

Learning Classifier Systems

JASON BROWNLEE

Technical Report 070514A

Complex Intelligent Systems Laboratory, Centre for Information Technology Research,
Faculty of Information and Communication Technologies, Swinburne University of Technology
Melbourne, Australia
jbrownlee@ict.swin.edu.au

Abstract—Learning Classifier Systems are a machine learning technique that may be categorised in between symbolic production systems and sub-symbolic connectionist systems. Classifiers are cognitive paradigm for adaptation that learn in environments of perpetual novelty with minimal and delayed reward. They employ two principle processes (1) reinforcement learning called ‘trial-and-error’, and genetic evolution called ‘survival-of-the-fittest’. This work provides a brief review of classifier systems with a focus on the principles of the learning paradigm.

Keywords—Learning Classifier System, LCS, Reinforcement Learning

I. INTRODUCTION

Learning Classifier Systems (LCS) are a machine learning approach that employs reinforcement learning and a genetic algorithm to evolve a set of binary encoded rules. They are traditionally applied to fields including autonomous robot navigation, supervised classification, and data mining. LCS were proposed in the late 1970’s and intensely developed in the 1980’s. The complexity of the approach lead to the loss of interest in the approach during the early 1990’s, although the approach resurged, likely due to the proposal of the simplified and effective XCS variation. This work provides a brief review of LCS, with a focus on the principles surrounding the paradigms inception by John Holland. For relevant and recent books on the topic, see [15,16,24] and a seminal reference [26].

II. HOLLAND’S CLASSIFIER SYSTEMS

Classifier systems are message-passing rule-based systems that learn using credit assignment (the bucket brigade algorithm) and rule discovery (the genetic algorithm) [14]. They are suited for problems with the following characteristics:

1. Perpetually novel events with large amounts of noise
2. Continual, and real-time requirements for action
3. Implicitly or inexactly defined goals
4. Sparse payoff or reinforcement obtainable only through long sequences of tasks

Classifier systems were proposed by John Holland [4,9], and later standardised [5]. In his 1992 edition of his seminal work on Adaptation [12], Holland suggested that classifier systems were proposed to investigate genetic-based learning in problem domains that were

characteristic of most learning situations for animals and humans.

“How does a system improve its performance in a perpetually novel environment where overt ratings of performance are only rarely available?” ([12] pg. 172)

In [9] Holland presents his classifier system as a ‘computationally complete’ cognitive system with four elements: (1) A set of elementary interacting units called classifiers. (2) A performance algorithm that directs the action of the system in the environment. (3) A simple learning algorithm that keeps track of each classifiers success in receiving rewards. (4) A more complex learning algorithm that modifies the set of classifiers such that good classifiers persist, and new variants of good classifiers are proposed. The result is that the system generates an experience-based cognitive map that lets the system look-ahead and assign credit during non-reward intervals.

The actors of the system include (1) detectors, (2) messages, (3) effectors, (4) feedback, and (5) classifiers.

Detectors: Used by the system to perceive the state of the environment
Messages: Information passed from the detectors into the system as discrete information packets. The system performs information processing on messages, and messages may directly result in actions in the environment
Effectors: Control the systems actions on and within the environment
Feedback: In addition to the system actively perceiving via its detections, it may also receive directed feedback from the environment (payoff)
Classifier: A condition-action rule that provides a filter for messages. If a message satisfies the conditional part of the classifier, the action of the classifier triggers. Rules act as message processors.

Figure 1 - Summary of the principle components of Holland’s classifier system

Messages are defined of length k using a binary alphabet. A classifier is defined as a bit string with a trinary alphabet of $\{1, 0, \#\}$, where the ‘#’ represents ‘do not care’. The classifier’s string has at least two parts: one or more conditional parts, and an action part.

1##01#10/001011

[condition] / [action]

Messages enter the system and are placed onto a message list, then a matching process occurs in which classifiers seek activation by matching their bit strings against all messages in the list. Those message that are activated, compete and the winners post their messages onto the message list. Thus the message list provides a blackboard for communication both internally (between classifiers), and externally (detectors and effectors). The

symbol of a classifier defines the generality or specificity of the conditional part of the message. In addition, a message may be such that the condition of classifiers is negated (NOT), thus messages may be assigned sign (+/-). After competition, more than one winning-activated rules may act concurrently. This feature facilitates emergent concepts of the environment innate in the collective of rules, and a transfer of experience to new situations.

The classifier system executes in discrete cycles, as follows:

- 1) Messages from the environment are placed on the message list
- 2) The conditions of each classifier are checked to see if they are satisfied by at least one message in the message list
- 3) All classifiers that are satisfied participate in a competition, those that win post their action to the message list
- 4) All messages directed to the effectors are executed (causing actions in the environment)
- 5) All messages on the message list from the previous cycle are deleted (messages persist for a single cycle)

Figure 2 - Classifier system execution cycle (from [12] pg.175)

Competition is used to determine which of the activated classifier are triggered for a given execution cycle of the system. Each classifier is assigned a strength that summarises that classifiers usefulness in past cycles. Thus, a classifier may be considered an IF-THEN hypothesis where the strength describes the validity of the hypothesis as defined by the systems experience. Competition occurs in a bidding process (bidding function) that not only includes the strength of the classifier, but also its specificity. Winners are selected probabilistically proportional to their bid, such that stronger rules are more likely to win the competition.

An important problem is that of rating rules, referred to as credit assignment. This problem is complicated, because feedback from the environment regarding the systems performance (holistic) is provided intermittently. Further, this problem is of particular concern early in the systems genesis, before any feedback has been received from the environment. Credit assignment is addressed using the bucket brigade credit assignment algorithm that provides a way of distributing payoff *if* and *when* it is received from the environment.

In this algorithm, a classifier may be a *consumer* and or a *supplier* of payoff. A classifier may consume the message output from another classifier from the previous cycle. When a classifier consumes a message and wins a competition, the supplier of that message receives payment, which comes at a personal cost to the winner classifier. Thus, the winning bid made by a classifier during a competition is subtracted from the winner and paid to the supplier of the message. When the system is given feedback from the environment, it is distributed to all currently active rules, increasing their strength. In this way, the environment reinforces the current classifiers strengths ensuring that only those classifiers that are in the chain that received reinforcement are made stronger. The system requires multiple plays of the game (many cycles with environment reinforcement) to stabilise a coherent classifier set. Interestingly, this process requires no overt memory, rather memory is implicit in the

system.

A final major concern is that of getting new rules into the system, referred to as rule discovery. The system needs an effective way of proposing plausible classifiers to replace low-strength classifiers. In fact, the systems performance as an induction system is dependent on its ability to propose plausible replacement rules. The principle employed is that the systems experience biases the generation of replacement rules. A genetic algorithm is employed that probabilistically selects and recombines rules from the current set to propose replacement rules. The fitness of a classifier is defined by its usefulness (strength), which as has been discussed is an experience-dependant guideline or estimation with inherent errors. Thus, careful consideration is required regarding the genetic algorithms selection strength, and the rules for selecting the classifiers in the system to replace.

Holistically, the system provides an incremental way of modelling an environment, where the system perpetually gains experience and tries new rules. Thus, the system is designed to continually adapt to the environment, attempting to balance *exploration* (acquisition of new information and capabilities) and *exploitation* (the efficient use of information and capabilities already established) [12].

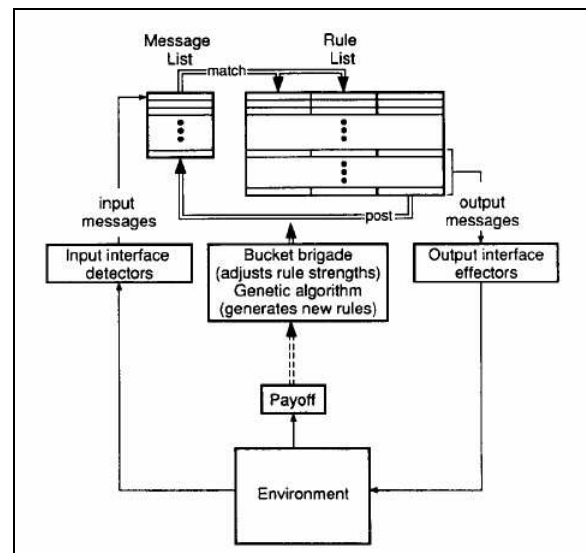


Figure 3 - Overview of Holland's classifier system ([6], pg. 191)

In their seminal work on induction and inductive systems [10], Holland, et al. propose three important properties of classifiers systems: parallelism, message passing, and the systems lack of reliance of interpreters, as follows:

- Parallelism:** Large numbers of classifiers can be active at the same time. There is no need to schedule rules because classifiers may only post messages to the message list. Rules are used as building blocks, where the activation of multiple rules encapsulates concepts, which may be acted upon in the domain.
- Message Passing:** All communication in and out of the system is performed in messages, this includes input messages from the environment, and action messages from triggered rules.
- Lack of Interpreters:** Interaction is based on messages, and message triggering is based on matching, thus there is no need for high-level interpreters. The system is modular and graceful, and it is possible to add new candidate classifiers to the system without global disruption.

Figure 4 - Summary of the important properties of classifier systems (from

Holland proceeds to suggest that the general classifier system may be augmented in various ways such as specialized algorithms for planning, learning, and inference, such that it may be applied to varied domains. He provides a number of handcrafted examples as follows: Simple stimulus-response classifier (if then rule), Rules for encoding relations (compound object identified by the triggering of multiple rules), Simple memory (an internal alert status for a specific event), Building blocks (combining a number of active rules to handle complex situations), Networks of tagging (networks as hierarchies of classifiers, where classifiers are coupled using tags).

David Goldberg provided perhaps the seminal application of Holland's classifiers (bucket brigade and a genetic algorithm) in his Ph.D. dissertation [3]. A classifier system was designed to control a cart-and-pole balancing problem, and to learn to regulate a simulated gas-pipeline system during normal summer and winter periods, and detect gas leaks with increasing effectiveness. The systems performance was achieved through effective learning from a random starting point, without implanted expert knowledge.

Other classical example applications include Forrest's [31] application to the classification of knowledge in semantic networks. Smith's development of the LS-1 version of the classifier system and the application to a maze problem and poker [35]. Booker's work investigated the connections between classifier and cognitive systems, and the adaptive behaviour of an artificial creature in a two-dimensional environment [17]. Finally, Wilson's work with artificial animals called (Animat's) [36,37].

Belew and Forrest explored the relationship between classifier systems and symbolic and sub-symbolic learning systems in their work that proposes hybrids between classifier systems and symbolic systems [28]. Production systems and expert systems are classic examples of symbolic-based learning systems in which knowledge is explicitly encoded, stored, and employed for reasoning. Such systems have good lookahead capabilities using means-ends analyses, although are poor at autonomous construction of an experience based model. These systems may be contrasted with sub-symbolic approaches of which connectionist models (neural networks) are a typical example. These systems construct internal models from provided example data thus are good at autonomous construction of experience-based models, although are not good at organizing knowledge into models that guide the system by lookahead and anticipation.

Classifier systems are rule-based lookahead-oriented systems and use means-ends analysis, although they use a sub-symbolic representation and autonomous experienced based learning processes. They are representative of a middle ground between symbolic and sub-symbolic based inductive learning approaches, exploiting the parallel processing of connectionist approaches, and the anticipatory properties of expert systems.

This relationship is discussed by Booker, et al, [14], and by Holland in his proposal of tags for augmenting the lookahead and anticipatory properties of classifier systems [6]. Holland proposes that a lookahead internal model of the world is required to handle the constant flow of performance-unrelated information from the environment. He proposes the classifier systems construct lookahead models from sets of rules, where the anticipatory feature is emergent.

Both symbolic and sub-symbolic systems suffer from the same problem in categorizing signals from the environment – they can only categorise signals that can be distinguished. In order to lookahead, Holland proposes a system requires internal stimulus as well as external. For the system to be effective at looking ahead, the system needs to be working on the problem in the absence of inputs. These are so-called 're-entrant connections', or 're-circulation of input pulses' in connectionist approaches. This is a feature possessed by classifier systems in their ability to continue to process information and assign payoff (via the bucket brigade algorithm) in if the presence of infrequent environmental feedback.

III. LEARNING CLASSIFIER SYSTEMS

Learning Classifier Systems (LCS) refers to a paradigm (pattern of thinking), thus a whole class of adaptive, learning, cognitive systems based on the principles of Holland's original proposal, so called Genetics-Based Machine Learning (GBML) [2]. See [39] and [14] for classical reviews of classifier systems, and [25] for an effective summary of the field in the intervening 10 years. In that review, learning classifier systems are defined with regard to two key principles: '*survival of the fittest*' to trigger adaptation in a system to an unknown environment (fields that became evolutionary computation, adaptive behaviour, and artificial life), and '*trial and error*' in learning through interactions with an environment in which the system seeks to maximise rewards (reinforcement learning).

An excellent summary of what learning classifiers are is provided by the past and present leaders in the field [11]. Among the many comments regarding the history, application, and theory of classifier systems is Riolo's insightful summation of the principle characteristics of a LCS in the context of modelling complex systems, as follows:

Message Board: A place to add and remove messages
Rules: A rule-based representation of knowledge
Competition: A competition for rules to become active based on inputs, past performance, and predictions of future expected outcomes
Parallel Firing of Rules: Consistency and coordination emerge from the dynamics of the bidding process. Explicit conflict resolution occurs only with effectors
Credit Assignment: Credit is assigned using temporal-difference learning (TD) methods such as bucket brigade, profit sharing, Q-learning, and mixtures of such schemes operating at the same time.
Rule Discovery: New rules are discovered using heuristics appropriate to the specific application, with genetic algorithms as the traditional choice.

Figure 5 - Summary of the principle characteristics of LCS (Riolo [11])

From the same work, Smith, provides a definition that is independent of syntax or implementation details as a system defined by a population of entities that:

Act individually, responding to and tacking actions on an external environment, while evolving as population members, under the action of evolutionary computation

A review of the field is provided in [8], in that review, three arguments are provided in support of classifier systems. They are as follows:

Adaptability: Classifier systems are adaptive, capable of online learning in rapidly changing environments. They have demonstrated their adaptiveness over different time scales, such on data collected over weeks and months, on data collected over days, and in real-time scenarios.

Generalization: An important feature of learning systems is their ability to generalise, which includes (1) the ability to represent what it has learned in a compact form, and (2) apply what it has learned to unseen situations.

Scalability: An important feature of a learning system is an understanding of the relationship between system size and problem complexity; how rapidly does size of the system increase with the increase in problem complexity? The question of scalability is difficult to address given the complexity of the system although, there is evidence to suggest that that for XCS, the size increases as a *low-order polynomial* of the complexity of the problem rather than *with* the size of the complexity of the problem.

Figure 6 - Summary of the supportive arguments for LCS (from [8])

A. Michigan and Pittsburgh

The number of variations and types of classifiers is enormous, although given this enormity; two main streams of classifier research have emerged, demarcated by De Jong [13]. They are the classical (Holland) Michigan style classifier system and Pittsburgh learning classifier system of Smith [32,35].

Michigan Approach: (*classical*) The traditional approach based on the principle of using a genetic algorithm to develop new classifiers and relies on the bucket brigade algorithm to encourage systems of these classifiers to work together. Suitable for online learning in environments where radical changes in behaviour cannot be tolerated.

Pittsburgh Approach: (*Pitt*) An approach in which a population of classifiers are concatenated together to form an individual to be operated upon by the genetic algorithm. Suitable for offline learning in environments where larger changes in behaviour may be tolerated.

Figure 7 - Summary of the two main learning classifier systems

Wilson proposed two variations of the Michigan style learning classifier which have become ubiquitously employed in the field. The ZCS [33] which is a simpler variation for the system for investigation, and the XCS [34,38] which is an effective classifier for general application.

ZCS: (*zeroth-level classifier system*) A simpler variation of the Michigan style classifier system in which there is no message list, and a Q-learning like reinforcement learning algorithm is used called QBB.

XCS: (*accuracy-based*) An archetypal application LCS (with accurate and maximally general classifiers) in which the bucket brigade algorithm is replaced with an adaptation of Q-learning (different to the hybrid approach used in ZCS). In addition, credit is assigned based upon the *accuracy* (usefulness) of classifiers (as opposed to the predicted reward). The genetic algorithm is applied to subsets of classifiers (called environmental niches) that apply to the same situations (rather than the entire population).

Figure 8 - The two variations of the Michigan learning classifier system

In [11], Wilson summarises his two major tensions (problems) with classical learning classifiers. (1) The trade-off between the *competition* between classifiers and the *cooperation* needed for chains of classifiers to form. (2) The trade-off between the *generality* of classifiers with the *performance* gains of specific classifiers. His proposed solution (in the XCS learning

classifier) is to employ a niching approach for matching classifiers, and use fitness based on accuracy, such that the niches do not over-generalize. He claims that the combination of these two features overcome the limitations of the first 20 years of learning classifier research.

Problem	Solution
<i>Cooperative vs. Competitive</i>	GA amongst similar classifiers
<i>Performance vs. Generality</i>	Fitness based on accuracy

Table 1 - Summary of the problems and the their solutions in XCS

B. Reinforcement Learning

Reinforcement learning (RL) is an area of study in the field of machine learning and is characterised by system (or agent) learning to perform a task through a process of *trial-and-error* in an environment that provides minimal feedback in the form of *delayed* rewards. Sutton and Barto [29] provide a seminal treatment of reinforcement learning. Kaelbling, et al, [18] provide a seminal treatment of Temporal Difference learning (TD) that includes techniques such as adaptive heuristic critic, Q-learning and TD(lamda).

Moriarty, et al. [1] review the role of evolutionary algorithms in reinforcement learning, referring to the field of study as Evolutionary Algorithm for Reinforcement Learning (EARL). They suggest two main thrusts of RL research; (1) value function methods that search a function space (such as TD), and (2) evolutionary methods that search policy space. Searching policy space involves using search operators to modify explicit representations of policies. In searching value function space, no explicit policies are represented, rather the process involves learning a function that maximizes rewards. The two approaches are complementary, with differing objectives and suitable applications. Generally, the EARL approach pays less attention to individual decisions than TD learning, and provides a robust path to designing good polices in the face of noisy and incomplete information.

From a reinforcement learning perspective, learning classifiers (as a holistic system) are a policy searching approach in which a policy is decomposed into a number of subtasks. The search process operates upon the distributed representation (rather than a monolithic policy), providing finer granularity of the search over the subtasks. In addition, the distributed representation facilitates the integration of fine-grained domain knowledge regarding the sub-tasks.

In addition, see Kovacs [40] on the two perspectives of learning classifier systems: evolutionary algorithms and reinforcement learning.

C. Addendum

Learning classifiers evolve a interconnected set of rules that allow an agent to take actions in a domain through a process of forward chaining a parallel rule set. This permits the system to execute sequences of actions in chains.

An important feature of LCS is their ability to develop overlapping sets of rules called '*default*

hierarchies' [10] (also see [30]). Default hierarchies increase rule set parsimony, enlarge the solution set, and lend themselves to graceful refinement by the genetic algorithm [27]. They allow default (general) rules that are mostly correct to be enhanced through exception (specific) rules. LCS organizes default hierarchies by favouring the more specific exception rules over the more general default rules in the bucket brigade credit assignment scheme, and during competition.

A default hierarchy is an abstract concept to describe rule-complexes that form in which from a high-level a general concept in the domain is described, and from a low-level, specific cases are described. From a cognitive perspective, a default hierarchy is a quasi-homomorphic (overlapping map) model of the world, which is typically a more compact (less rules) than a homomorphic (non-overlapping structure preserving map) model [7,20].

Finally, ALECSYS is a parallel learning classifier system in which a problem is explicitly decomposed into a number of tasks, and a classifier is assigned to each task, thus a hierarchy of LCS are employed [23]. The architecture facilitates low-level parallelism at the level of a single LCS, and high-level parallelism in that multiple LCS work together on independent tasks to address a large problem.

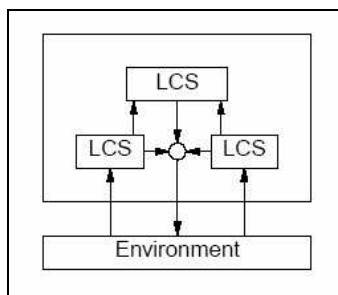


Figure 9 – Depiction of a distributed LCS (taken from [1] pg. 253)

A 'switch architecture' is used, such that messages and responsibility are discriminated and allocated to sub-LCS. Behaviour shaping is employed to train the switch architecture where two distinct learning processes may be used: (1) a holistic approach allows responsibility to emerge, and (2) a modular approach allows individual systems to be trained, then frozen whilst the discriminator system is trained. The architecture was used employed for autonomous robot tasks both in simulation and with real-robots [21]. From this work, Dorigo and Colombetti propose a methodology for engineering behaviours in autonomous robots called Behaviour, Analysis, Training (BAT), with ALECSYS as a representative implementation [19,22].

ACKNOWLEDGMENTS

Tim Hendtlass for his patience and for providing useful feedback on drafts of this paper

REFERENCES

[1] D. E. Moriarty, A.C. Schultz, and J.J. Grefenstette, Evolutionary Algorithms for Reinforcement Learning *Journal of Artificial Intelligence Research*, vol. 11, pp. 199-229, 1999.

[2] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*, USA, Canada: Addison Wesley Publishing Company, Inc., 1989.

[3] David Edward Goldberg, Computer-Aided GAS Pipeline Operation Using Genetic Algorithms and Rule Learning 1983. University of Michigan.

[4] J. H. Holland. Adaptation. In: *Progress in Theoretical Biology IV*, eds. R. Rosen and F. Snell. Academic Press, 1976.pp. 263-293.

[5] J. H. Holland, Adaptive algorithms for discovering and using general patterns in growing knowledge-bases *International Journal of Policy Analysis and Information Systems*, vol. 4, pp. 217-240, 1980.

[6] J. H. Holland, "Concerning the emergence of tag-mediated lookahead in classifier systems," *Proceedings of the ninth annual international conference of the Center for Nonlinear Studies on Self-organizing, Collective, and Cooperative Phenomena in Natural and Artificial Computing Networks on Emergent computation*, Los Alamos, New Mexico, United States, pp. 188-201, 1990.

[7] J. H. Holland, K. J. Holyoak, R. E. Nisbett, and P. R. Thagard. Classifier systems, Q-morphisms, and induction. In: *Genetic Algorithms and Simulated Annealing*, ed. L. D. Davis. London: Pittman, 1987.pp. 116-128.

[8] J. H. Holmes, P. L. Lanzi, W. Stolzmann, and S. W. Wilson, Learning classifier systems: New models, successful applications *Information Processing Letters*, vol. 82, pp. 23-30, Apr. 2002.

[9] John H. Holland and Judith S. Reitman, Cognitive systems based on adaptive algorithms *ACM SIGART Bulletin*, vol. 63, pp. 49-49, Jun, 1977.

[10] John H. Holland, Keith J. Holyoak, Richard E. Nisbett, and Paul R. Thagard. *Induction : processes of inference, learning, and discovery*, Cambridge, Mass, USA: MIT Press, 1986.

[11] John H. Holland, Lashon B. Booker, Marco Colombetti, Marco Dorigo, David E. Goldberg, Stephanie Forrest, Rick L. Riolo, Robert E. Smith, Pier Luca Lanzi, Wolfgang Stolzmann, and Stewart W. Wilson. What is a Learning Classifier System? In: *Learning Classifier Systems: From Foundations to Applications*, Anonymous Berlin / Heidelberg: Springer, 2000.pp. 3

[12] John Henry Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, USA: MIT Press, 1992.

[13] Kenneth de Jong, Learning with Genetic Algorithms: An Overview *Machine Learning*, vol. 3, pp. 121-138, Oct, 1988.

[14] L. B. Booker, D. E. Goldberg, and J. H. Holland, Classifier systems and genetic algorithms *Artificial Intelligence*, vol. 40, pp. 235-282, Sep, 1989.

[15] Larry Bull. *Applications Of Learning Classifier Systems*, Springer, 2004.

[16] Larry Bull and Tim Kovacs. *Foundations of Learning Classifier Systems (Studies in Fuzziness and Soft Computing)*, Springer, 2005.

[17] Lashon Bernard Booker, Intelligent behavior as an adaptation to the task environment 1982. University of Michigan.

[18] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore, Reinforcement learning: a survey *Journal of Artificial Intelligence Research*, vol. 4, pp. 237-285, 1996.

[19] M. Colombetti, M. Dorigo, and G. Borghi, Behavior analysis and training-a methodology for behavior engineering *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 26, pp. 365-380, Jun, 1996.

[20] Marco Dorigo, "New Perspectives about Default Hierarchies Formation in Learning Classifier Systems," *Proceedings of the 2nd Congress of the Italian Association for Artificial Intelligence on Trends in Artificial Intelligence*, pp. 218-227, 1991.

[21] Marco Dorigo, Alecsys and the AutoMouse: Learning to control a real robot by distributed classifier systems *Machine Learning*, vol. 19, pp. 209-240, Jun, 1995.

[22] Marco. Dorigo and Marco Colombetti. *Robot Shaping: an experiment in behavior engineering*, USA: MIT Press, 1998.

[23] Marco Dorigo, E. S., "Alecsys: A Parallel Laboratory for Learning Classifier Systems," *Proceedings of the 4th International Conference on Genetic Algorithms*, San Diego, CA, USA, pp. 296-302, 1991.

[24] Martin V. Butz. *Anticipatory Learning Classifier Systems (Genetic Algorithms and Evolutionary Computation)*, Springer, 2002.

[25] Pier Luca Lanzi and Rick L. Riolo. A Roadmap to the Last Decade of Learning Classifier System Research. In: *Learning Classifier Systems: From Foundations to Applications*, Anonymous Berlin / Heidelberg: Springer, 2000.pp. 33

- [26] Pier Luca Lanzi, Wolfgang Stolzmann, and Stewart W. Wilson. *Learning Classifier Systems: From Foundations to Applications*, Berlin / Heidelberg: Springer, 2000.
- [27] R. E. Smith. D. E. Goldberg, "Variable Default Hierarchy Separation in a Classifier System," *Foundations of Genetic Algorithms*, Bloomington campus, Indiana University, pp. 148, 1991.
- [28] Richard K. Belew and Stephanie Forrest, Learning and programming in classifier systems *Machine Learning*, vol. 3, pp. 193-223, Oct, 1988.
- [29] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*, Cambridge, Massachusetts; London, England: The MIT Press, 1998.
- [30] Rick L. Riolo, "The emergence of default hierarchies in learning classifier systems," *Proceedings of the third international conference on Genetic algorithms*, George Mason University, United States, pp. 322-327, 1989.
- [31] S. Forrest, Study of parallelism in the Classifier system and its application to classification in KL-ONE semantic networks 1985. Michigan University.
- [32] S. Smith, "Flexible Learning of Problem Solving Heuristics Through Adaptive Search," *Proceedings 8th International Joint Conference on Artificial Intelligence*, West Germany, pp. 422-425, 1983.
- [33] S. W. Wilson, ZCS: A Zeroth Level Classifier System *Evolutionary Computation*, vol. 2, pp. 1-18, Spring, 1994.
- [34] S. W. Wilson, Classifier Fitness Based on Accuracy *Evolutionary Computation*, vol. 3, pp. 149-175, 1995.
- [35] Stephen Frederick Smith, A learning system based on genetic adaptive algorithms 1980. Department of Computer Science, University of Pittsburgh.
- [36] Stewart W. Wilson, "Knowledge Growth in an Artificial Animal," *Proceedings of the 1st International Conference on Genetic Algorithms*, pp. 16-23, 1985.
- [37] Stewart W. Wilson, Classifier systems and the animat problem *Machine Learning*, vol. 2, pp. 199-228, Nov, 1987.
- [38] Stewart W. Wilson. State of XCS Classifier System Research. In: *Learning Classifier Systems: From Foundations to Applications*, Anonymous Berlin / Heidelberg: Springer, 2000, pp. 63
- [39] Stewart W. Wilson and David E. Goldberg, "A critical review of classifier systems," *Proceedings of the third international conference on Genetic algorithms*, George Mason University, United States, pp. 244-255, 1989.
- [40] Tim Kovacs, "Two Views of Classifier Systems," *Advances in Learning Classifier Systems : 4th International Workshop, IWLCS 2001 (Revised Papers)*, San Francisco, CA, USA, pp. 74, 2002.