Learning Classifier Systems: a way of reinforcement learning based on evolutionary techniques

Katarzyna Wasielewska [1] and Franciszek Seredyński [2,3]

[1] University of Applied Science and Art in Elblag, Institute of Applied Informatics, Elblag, Poland, email: kawa@pwsz.elblag.pl
[2] Polish Japanese Institute of Information Technology, Warsaw, Poland, email: sered@ipipan.waw.pl
[3] Polish Academy of Sciences, Institute of Computer Science, Warsaw, Poland

Abstract. Learning classifier systems (LCSs) are rule-based learning machines in which a reinforcement learning is conducted with use of evolutionary techniques. Currently, they are a subject of intensive study and of interesting applications. In this paper we present a review of LCSs. We give a short history of LCSs and overview current models. We also present some interesting and successful applications of LCSs.

1 Introduction

In machine learning, the *machine* is a software system running on a computer, while *learning* is analogous to the human learning behavior. The *behavior* is a product of an *interaction* between an *agent* and its environment, where the *agent* is some entity that can perform actions. The environment provides a positive or negative reward for received action.

The *Learning Classifier System* (LCS) is a rule-based learning machine introduced by John Holland [33,34] in the 1970s. This technique combines a reinforcement learning and evolutionary computing to produce adaptive systems. The LCS is the system that learns a syntactically simple string rules (called *classifiers*). Each classifier consists of two parts: <condition>:<action>. This rule means: "if a current observed state of the environment matches the *condition*, then execute the *action*". In most LCSs, classifier conditions have simple representation as strings in the ternary alphabet {0,1,#} while classifier actions are binary strings. Classifiers interacting with the environment receive real-value *rewards* and their fitness is updated. The idea of the LCS is presented in Figure 1. We can see three main components of LCS. The *performance* component governs an interaction with an environment. The *reinforcement* component (called *credit assignment* component) distributes the reward received from the environment to the classifiers. The *discovery* component is responsible for discovering better rules. The rules are discovered with the use of Genetic Algorithm (GA). The GA operates on a population of classifiers.

We can distinguish some types of LCSs: a *Michigan-style*, in which classifier system evolves a single population of rules and GA recombines and reproduces usually a very small number of the best rules in a rule set; a *Pittsburgh-style*, where classifier system evolves a population of rule sets, and GA acts on rule sets, and *anticipatory* LCS with anticipatory learning process.

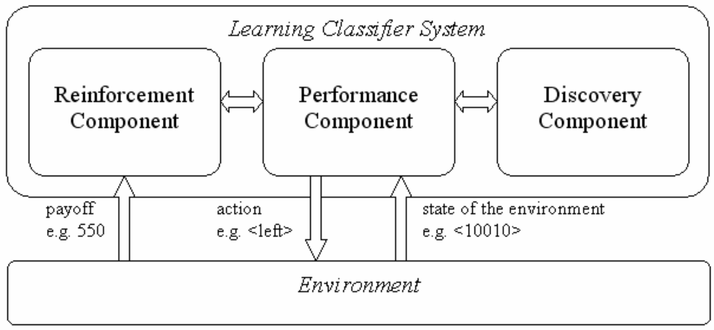

Figure 1. Idea of the LCS.

The rules are evaluated with a reinforcement learning algorithm (a credit assignment algorithm). In earlier LCSs a bucket brigade algorithm [34] was used for this purpose, where *strength* of rules was assigned according to the payoff prediction. The GA task is to discover new and potentially better rules. The GA uses a measure, calculated by the credit assignment algorithm, as the fitness of each rule. Currently Q-Learning algorithms are used (first in [76]) for this purpose.

The environments that LCS has to learn within are divided into two classes: *a single-step* and *a multi-step*. In single-step environments an environmental feedback is returned on each step of the LCS. The second class contains environments where a feedback is given after some number steps. The single-step problems are simpler for implementation and easier to learn, because each of actions is estimated. In contrast, the multi-step environment requires a chain of actions before a feedback is received.

The multi-step LCSs are applied in *Markov* and *non-Markov* (or partially observable) environments. The distinction between a Markov and a non-Markov environment is fundamental in reinforcement learning and it shows a separation between the LCSs and other learning systems which use traditional reinforcement learning algorithms. In the Markovian environment an agent learns rely on its sensors completely, while, in the non-Markovian environment, the agent needs additional memory of previous experience. First reports indicated unsatisfactory performance of traditional classifier system [60] when performing in the non-Markovian environments.

In 1989 Wilson and Goldberg presented a critical review of LCS research [74]. They summarized existing research results and suggested directions to development. They showed interesting applications of LCSs of the time (e.g. autonomous robotics, medical data analysis, agents environments). In [47] Lanzi and Riolo summarized the applications of LCSs in the next ten years.

The paper is organized as follows. Section 2 briefly overviews LCSs. In Section 3 we present a short description of one of most popular LCSs. Section 4 presents some interesting applications of LCSs. The last section contains conclusions.

2 Metamorphosis of LCS

In 1988 Riolo's CFS-C implementation appeared [54]. At that time, a Goldberg's standard implementation was made available [30]. Both implementations were based on the original architecture proposed by Holland [34] and used (a) a bucket brigade algorithm to distribute a reward received to the classifiers, (b) strength parameter of classifier, (c) an internal message list. Small modifications to the original framework were also proposed. Wilson introduced classifier system BOOLE [73] which solves a multiplexer problem and showed that it learns faster than neural networks. Then Bonelli [8] made changes in the performance of BOOLE and showed that his new classifier system NEWBOOLE is faster.

In 1989 Booker introduced a new type classifier system, GOFER-1, which has the property of anticipation [10]. In Booker's classifier system GA works in environmental niches and classifiers fitness is a function of both reward and other information. GOFER-1 was applied to a multiplexer problem and Booker obtained very good results.

In 1994 Wilson [75] simplified the original architecture of the LCS. He showed *Zeroth-level* Classifier System (ZCS) which has no internal message list and the reward is distributed to classifiers by QBB [75], a technique similar to Watkins' Q-learning [72] – a feature of both bucket brigade and Q-learning. The ZCS was modified slightly, and as an example of such a modification can serve the Dorigo's AlecSys classifier system [22]. However, first experiments showed that ZCS performance is not optimal [75]. Adding internal memory to the ZCS [83] improved the performance of system in the multi-step environments.

In 1995 Wilson made a breakthrough in LCS. He introduced the eXtended Classifier System (XCS) [76]. The architectures of XCS differs from all the previous architectures. The most important difference between XCS and other LCS is that classifier fitness for the GA is based on the accuracy of prediction of reward (e.g. in ZCS the rule fitness is based on a payoff received by classifier). The goal of the XCS is to form a complete and accurate mapping of the problem space through efficient generalizations [77]. The XCS uses standard Q-learning algorithm [72]. GA acts in environmental niches. To select parents in the GA, traditional roulette wheel [72] or tournament selection [16] are used.

Wilson introduced a version of XCS adjusted for continuously-valued inputs (called XCSR), in opposite to the binary values which were used in traditional systems [78]. The XCSR has modified representation of conditions (real values) and mechanisms of mutation and covering. Similarly, Wilson showed the XCS modified for integer values of inputs (called XCSI) [80]. Lanzi introduced other modifications of classifier's condition. Firstly, he proposed a LCS in which conditions were not of the same length (the classifier system called XCSm) [45] and secondly, a condition was based on S-expressions (the system called XCSL) [46]. Lanzi performed tests showing that these two systems reach optimal solutions. Tharakunnel and Goldberg [64] modified the prediction parameter of the XCS (they called it an Average Reward XCS - AXCS). They showed that this system learns similarly to the XCS in a multi-step environment.

There exist well-known works which blend features of fuzzy logic with classifier system (Fuzzy Learning Classifier Systems - FLCS) [69,7]. In these hybrid systems fuzzy rules are derived from human experts as linguistic *if-then* rules or are automatically generated from numerical data without domain experts.

Cliff and Ross applied ZCS with internal memory (ZCSM) [83] to solve problems in non-Markovian environments. They showed that ZCSM can solve these problems when the size of internal states is limited and they observed that when the size of internal memory grows then learning becomes unstable. The XCS can learn an optimal policy in Markovian environments where an optimal action is always determined solely by the state of current sensory inputs. However, when the XCS wants to solve a problem in non-Markovian environments, then it needs some memory mechanism too. The XCSM by Lanzi introduced a constant length of bit-register memory into general classifier system structure to record agent's experience [42]. Lanzi and Wilson showed that XCSM can learn optimal solutions in more difficult non-Markovian environments [43]. However, it turned out that in some situations the memory mechanism becomes useless. In [44] Lanzi introduced an extension to XCSM (called XCSMH) which is capable of learning an optimal policy in much more difficult partially observable environments.

There were proposed another mechanisms instead of the internal memory. Barry's idea consists in updating classifier prediction parameter if an action has caused a change in the inputs [4,5]. Tomlinson and Bull used corporations of classifiers, first, to ZCS [66], and then to XCS [67].

In 1997 Stolzmann [62] developed the Anticipatory Classifier System (ACS) inspired by a theory of Tolman [65]. The ACS combines the LCS framework with a representation of anticipations and anticipatory learning process. Classifiers within ACS are augmented with a further element in the form of a condition but specifying the form of the anticipated next input after the action is performed. The original structure of ACS did not include any generalization mechanism. In 2000 the mechanism of generalization was included [12,13]. Furthermore, different ACS applications were published (more information in [14]). The ACS has also different modifications, e.g. ACS2 [15], XACS [15], YACS [26], MACS [27].

Wilson [79,81] introduced a classifier system called XCSF, in which the prediction estimation mechanism is used to learn an approximation to functions. This system can be used in the learning of any function or mapping from a vector of input values to output values. Later, the XCSF was extended to XCS-LP [82] for single-step problems defined over continuous domains involving discrete actions. Next, XCSF was applied to tackle multi-step problems involving continuous inputs [48].

Llorà et al. [52] showed how accurate and maximally general classifiers can be evolved in Pittsburgh-style classifier system. They used the compact genetic algorithm (cGA) [32] and they introduced a Compact Classifier System (CCS) based on an estimation of distribution algorithm. Other LCSs with Pittsburgh-style approach have been also proposed. These are GABIL [19], GIL, COGIN, REGAL [28], GA-Miner, GALE [49], MOLCS-GA [51] or Gassist [2].

3 Short overview of XCS

XCS developed by Wilson [76] is currently one of the most popular LCSs. Classifiers of XCS have three parameters: prediction, prediction error and fitness (see, an example in Figure 2). These parameters are updated by Q-learning technique. XCS consists of classifiers sets: a *population set* [P] of all classifiers, a *match set* [M] containing classifiers from the population whose condition part matches the current input, and an action set [A] – the set of classifiers which actions will be sent to an environment (see, Figure 3).

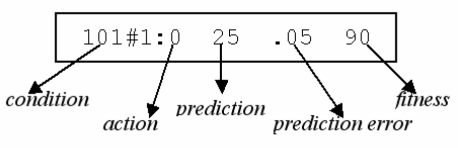

Figure 2. An example of a classifier of XCS.

At each time step the system receives a message from the environment. The system compares this message with conditions of classifiers from population [P] and creates a *match set* [M]. If the [M] is empty a new classifier is created through *covering* mechanism. Then for each possible action a_i the system prediction $P(a_i)$ is computed and prediction array P(A) is created. The value $P(a_i)$ gives an evaluation of the expected reward if action a_i is performed. Then, action selection is performed. The classifiers in [M] (which propose a selected action) are placed in the *action set* [A]. The selected action is executed and an immediate reward is returned to the system. The reward is used to update the parameters of the classifiers in [A]. GA in XCS is applied to [A].

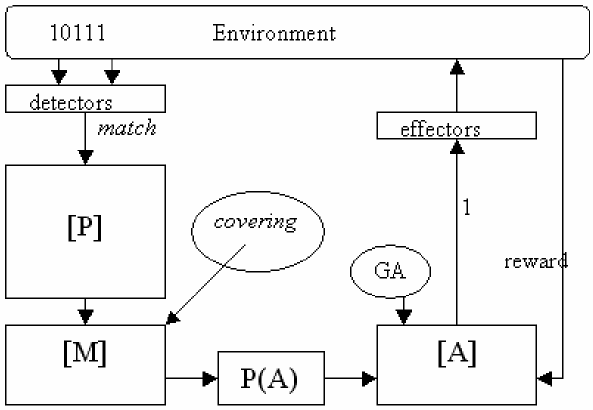

Figure 3. Schematic diagram of XCS for a single-step problem.

4 Applications of LCSs

In the following section we present an overview of some interesting applications of LCSs. First LCSs were used to test capabilities of learning processes and for simple problems in a single- and multi-step environments. A multiplexer problem (for single-step) and a maze (for multi-step) were favorite environments for tests of the LCSs.

4.1 Data mining

The LCSs were applied successfully to data mining problems. Holmes adjusted the LCS (called EpiCS) for solving epidemiological surveillance [35]. In particular, the EpiCS was applied to the problem of the head injuries of children involved in automobile crashes [36]. Holmes also performed its tests in EpiXCS (see [37]). Wilson applied the XCSI to the Winconsin Breast Cancer Database [80]. Llora built the GALE (*Genetic and Artificial Life Environment*) system for knowledge discovery in epidemiologic databases [50]. 2003 Bagnall et al. [3] used the LCS for the Forest Cover Data Set and they compared the process of learning of the LCS with other learning techniques (e.g. neural networks). Dam et al. introduced an extension of XCS for distributed data mining (DXCS) [84]. More information about the discovery of patterns within data can be found in [6].

4.2 Control

Booker used a LCS to control a simulated creature in a simple two-dimensional environment [9]. Goldberg demonstrated the application to the control of gas flow through a national pipeline system [29]. Vergas et al. used the LCS to the on-line reconfiguration of electric power distribution networks [70]. Cao et al. [18], Sha'aban et al. [59] and Bull et al. [11] used the LCS to an adaptive traffic control problem. The Fuzzy LCS have been successfully applied to various control problems [7,38,17] too.

4.3 Autonomous robotics and agents

Another area of applications of LCS is autonomous robotics. A lot of research concerning the animation of virtual entities was done. This is a very important field for experimentation with LCSs. The models use the LCS to build the dynamical behavior of agents. Many people applied different version of LCSs for this problem. The results of their works can be found in many of Dorigo's and Colombetti's papers (for example [23]), in works from Donnart's and Meyer's (e.g. [20]). Vasilyev used the LCS for autonomous agent control tasks. He links the classifier system with artificial neural networks (ANN) (e.g. [71]). Dorigo and Sirtori have developed a robot path planning system utilizing many classifier systems simultaneously [21]. Roberts applied classifier systems for learning in dynamic planning problems, such as determining plans of movement through artificial environments in search of food [55]. Donnart and Meyer developed a hierarchical architecture called MonaLysa for controlling autonomous agents [20]. This system links a module of classifier system with a set of other modules (e.g. place recognition module, planning module). Sanza et al. [56] applied LCSs for learning of agents in virtual soccer. Katagami and Yamada [39] introduced Interactive Classifier System (ICS) which was applied to create a mobile robot. They showed that a robot is able to learn rules quickly and a human operator can easily teach a physical robot [40,41]. Sato and Kanno showed the application of hybrid systems to the acquisition of decision-making algorithms for agents in online soccer games [57]. Carse and Pipe used Fuzzy LCS in tests in a real robot environment [17]. Sen et al. [58] studied multi-agent system coordination with LCS. The policy mapping of actions from perceptions to actions were used by multiple agents to learn coordination strategies without relying on shared information. They obtained results which indicated that classifier systems can be more effective than the more widely used Q-learning scheme for multi-agent coordination.

4.4 Other applications

Many other applications classifier systems exist. Swartz used Riolo's CFS-C [54] with little modifications to parse English text [63]. Richards applied the LCS to two- and three-dimensional shape optimization [53]. Federman et al. applied classifier system to predict the next note of music [24]. The LCSs were used for recognition problems, e.g. a letter recognition problem [25] or patterns recognition in language grammar [68]. Smith et al. [61] showed that the LCSs can be used to discover a novel fighter is maneuvering strategies. Guessoum et al. [31] used the XCS to build adaptive agents to simulate economic models. Afanasyeva [1] showed that classifier systems can be used for solving such well-known statistic problems as classification and that this approach can be a good alternative to existing classification methods (e.g. a discriminant analysis or neural networks).

5 Conclusion

Learning classifier systems are a way of using GA to machine learning problems. Classifier systems have been applied in many different areas. In particular one should not forget their contribution to research into adaptive systems. However, there are a lot of difficulties with the representation of knowledge and rate of learning which are a subject for current research.

Bibliography

[1] Afanasyeva H. Fuzzy Learning Classifier Systems for Classification Task. Transport and Telecommunication, Vol. 3, No. 3, 43–51, 2002.

[2] Bacardit J., Garrell J.M. Bloat control and generalization pressure using the minimum description length principle for a Pittsburgh approach learning classifier system. In *Proceedings of the 6th International Workshop on Learning Classifier Systems*, LNAI, Springer, 2003.

[3] Bagnall A.J., Cawley G. C. Learning classifier systems for data mining: A comparison of XCS with other classifiers for the Forest Cover dataset. *In Proceedings of the IJCNN*, Portland, Oregon, USA, Vol. 3, 1802-1807, 2003.

[4] Barry A.M. Aliasing in XCS and the Consecutive State Problem 1 – Problems. In Banzhaf W. et al. (editors), In *Proceedings of the GECCO*. Morgan Kaufmann, 19-26, 1999.

[5] Barry A.M. Aliasing in XCS and the Consecutive State Problem 2 – Solutions. In Banzhaf W. et al. (editors), In *Proceedings of the GECCO*. Morgan Kaufmann, 27-34, 1999.

[6] Barry A.M., Holmes J., Llorà X. Data Mining using Learning Classifier Systems. In Bull L. (editor), Applications of Learning Classifier Systems, Springer-Verlag, 2004.

[7] Bonarini A. An Introduction to Learning Fuzzy Classifier Systems. In Lanzi P.L., Stolzmann W., Wilson S.W. (editors), Learning Classifier Systems. From Foundations to Applications, LNAI 1813, Berlin, Springer-Verlag, 83- 104, 2000.

[8] Bonelli P., Parodi A., Sen S., Wilson S.W. NEWBOOLE: A Fast GBML System. ML, 153-159, 1990.

[9] Booker L.B. Intelligent Behavior as an Adaptation to the Task Environment. Ph.D. Dissertation, University of Michigan, 1982.

[10] Booker L.B. Triggered Rule Discovery in Classifier Systems. Proceedings of the ICGA, 265-274, 1989.

[11] Bull L., Sha'Aban J., Tomlinson, A., Addison J.D., Heydecker B.G. Towards distributed adaptive control for road traffic junction signals using learning classifier systems. In Bull L. (editor), Applications of learning classifier systems. Studies in Fuzziness and Soft Computing (150), Springer, New York, 279-299, 2004.

[12] Butz M.V., Goldberg D.E., Stolzmann W. Introducing a Genetic Generalization Pressure to the Anticipatory Classifier System Part 1: Theoretical Approach. In *Proceedings of the GECCO*, 34-41, 2000.

[13] Butz M.V., Goldberg D.E., Stolzmann W. Introducing a Genetic Generalization Pressure to the Anticipatory Classifier System Part 2: Performance Analysis. In *Proceedings of the GECCO*, 42-49, 2000.

[14] Butz M.V. Anticipatory learning classifier systems, Boston, MA: Kluwer Akademic Publishers, 2002.

[15] Butz M.V., Goldberg D.E. Generalized state values in an anticipatory learning classifier system. In Butz M.V., Sigaud O., Gérard P. (editors) Anticipatory Behavior in Adaptive Learning Systems, Springer-Verlag, Berlin Heidelberg, 282-301, 2002.
[16] Butz M.V., Sastry K., Goldberg D.E. Tournament Selection in XCS. In *Proceedings of the GECCO*, Springer Verlag, Berlin, 1857-1869, 2003.
[17] Carse B., Pipe A.G. X-FCS: a fuzzy classifier systems using accuracy based fitness – first results. In *Proceedings of the EUSFLAT*, 195-198, 2001.
[18] Cao Y.J., Ireson N., Bull L., Miles R. An Evolutionary Intelligent Agents Approach to Traffic Signal Control. International Journal of Knowledge-based Intelligent Engineering Systems 5(4), 279-289, 2001.
[19] DeJong K.A., Spears W.M., Gordon D.F. Using genetic algorithms for concept learning. Machine Learning, 13(2/3), 161–188, 1993.
[20] Donnart J.Y., Meyer J.A. Hierarchical-map building and self-positioning with MonaLysa. Adaptive Behavior, 5(1), 29-74, 1996.
[21] Dorigo M., Sirtori E. Alecsys: A Parallel Laboratory for Learning Classifier Systems. In *Proceedings of 4th International Conference on Genetic Algorithms*, San Diego, California, Morgan Kaufmann, San Mateo, CA, 1991.
[22] Dorigo M. Genetic and Non-Genetic Operators in Alecsys**.** Evolutionary Computation, 1, 2, MIT Press, 151-164, 1993.
[23] Dorigo M., Colombetti M. Robot Shaping: An Experiment in Behaviour Engineering. MIT Press/Bradford Books 1997. In Special issue on Complete Agent Learning in Complex Environments, M.J. Mataric (Ed.), Adaptive Behavior, 5, 3-4, 391-406, 1997.
[24] Federman F., Dorchak S. F. Representation of Music in a Learning Classifier System. ISMIS, 267-276, 1997.
[25] Frey P.W., Slate D.J. Letter recognition using Holland-style adaptive classifiers. Machine Learning, 6, 161-182, 1991.
[26] Gérard P., Stolzmann W., Sigaud O. YACS: a new Learning Classifier System using Anticipation. Journal of Soft Computing, Special Issue on Learning Classifier Systems, Berlin, Springer-Verlag, 2000.
[27] Gérard P., Sigaud O. Designing efficient exploration with macs: Modules and function approximation. In *Proceedings of the GECCO*, Chicago, IL. Springer-Verlag, 1882-1893, 2003.
[28] Giordana A., Saitta L. Regal: an integrated system for learning relations using genetic algorithms. In *Proceedings of 2nd International Workshop on Multistrategy Learning*, Morgan Kaufmann, 234-249, 1993.
[29] Goldberg D.E. Computer-Aided Gas Pipeline Operation Using Genetic Algorithms and Rule Learning. Ph.D. Dissertation, University of Michigan, 1993.
[30] Goldberg D.E., (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Publishing Co. 1989.
[31] Guessoum Z., Rejeb L., Sigaud O. Using XCS to build adaptive agents. Proceedings of the AAMAS Symposium, Leeds, 2004.
[32] Harik G., Lobo F., Goldberg D.E. The compact genetic algorithm. In *Proceedings of the IEEE International Conference on Evolutionary Computation*, 523–528, 1998.
[33] Holland J.H. Processing and processors for schemata. In Associative Information Techniques. Jacks E. L. (editor) Elsevier, 127-46, 1971.
[34] Holland J.H., Reitman, J. Cognitive systems based on adaptive algorithms. In Waterman D., Hayess-Roth F. (editors), Pattern-directed Inference Systems. Academic Press, New York, 1978.

[35] Holmes J.H. Evolution-Assisted Discovery of Sentinel Features in Epidemiologic Surveillance. PhD thesis, Drexel University, 1996.
[36] Holmes J. H. Learning Classifier Systems Applied to Knowledge Discovery in Clinical Research Databases. Learning Classifier Systems, 243-262, 1999.
[37] Holmes J.H., Sager J.A. Rule Discovery in Epidemiologic Surveillance Data Using EpiXCS: An Evolutionary Computation Approach. AIME 2005, 444-452, 2005.
[38] Ishibuchi H., Nakashima T., Murata T. Performance Evaluation of Fuzzy classifier Systems for Multi-Dimensional Pattern Classification Problems. IEEE Transactions on Systems, Man, and Cybernetics, Vol. 29, No. 5, 601-618, 1999.
[39] Katagami D., Yamada S. Interactive Classifier System for Real Robot Learning. IEEE International Workshop on Robot and Human Interaction (ROMAN), 258-263, 2000.
[40] Katagami D., Yamada S. Real Robot Learning with Human Teaching, The 4th Japan-Australia Joint Workshop on Intelligent and Evolutionary Systems, 263-270, 2001.
[41] Katagami D., Yamada S. Interactive Evolutionary Computation for Real Robot from Viewpoint of Observation. The 7th International Conference on Intelligent Autonomous, 158-165, 2002.
[42] Lanzi P.L. Adding Memory to XCS. In *Proceedings of the IEEE World Congress on Computational Intelligence*, The 1998 IEEE International Conference on Evolutionary Computation, May 4-9 Anchorage (AL), IEEE Press, 609-614, 1998.
[43] Lanzi P.L. An analysis of the memory mechanism of XCSM. 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. (editors), In *Proceedings of the 3rd Annual Conference*, San Francisco, CA, USA, Morgan Kaufmann, 643-651, 1998.
[44] Lanzi P.L., Wilson S.W. Optimal classifier system performance in non-Markov environments. Technical Report 99.36, Dipartimento di Elettronica e Informazione, Politecnico di Milano, 1999.
[45] Lanzi P.L. Extending the Representation of Classifier Conditions Part I: From Binary to Messy Coding. In Banzhaf W., Daida J., Eiben A.E., Garzon M.H., Honavar V., Jakiela M., Smith R.E. (editors), In *Proceedings of the GECCO*, Orlando (FL), Morgan Kaufmann, 337-344, 1999.
[46] Lanzi P.L., Perrucci A. Extending the Representation of Classifier Conditions Part II: From Messy Coding to S-Expressions. In Banzhaf W., Daida J., Eiben A.E., Garzon M. H., Honavar V., Jakiela M., Smith R. E. (editors), In *Proceedings of the GECCO*, Orlando (FL), Morgan Kaufmann, 345-352, 1999.
[47] Lanzi P.L., Riolo R.L. A Roadmap to the Last Decade of Learning Classifier System Research. In Lanzi P.L. Stolzmann W., Wilson S.W. (editors). Learning Classifier Systems. From Foundations to Applications, LNAI 1813. Springer, 33-62, 2000.
[48] Lanzi P.L., Loiacono D., Wilson S.W., Goldberg D.E. XCS with computed prediction in continuous multistep environments. IlliGAL Report, 2005.
[49] Llorà X., Garrell J.M. Knowledge-independent data mining with fine-grained parallel evolutionary algorithms. In *Proceedings of the 3rd Genetic and Evolutionary Computation Conference*, Morgan Kaufmann, 461–468, 2001.
[50] Llorà X. Genetic Based Machine Learning using Fine-grained Parallelism for Data Mining. PhD thesis, Enginyeria i Arquitectura La Salle. Ramon Llull University, Barcelona, 2002.
[51] Llorà X., Goldberg D.E., Traus I., Bernadó E., Mansilla I. Accuracy, Parsimony, and Generality in Evolutionary Learning Systems via Multiobjective Selection. IWLCS, 118-142, 2002.

[52] Llorà X., Sastry K., Goldberg D.E. The compact classifier system: Scalability analysis and first results. In *Proceedings of the Congress on Evolutionary Computation***, 1,** 596-603, 2005.
[53] Richards R. A. Zeroth-order shape optimization utilizing a learning classifier system. Web version of book: http://www.stanford.edu/~buc/SPHINcsX/book.html, 1995.
[54] Riolo R.L. CFS-C: A Package of Domain-Independent Subroutines for Implementing Classifier Systems in Arbitrary User-Defined Environments. Technical Report, University of Michigan, 1988.
[55] Roberts G.R. Dynamic planning for Classifier Systems. Forrest S. (editor), Proceedings of the 5th International Conference on Genetic Algorithms, Morgan Kaufmann, San Mateo, CA, 231-237, 1993.
[56] Sanza C., Panatier C., Duthen Y. Communication and Interaction with Learning Agents in Virtual Soccer. 2nd International Conference on Virtual Worlds. LNCS 1834 Juillet, Paris, FRANCE, 2000.
[57] Sato Y., Kanno R. Event-driven learning classifier systems for online soccer games. GECCO, 2201-2202, 2005.
[58] Sen S., Sekaran M. Multi-agent coordination with learning classifier systems. Proceedings of the IJCAI Workshop on Adaption and Learning in Multi-Agent Systems, Weiß G., Sen S. (editors), Vol. 1042. Springer Verlag, 218–233, 1996.
[59] Sha'Aban J., Tomlinson A., Heydecker B.G., Bull L. Adaptive Traffic Control Using Evolutionary Algorithms. In *Proceedings of 9th Meeting of the EURO Working Group on Transportation*, Bari, Italy, 2002.
[60] Smith R.E. Memory Exploitation in Learning Classifier Systems. Evolutionary Computation 2(3), 199-220, 1994.
[61] Smith R.E., Dike B.A., El-Fallah A., Mehra R.K. The Fighter Aircraft LCS: A Case of Different LCS Goals and Techniques. In Lanzi P.L., Stolzmann W., Wilson S.W. (editors), Learning Classifier Systems: From Foundations to Applications, Springer, 283-300, 2000.
[62] Stolzmann W. Antizipative Classifier Systems. Ph.D. Thesis, Fachbereich Mathematik/Informatik, University of Osnabrueck, 1997.
[63] Swartz R., Using a Classifier System to Parse English Text, Princeton University, web version: http://www.cs.princeton.edu/~rswartz/classifier/writeup.html.
[64] Tharakunnel K., Goldberg D.E. XCS with Average Reward Criterion in Multi-Step Environment. IlliGAL Technical Report 2002008, 2002.
[65] Tolman E.C. Purposive Behaviour in Animals and Man, New York, Appleton, 1932.
[66] Tomlinson A., Bull. A Corporate Classifier System. In Eiben A.E., Baeck T., Schoenauer M., Schwefel H.P. (editors), Parallel Problem Solving from Nature - PPSN V. Springer Verlag, 550-559, 1998.
[67] Tomlinson A., Bull L. A Corporate XCS. In Lanzi P.L., Stolzmann W., Wilson S.W., editors, Learning Classifier Systems: From Foundations to Applications, Springer, 194-208, 2000.
[68] Unold O., Dabrowski G. Use of Learning Classifier System for Inferring Natural Language Grammar. HIS, 272-278, 2003.
[69] Valenzuela-Rendón M. The Fuzzy Classifier System: A Classifier System for Continuously Varying Variables. ICGA, 346-353, 1991.
[70] Vargas P.A., Lyra C., Von Zuben F.J. Application of Learning Classifier Systems to the On Line Reconfiguration of Electric Power Distribution Networks. In Bull L. (editor),

Applications of Learning Classifier Systems, Springer Series: Studies in Fuzziness and Soft Computing, Vol. 150, 260-275, 2004.

[71] Vasilyev A. Autonomous Agent Control Using Connectionist XCS Classifier System. Transport and Telecommunication, Vol. 3, No 3, 56–63, 2002.

[72] Watkins C.J.C.H. Learning from Delayed Rewards. Ph.D thesis, Cambridge University, 1989.

[73] Wilson S.W. Classifier Systems and the Animat Problem. Machine Learning 2: 219-228, 1987.

[74] Wilson S.W., Goldberg, D.E. A critical review of classifier systems. In *Proceedings of the Third International Conference on Genetic Algorithms*, Los Altos, California: Morgan Kaufmann, 244-255, 1989.

[75] Wilson S.W. ZCS: A Zeroth-level Classifier System. Evolutionary Computation 2(1), 1-18, 1994.

[76] Wilson, S.W. Classifier Fitness Based on Accuracy. Evolutionary Computation, 3(2), 149-76, 1995.

[77] Wilson S.W. Generalization in the XCS classifier system. In *Proceedings of the Third Annual Conference*, 665-674, 1998.

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

[79] Wilson S.W. Function approximation with a classifier system. In L. S. et al., editor, In *Proceedings of GECCO*, San Francisco, California, USA, Morgan Kaufmann, 974–981, 2001.

[80] Wilson S.W. Mining oblique data with XCS. In Lanzi, P. L., Stolzmann, W., & Wilson, S. W. (Eds.), Advances in learning classifier systems: Third international workshop, IWLCS 2000, LNAI 1996. Berlin Heidelberg: Springer-Verlag, 158 – 174, 2001.

[81] Wilson S.W. Classifiers that approximate functions. Journal of Natural Computating, 1(2-3), 211–234, 2002.

[82] Wilson S.W. Classifier systems for continuous payoff environments. In K. D. et al., editor, GECCO, Part II, LNCS 3103, Springer-Verlag, 824–835, 2004.

[83] Cliff D., Ross S. Adding Temporary Memory to ZCS. Adaptive Behavior, 3(2), 101–150, 1994.

[84] Dam H.H., Abbass H.A., Lokan Ch. DXCS: an XCS System For Distributed Data Mining. In *Proceedings of GECCO*, 2005.