, in which some subspaces may be very easily approximated while others may be much more complex. In these cases, it has been shown, though, that the θ_{GA} threshold can be increased to ensure a representation of the complete problem space [11]. Future research should consider adapting θ_{GA} hand-in-hand with the offspring set sizes. In which way this may be accomplished exactly still needs to be determined. Nonetheless, it is hoped that the results and considerations of this work provide good clues in the right direction in order to speed-up XCS(F) learning and to make XCS(F) learning more robust in problems with low signal-to-noise ratios.

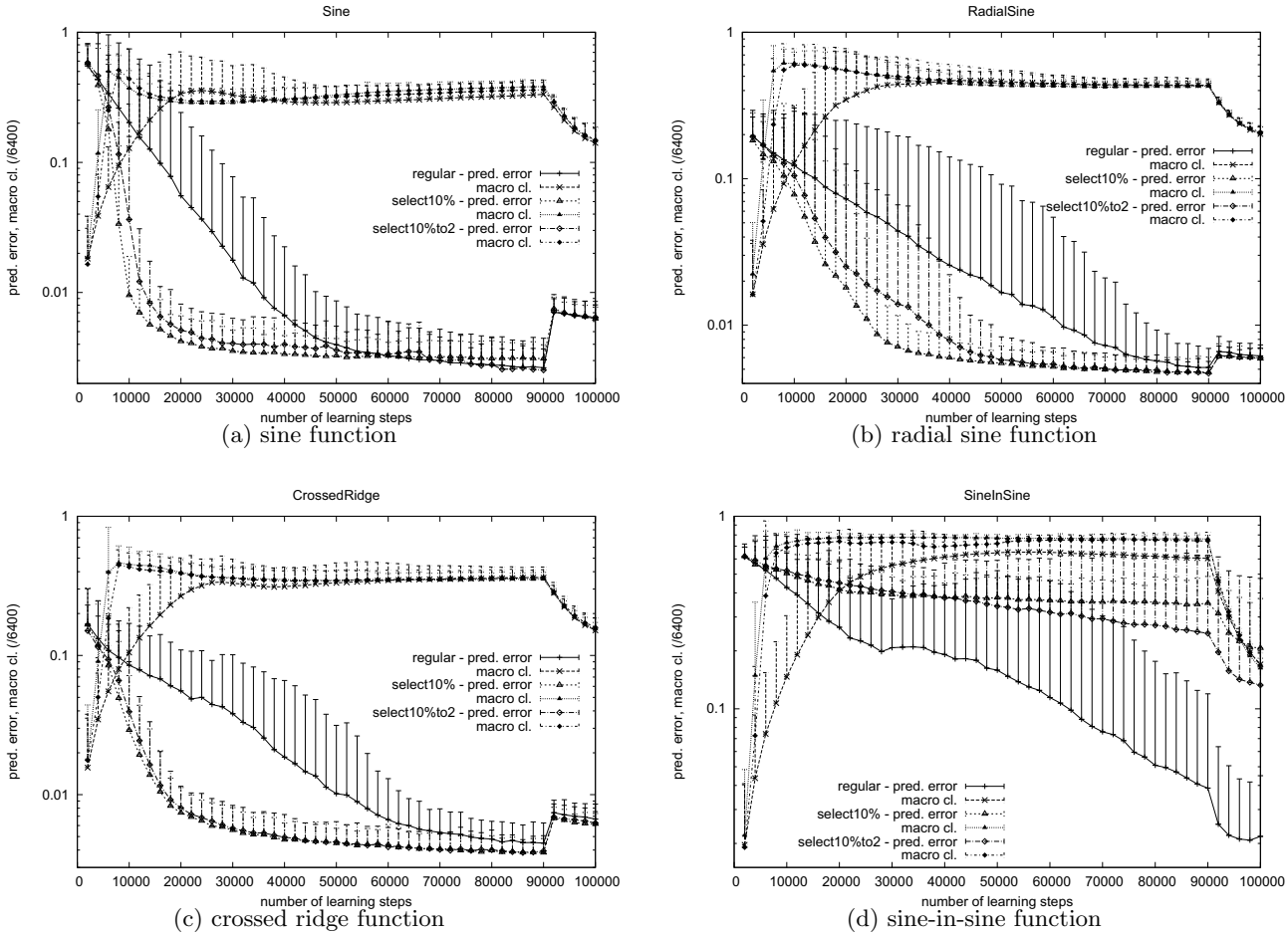


Figure 4: When decreasing the number of generated offspring over the learning trial, learning speed is kept high while the error convergence reaches the level that is reached by always generating two offspring classifiers (a,b,c). However, in the case of the challenging sine-in-sine function, further learning would be necessary to reach a similarly low error level (d).

6. ACKNOWLEDGMENTS

The authors acknowledge funding from the Emmy Noether program of the German research foundation (grant BU1335/3-1) and like to thank their colleagues at the department of psychology and the COBOSLAB team.

7. REFERENCES

[1] M. V. Butz, D. E. Goldberg, and P. L. Lanzi. Bounding learning time in XCS. *Proceedings of the Sixth Genetic and Evolutionary Computation Conference (GECCO-2004): Part II*, pages 739–750, 2004.

[2] M. V. Butz, D. E. Goldberg, P. L. Lanzi, and K. Sastry. Problem solution sustenance in XCS: Markov chain analysis of niche support distributions and the impact on computational complexity. *Genetic Programming and Evolvable Machines*, 8:5–37, 2007.

[3] M. V. Butz, D. E. Goldberg, and K. Tharakunnel. Analysis and improvement of fitness exploitation in XCS: Bounding models, tournament selection, and bilateral accuracy. *Evolutionary Computation*, 11:239–277, 2003.

[4] M. V. Butz, T. Kovacs, P. L. Lanzi, and S. W. Wilson. Toward a theory of generalization and learning in XCS. *IEEE Transactions on Evolutionary Computation*, 8:28–46, 2004.

[5] M. V. Butz, P. L. Lanzi, and S. W. Wilson. Hyper-ellipsoidal conditions in XCS: Rotation, linear approximation, and solution structure. *GECCO 2006: Genetic and Evolutionary Computation Conference*, pages 1457–1464, 2006.

[6] M. V. Butz, P. L. Lanzi, and S. W. Wilson. Function approximation with XCS: Hyperellipsoidal conditions, recursive least squares, and compaction. *IEEE Transactions on Evolutionary Computation*, in press.

[7] J. Drugowitsch and A. Barry. A formal framework and extensions for function approximation in learning classifier systems. *Machine Learning*, 70:45–88, 2008.

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

[9] J. H. Holland. Properties of the bucket brigade algorithm. *Proceedings of an International Conference on Genetic Algorithms and their Applications*, pages 1–7, 1985.

[10] P. L. Lanzi, D. Loiacono, S. W. Wilson, and D. E. Goldberg. Prediction update algorithms for XCSF: RLS, kalman filter and gain adaptation. *GECCO 2006: Genetic and Evolutionary Computation Conference*, pages 1505–1512, 2006.

[11] A. Orriols-Puig and E. Bernadó-Mansilla. Bounding XCS's parameters for unbalanced datasets. *GECCO 2006: Genetic and Evolutionary Computation Conference*, pages 1561–1568, 2006.

[12] B. Widrow and M. Hoff. Adaptive switching circuits. *Western Electronic Show and Convention*, 4:96–104, 1960.

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

[14] S. W. Wilson. Get real! XCS with continuous-valued inputs. In P. L. Lanzi, W. Stolzmann, and S. W. Wilson, editors, *Learning classifier systems: From foundations to applications (LNAI 1813)*, pages 209–219. Springer-Verlag, Berlin Heidelberg, 2000.

[15] S. W. Wilson. Classifiers that approximate functions. *Natural Computing*, 1:211–234, 2002.