A Survey of Formal Theoretical Advances Regarding XCS

David Pätzel*
University of Augsburg
Organic Computing Group
Augsburg, Germany
david.paetzel@informatik.uni-augsburg.de

Anthony Stein*
University of Augsburg
Organic Computing Group
Augsburg, Germany
anthony.stein@informatik.uni-augsburg.de

Jörg Hähner
University of Augsburg
Organic Computing Group
Augsburg, Germany
joerg.haehner@informatik.uni-augsburg.de

ABSTRACT

Learning Classifier Systems (LCSs) are a unique machine learning paradigm. The probably most well-known and investigated instance of these is XCS. LCSs, and with them, XCS, have developed in parallel to mathematically more rigorously founded paradigms such as today's reinforcement learning. This is probably the reason why XCS was initially defined without a formal basis. Nevertheless, the pursuit of a formal understanding of XCS has been one of the primary goals since its invention. Over the years, this led to a large and seemingly underestimated body of formal analysis of it. We present our try at a comprehensive overview of the various angles from which XCS was regarded formally. With this paper, we aim at (1) mitigating the misconception we sometimes observed that research on XCS contains some sort of 'formal theory gap', (2) supporting researchers interested in formal advances regarding XCS and (3) identifying future research directions.

CCS CONCEPTS

• **Theory of computation → Design and analysis of algorithms**; • **Computing methodologies → Machine learning**; **Rule learning**;

KEYWORDS

XCS, Learning Classifier Systems, Evolutionary Machine Learning, Formal Theory, Formalisation, Hyperparameter

ACM Reference Format:
David Pätzel, Anthony Stein, and Jörg Hähner. 2019. A Survey of Formal Theoretical Advances Regarding XCS. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3319619.3326848

1 INTRODUCTION

Learning Classifier Systems (LCSs) look back on more than 40 years of research since their invention by Holland; their first literature occurrence was in 1976 [35]. As an at that time entirely new and intriguing concept, Holland's ideas were adopted quickly which led to the emersion of a whole new research direction.

Two decades later, Wilson revolutionized the field by devising the XCS classifier system [68] which grew to be the most well-known and investigated LCS today [44, 57, 65]. Initially, XCS was merely defined algorithmically without a formal basis [23, 68]. Nevertheless, there has been an effort of furthering a formal understanding of this algorithm from the very start. This resulted in a large number of publications using formal theory to analyse different aspects of the system.

1.1 Goal and approach

With this paper, we pursue three main contributions: We want to

(1) show that there is a lot of formal theory on XCS and thus mitigate a misconception we observed, namely, that there exists some ominous 'formal theory gap' in LCS research in general and in XCS research in particular,
(2) provide a starting point for researchers new to the field that are interested in formal advances in the domain of XCS, as well as
(3) identify future research directions.

In order to achieve these goals, we present our attempt at a comprehensive overview of publications containing some sort of formal analysis of XCS or its parts. We deliberately restrict ourselves on *research regarding XCS*, since that system forms the basis for several derivatives which have achieved competitive state-of-the-art results in several application domains such as biomedical engineering [cf. e. g. 36, 66] and general supervised learning tasks [cf. e. g. 3, 52]. Because of our focus on *formal* work, we exclude various equally important extensions to the original system that were developed in a less or entirely non-formal way (i. e. heavily relying on empirical results, not using formal reasoning etc.).

Through its structure, our work is meant to facilitate a more precise identification of aspects that should be investigated further in the future or where a combination of existing facets is possible as well as to help to uncover any actually existing gaps.

1.2 Structure of this paper

To increase its usefulness, our overview is not merely structured chronologically but into sections with different topics. To accomplish this, we sorted publications dealing with similar concepts into groups; some publications are part of several such groups as these groups, to some extent, build upon each other. This segmentation into groups is reflected by the sections succeeding Section 2, where we give an overview of other literature reviews on LCSs.

*Corresponding author.

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3326848

Note that at several places we provide more than one reference; this is not to artificially blow up the bibliography but rather to reach our goal of being truly comprehensive and to refer the reader to other sources extending the content or looking at it from a different angle. This happens especially if a result was first presented at a workshop or conference and later reiterated as part of a more extensive book, book chapter or journal article.

2 FORMAL THEORY IN LCS LITERATURE REVIEWS

Before starting out with our own literature review on formal XCS theory, we want to appreciate existing literature reviews as well as more holistic surveys on LCSs in general. The overall number of review contributions is reasonable and each is coming with its own specific perspective on looking at achievements and historical developments of LCS research. Note that we do not restrict ourselves to XCS surveys in this particular section.

One of the first reviews was written by Wilson and Goldberg in 1989 [72]. Besides numerous suggestions regarding algorithmic details, the authors take a critical look at the preceding decade of LCS research. They summarize the progress made from the inception of *classifier systems*[1] in 1971 up to the late 1980s. Although certain aspects that need a deeper understanding are discussed, this work barely mentions insights based on formal analysis except for a somewhat unremarkable derivation of bids made by so-called *default* and *exception classifiers* in the formation of *default hierarchies*.

Ten years later, Lanzi and Riolo [45] published a follow-up. Once more, the authors outline the preceding decade up to 1999 and elaborate on the main achievements as well as the main research directions that became apparent throughout that time. They include a section on advanced topics where they summarize the first formal advances on generalization, scalability and successful maintenance of long action chains.

Shortly after, Wilson gave an overview of advances more specific to XCS [71]. In this publication, he emphasizes the relation between XCS and the mathematically well-grounded RL framework (especially, Q-learning). Additionally, he points out the research on generalization and population optimality available at the time and sketches a 'very tentative' theory for the learning time complexity of XCS.

In 2005, a book titled *Foundations on Learning Classifier Systems* was released [10]; acknowledging that the interaction of genetic algorithms (GAs) and reinforcement learning (RL) lacked understanding severely, its goal was to 'bring together current work aimed at understanding LCS in the hope that it will serve as a catalyst to a concerted effort to produce such understanding'. Several of the book's chapters are reviewed in the remainder of this paper. Also, its introduction chapter provides an overview of earlier research on the foundations of LCS. However, the authors do not discern between formal and non-formal theoretical work and, since they review literature on understanding LCS in general, XCS only plays a secondary role. Furthermore, said book was published over a decade ago; since then, many new insights have been gained.

[1]According to Bull [9] the prefix *learning* was used not until Goldberg's work in 1985 [34].

Two years later, in 2007, Sigaud and Wilson issued a survey article that focused on the notion of LCSs as generalizing systems that solve multi-step[2] problems and can be brought in line with the common RL notion [57]. Following a short historical outline, the major systems that constitute milestones in RL-related LCS development are presented: The *Zeroth-level Classifier System* (ZCS), XCS and the *Anticipatory Classifier System* (ACS). Besides relating LCSs to RL, this survey does not provide a dedicated section about formal theoretical advances.

Only one year later, Lanzi published another review article [44]. This article can be seen as the second follow-up to Wilson and Goldberg's initial review as it once again recapitulates the preceding decade of LCS research. After giving a brief retrospect, Lanzi elaborates on the general characteristics that underlie all LCSs. The section called 'Is there any theory?' is especially interesting for the present paper as it briefly mentions a number of formal accomplishments. However, that section gets lost a little bit in the shuffle of the many more topics covered.

The only other review we found (and that we are aware of) about LCSs that directly gives an overview of formal advances is another 2008 article by Bacardit et al. [1]. However since their goal is to look at the whole LCS field including the Pittsburgh-style systems, the part on formal work understandably plays a minor role and does not include everything there was at the time.

In the following year, Urbanowicz and Moore wrote another review article on LCSs [65]. One distinguishing aspect of this work is that it comprehensively lists descendants of the LCS family that were developed since its foundation in the mid 1970s. Additionally, it presents a so-called component roadmap that provides a generic LCS, a system abstracted from more specific (Michigan-style) LCS variants. Whereas LCS derivatives that have been explicitly created to facilitate theoretical analysis are mentioned, this review does not handle formal analysis of XCS as a central topic.

Another article that needs to be considered here is the review paper Bull published in 2015 [9]. It gives an overview of the historical development of LCSs from the initial *Cognitive System One* (CS-1) up to XCS and more recent derivatives such as *XCS for function approximation* (XCSF) or XACS. A family tree showing which system influenced which is included as well as schematics for the most influential ones. In each of the corresponding sections, the author also briefly refers to formal investigations.

The most recent work we want to refer to is the introductory book on LCSs released by Urbanowicz and Browne in 2017 [64]. This book constitutes a short but comprehensive introduction to the basics behind modern LCSs and contains several sections mentioning formal insights and their implications.

3 EARLY CONSIDERATIONS

After Wilson had laid out the foundations for XCS with his work on it as an LCS whose genetic search is guided by rule accuracy instead of strength [68], it did not take long until there was at least a minimum of analytical considerations. While some of these were not strictly formal, they altogether can be said to have paved the

[2]Another term for this is *sequential*; however, *multi-step* is more commonly used in LCS research.

way for later, more formal, work. This early analytical work, in parts formal, in parts not, is presented in the following sections.

3.1 The two hypotheses

In the very first paper about XCS, Wilson proposes what he calls the *generalization hypothesis* which states that the combination of the evolutionary pressures in XCS pushes towards more *accurate* and *maximally general* classifiers at the same time [68]. While Wilson does not corroborate his hypothesis formally, his reasoning about it is conclusive and can be said to have been the motivation if not have formed the basis for much of the later formal work on the evolutionary pressures in XCS (see Section 3.3). The same publication also contains a first—again, informal—attempt at an analogy with Q-learning.

Besides classifier accuracy and maximal generality, a third important property of classifier populations exists: *optimality*, that is, in addition to fulfilling the other two properties, using the smallest possible set of classifiers. Kovacs's *optimality hypothesis* states that XCS is able to achieve this reliably as well under certain conditions [37, 38]. Just like Wilson's generalization hypothesis, this hypothesis was not proven formally either; Kovacs derived it from a number of observations and experiments.

3.2 Overgeneralization

In the years directly after XCS's invention, there had been little to no formal work as to why an accuracy-based LCS like XCS often times works so much better in practice than earlier strength-based LCS. This changed with Kovacs's work on comparing the two directly [39]: He recounts (and restructures) several already known or suspected disadvantages and advantages of XCS regarding strength-based LCSs. In doing so, he loosely defines the problem of *overgeneral classifiers* that strength-based LCSs suffer from and explains why it does not occur in XCS. Although this is not done formally either, he makes a very strong point.

In a later work, Kovacs builds upon this by defining overgeneral classifiers formally as well for both strength-based and accuracy-based LCSs [40]. Based on this definition, he evaluates the fitness landscapes generated by the combination of the reward function and the chosen fitness scheme and proves a number of theorems about it.

3.3 Evolutionary pressures

A large part of the existing formal analysis of XCS is—directly or indirectly—based on analysing evolutionary pressures. Inspired from GA theory, this research direction is often called *facet-wise approach* since XCS's parts (e. g. the RL component or the GA) are first analysed each on its own in order to gain insights into the system as a whole [11, 12]. Butz and Pelikan were the first who went in this direction: In their seminal publication they distinguish and investigate five different *evolutionary pressures* that drive XCS (i. e. fitness, set, mutation, deletion and subsumption pressure) and reason why their interaction results in XCS evolving accurate but maximally general classifiers [20, 21]. The main tool they introduce to do so is a formal analysis of the changes in *classifier specificity* which they apply to the set, deletion and mutation pressure.

Somewhat more detail and more elaborate explanations are also provided by parts of other publications [17, 19]. Also, Butz's dissertation [11] and the corresponding book [12] encompass and summarize most if not all of the earlier theory on evolutionary pressures in XCS.

4 ANALYSIS OF THE GA IN XCS

Since one of the unique charasteristics of LCSs in general and XCS in particular is the GA used to optimize rule structure, it is not surprising that this mechanism has been the focus of several publications trying to analyse XCS formally.

A first step in this direction was made by Bull, who developed a Markov model of the GA and used it to show formally that an accuracy-based GA is more stable than its strength-based counterpart on single-step tasks [4, 6].

He also provides an analysis of the multi-step setting—albeit with a different focus [5, 6]: The main result there is that if which behaviour is optimal in one niche is dependant on the chosen behaviour in another niche the selection pressure can become very unstable.

In the course of extending XCS with capabilities for integer- and continuous-valued input, several interval-based representations for the classifiers' conditions have been proposed. In 2003, Stone and Bull investigated the two, at that time, most prominent of these with regard to whether they lead to a bias in the GA and thus to overall performance differences [61, 62]. They conclude that this is the case with both and introduce another representation that does not suffer from this problem—the *unordered bound representation.*

Orriols-Puig et al. compare proportionate and tournament selection regimes in XCS's GA [54]. They perform a takeover time analysis (a technique from GA theory) and derive that both in theory and in practice the tournament selection regime can be expected to be the more robust of the two.

Whereas early LCSs mostly relied on a panmitic GA, XCS and other newer LCSs employ a niche GA which is usually thought to be superior in learning tasks. As a starting point to proving—or at least analysing—this superiority, Kovacs and Tindale derive a classifier's selection probability for both variants and manage to formally quantify both the fitness bonus resulting from higher rule generality as well as the fitness penalty resulting from rule overlap in niche GAs. [42]

5 HYPERPARAMETER DERIVATION

XCS stars several hyperparameters (which we just call *parameters* in the following): The algorithmic description [23], which only deals with the most basic, binary string input, defines 20 already—more elaborate derivatives capable of dealing with vectors of real input add even more on top of that [70]. A number of these parameters have to be configured correctly; otherwise XCS may learn only slowly or even fail to learn at all (e. g. due to a cover-delete cycle [e. g. 16, 19]). Since XCS's behaviour is so very dependent on a good configuration, several questions arise: Are there bounds on these parameters whose violation results in bad performance? Is it possible to derive correct parameter settings from properties of or assumptions about the task to solve and XCS's algorithmic structure? There has been put quite some effort into investigating these and related questions formally; the resulting publications are presented in the upcoming sections.

5.1 Learning challenges

The first direction that formal research into XCS's parameters took was what Butz initially called *learning challenges* [11–13, 18, 19] from which then parameter *bounds* were derived (see the next section). Most of the theory on challenges and bounds is based directly on the analysis of the evolutionary pressures in XCS which was already mentioned in Section 3.3. Note that in later publications, the term challenge is hardly used any more; instead, discussions only revolve around bounds [e. g. 58].

Challenges are preconditions that need to be fulfilled for XCS to even have a chance to reliably solve a learning problem. Before focus shifted towards the bound-oriented view, two such challenges were defined and investigated [11–13, 18, 19]:

- The *covering challenge* is met by an instance of XCS for a certain learning task if any possible incoming input is covered by at least one classifier in a random population this XCS instance could generate. Failing to fulfil this precondition results in a cover-delete cycle inhibiting the GA which makes learning practically impossible.
- The *schema challenge* is about XCS reliably generating classifiers that are part of a population that is a decent solution candidate. Not solving this challenge means that XCS tends to get stuck in a local optimum regarding rule structure.

5.2 Parameter bounds

Based on classifier specificity analysis (see Section 3.3) and the learning challenges mentioned in the previous section, a number of parameter bounds were derived. Most of these relate the parameter for the maximum population size N to a learning task's difficulty (i. e. in terms of dimensionality or niche characteristics); however, by transposing the resulting equations or by combining them with others, bounds for some of the other parameters or for properties of the learning problem can be derived as well.

The covering challenge already contains a bound for N which is why itself is often simply called *covering bound* in later work [11, 12]. In addition to that, three more bounds have been derived:

- The *schema bound* for N (sometimes called *representative bound*) ensures that classifiers with a high enough specificity are generated [11–13, 16, 17]. This also entails a bound for the mutation rate parameter μ given a learning task's difficulty and some temporal threshold at which a certain rule structure should be present in the population.
- The *reproductive opportunity bound* for N guarantees that accurate classifiers get a chance to reproduce before being deleted [11–13, 16, 17].
- The *niche support bound* for N ensures that XCS does not forget what it has learned [11, 12, 15, 16, 22].

This concludes what we consider to be the seminal work on the theory of learning challenges and bounds. The remainder of this section shortly summarizes extensions for certain cases as well as applications to specific XCS derivatives.

Orriols-Puig et al. regard XCS's performance on learning tasks with class (or niche) imbalances; in the process, they analyse learning challenges and bounds [49, 51, 55]. For example, they derive a bound for N [51, 55] as well as an upper bound on the imbalance ratio between the different classes which in theory ensures learnability [49]. They also extend their approach to XCS with real-valued inputs (usually called XCSR) [53].

Another derivative of XCS is XCSF, a real-valued function approximator used for supervised learning (i. e. regression). Because of its design, it is possible to derive the optimal value of N for XCSF (if rules are required to be non-overlapping), which is what Stalph et al. did [59]. They also transfer parts of the aforementioned challenges and bounds defined for the original XCS setting to XCSF and elaborate on three aspects that have a significant impact on learning performance, i. e. the representational power of the rules, the learning problem's complexity as well as the learning capacity in terms of the population size and the number of training instances [58, 60].

Debie and Shafi use techniques similar to the challenges and bounds developed by Butz et al. to reason about sampling if the input space is high-dimensional and thus the *curse of dimensionality* impedes learning [24]. While they do not regard XCS itself but its derivative for supervised learning, the *Supervised Learning Classifier System* (UCS), their findings seem to be applicable to either system.

5.3 Parameter selection and online parameter adaptation

Whereas the previous section looked at bounds on parameters, this section lists several, more practically oriented publications that give formally grounded guidelines for actually selecting certain parameters. Often, online parameter adaptation schemes have been derived from such guidelines which shall find mention here as well.

Orriols-Puig et al. derive optimal values for the learning rate β and the GA threshold θ_{GA} in the face of class imbalance [49–51]. They also develop an online adaptation mechanism for the reproduction probabilities to account for such imbalances.

Nakata et al. establish a theory for deriving optimal parameter values for XCS's learning rate β, the target error ϵ_0 as well as the the GA threshold θ_{GA} [48]. The main assumptions made by them are that, for the task to solve, the *true* accuracy of the *best inaccurate* classifier has to be determinable and that a binary reward scheme is applied.

Subsequently, they continued their work and investigated the number of offspring classifiers produced by the GA [47]. Their results allow for extending XCS with means to reliably identify inaccurate classifiers in the population; these classifiers can be deleted without the risk of detrimental forgetting. This enables a scheme for dynamically adjusting the number of offspring classifiers that the GA creates via reproduction.

6 MODELS OF XCS

As mentioned before, XCS's first description by Wilson did not provide a formal model; instead a more verbal form was chosen for its definition [68]. Several years later, Butz and Wilson published an algorithmic description of the system [23] that is often seen as a second official definition offering more details. While such an algorithmic description is—mostly—unambiguous, it is not suited well for formal analysis [56]. Because of that, there have been several efforts to model XCS (or a simplified version of it) formally and then analyse that model instead of the original XCS algorithm directly. These are presented in the following sections.

6.1 Simplified models

Bull created *YCS*, a simplified version of XCS [7]. The main difference is that YCS sports a panmitic GA instead of XCS's niche GA leading to worse generalization capabilities. However, due to this deviation, the author is able to create an infinite population model of the resulting LCS.

In a later work, Bull compares YCS with the *Minimal Classifier System* (MCS) (the corresponding ZCS analogue) and extends it with a niche GA thus closing the gap to XCS [8]. For the analysis of the resulting LCS he relies on Butz's pressure theory that was already presented in Section 3.3.

Another approach using a simplified model is the one by Wada et al. who aim at transfering existing insights into the convergence of the well-known Q-learning algorithm to XCS and ZCS (both without a GA) [67]. They try to transform the update rules of these LCS and Q-learning (which they augment with function approximation capabilities) to a common representation. For XCS, this does not pan out and they conclude that XCS is inconsistent with the Q-learning derivative they consider—an implication that is not undisputed [25].

6.2 Full models and formalizations

An early achievement in terms of creating a complete formal model of LCSs is the work by Lanzi the result of which shows a high similarity to XCS [43]. He replaces the Q-table of the Q-learning algorithm with a population of LCS classifiers and then gradually develops a general LCS by adding generalization facilities; for those, he considers both concept learning as well as GAs. He discusses the advantages and drawbacks of these generalization methods and identifies GAs as being more general and presumably more efficient in practice.

Another holistic approach is the one pursued by Drugowitsch and Barry who intent to formalize LCSs in a machine learning–centric manner. First, they formalize the function approximation done by LCSs [27, 31]. The resulting model is restricted to function approximation (e. g. in the RL context, single-step value functions) and a fixed number of classifiers. Since their model utilizes an abstract feature vector, their formalization is more general than XCS (it also encompasses XCSF as well as other derivatives). In this first, slightly simplified setting, they are able to show that an alternative update method based on Kalman filters is superior to the ones employed by XCS and XCSF. The second part of their work consists of an investigation of the relationship and interaction between RL and their earlier function approximation framework [28]. It connects their formalization to dynamic programming methods (e. g. value iteration) and temporal difference learning methods (e. g. Q-learning) in a natural way. Another topic considered in the process are general techniques for mixing local models to a global one [30]; this work includes a formal examination of the mixing method of XCS. Together with Loiacono et al., the authors also examine the error estimation of XCSF based on their function approximation model [46]. While Drugowitsch and Barry had initially planned to formalize classifier replacement as well—which would have completed their model—this endeavour proved to be unsuccessful since there could not be found a formal definition of an optimal set of classifiers leaving the optimization process of classifier replacement without a target [25, 26]. Because of that, Drugowitsch switched to a probabilistic, model-centred view which was the topic of his dissertation [25] and the corresponding book publication [26] and which is presented next.

Drugowitsch starts out with a formal description of the relevant types of learning problems and, based on that, defines an LCS's model as a probabilistic set of classifiers. This model can be trained using common methods from adaptive filter theory and statistical machine learning. The target of LCS training should be an optimal model, that is, an optimal set of classifiers which he defines based on Bayesian model selection; to be able to perform that selection, he augments his LCS model with priors resulting in a Bayesian model that can be trained using variational Bayesian inference. In order to be able to solve multi-step tasks, this in turn is combined with RL. While the resulting learning method is very general (more general than XCS, i. e. it also encompasses XCSF, for example), neither of the two exemplary algorithmic implementations Drugowitsch eventually proposes in his work is a Michigan-style LCS like XCS; instead, he shows how to realize his learning method using a Pittsburgh-style LCS as well as Markov Chain Monte Carlo methods—leaving the more challenging Michigan-style realization for future work. [25, 26]

Drugowitsch's revised approach is applied to UCS by Edakunni et al. [32, 33]. The result is a working probabilistic system model based on the mixture of experts paradigm.

A more recent project with the goal of formalizing XCS relies on an algebraic model which was created using functional programming [56]. While this work is still in its infancy, the authors' proclaimed goal is to close the gap between the aforementioned theory created by Drugowitsch and Barry and the renowned algorithmic description of XCS by Butz and Wilson.

7 LEARNING PROBLEM ANALYSIS

Another direction that XCS research took was analysing learning problems. The overall idea of this is that by studying learning problems and whether XCS is able to solve them reliably, insights about its behaviour can be gained. The larger part of the existing research on this topic is not formal but mainly empirical; this is why even though we present the more formal publications of this direction, these are probably the least formal ones we are considering in our review.

A very early discussion of the complexity of certain tasks (i. e. multiplexer problems) can be found in a paper by Wilson [69]. He speculates conclusively that their difficulty is polynomial in the number of accurate, maximally general classifiers; however, he bases his suspicion mainly on a number of empirical observations and defers a more in-depth analysis of it to future work.

The paper introducing the abovementioned challenges (see Section 5.1) identified two properties of the reward functions of a class of learning tasks that inherently solve these challenges: *layered payoff* and *biased generality* [18].

Kovacs and Kerber identify and compare several measures for the difficulty of single-step problems in binary representation [41]. Their findings result in the proposition of a small test suite that challenges XCS with learning tasks of different difficulty.

According to Tharakunnel et al., early benchmark tasks for XCS could have been solved using a mutating GA without crossover operator [63]. Based on two properties of learning problems, they show why these learning tasks do not really challenge XCS's generalization capabilities and devise simple alternatives that XCS can barely solve because of them being *accuracy-misleading* (i. e. concatenated multiplexer problems). They conclude that the accuracy definition has to be changed in order to be able to reliably and scalably solve these tasks.

Bernadó-Mansilla and Ho develop a methodology to characterize the complexity of classification problems using a set of geometrical descriptors [2]. They apply the result to several real-world classification problems and find correlations between XCS's performance and certain properties of the learning task. Besides that, they are able to identify learning problem characteristics which make XCS advantageous over other, more traditional classification schemes (e. g. nearest neighbour methods and decision trees).

The abovementioned work by Stone and Bull also discusses the real multiplexer problem with regard to its suitability as a benchmark problem [61, 62]. They conclude that it is inappropriate because there seem to be unwanted interdependencies with the interval representation (especially with the centre-spread representation) resulting in a positive sampling bias which 'relieve[s] the other mechanisms of XCS from much of the burden of solving the real multiplexer problem because the solution to the problem happens to match the nature of the classifiers being generated' [61, 62].

In the course of their analysis of XCS's performance on learning tasks with class imbalances, Orriols-Puig et al. develop two benchmark problems that enable an easy control of the complexity introduced by the imbalance ratio [51, 55].

8 CONVERGENCE AND TIME BOUNDS

An important question that has to be answered for RL methods is whether they converge and, if they do so, to the desired value. For XCS, this question has not been answered conclusively—arguably mainly due to the complexity introduced by the combination of RL and the GA.

Butz et al. use a domino convergence model to derive a bound on the time until maximally accurate classifiers are found; their results indicate that XCS scales in a machine learning competitive way [11, 12, 14]. This work is built upon later when the same authors prove that XCS can PAC-learn k-DNF problems [11, 12, 16].

In their aforementioned work, Wada et al. conclude that the RL process of ZCS and an enhanced Q-learning with function approximation capabilities is equivalent (as long as ZCS's rule discovery process is suppressed or during the periods between rule discovery events) [67]. Although they try, they fail to prove something similar for XCS and argue that XCS is therefore 'inconsistent' with their Q-learning derivative (note that they do not, however, imply that this means that XCS does not converge). As mentioned before though, this is a controversial result [25].

Based on their earlier work [28], Drugowitsch and Barry prove convergence of their (then still slightly simplified) version of LCS if the mixing weights of the classifiers are fixed [29]. They also outline a proof for the case that the mixing weights depend on classifier accuracy (which is the case in XCS) but defer that to future work.

In his dissertation, Drugowitsch performs a convergence analysis of his already mentioned revised (and complete) formalization [25, 26]. However, he does not come to a final conclusion.

9 DISCUSSION

In this section we want to briefly discuss the state of formal theory on XCS and what we think should be the next steps.

We have shown with our work that there has happened a decent amount of formal theory research on XCS in the past. However, the larger part of that happened before 2010; since then, only few publications were made—despite there being still many open questions.

9.1 Models

In our opinion, some of the models that were presented in Section 6 are, although in parts unfinished, very promising. Drugowitsch's work [25, 26] is by far the most advanced of these—in its extent, in its generality and also in its prospect. Because of its sophistication it is definitely a candidate for more formal analysis; perhaps even for more proofs that finally lead towards stronger convergence assurances. Moreover, for identifying future research directions we recommend his well-written discussion of his results and their consequences for XCS and XCSF as well as the possible future research directions he presents—this goes for both future research built on his work directly and formal research on XCS in general.

9.2 Theory for real-valued learning problems

Large parts of the formal work on XCS is based on Butz's evolutionary pressures, challenges and bounds. The central assumption of those is that the learning problem is binary encoded; this simplification allows specific constructs (e. g. binomial distributions) or methods (e. g. Markov chain analysis) to be used. Unfortunately, these techniques are not available in the more general, real-valued setting (i. e. XCSR or XCSF)—at least not without prior work, see for example Stalph et al.'s transfer of some of the formal analysis of XCS to XCSF [58, 60].

To the best of our knowledge, up to now, only Orriols-Puig et al., Stalph et al., Loiacono et al. and Drugowitsch et al. have considered analysing XCS with real-valued input or output [e. g. 25–28, 30, 31, 46, 53, 58–60]. However, to be able to solve relevant—and thus real-world—learning tasks this feature is a must; its analysis is therefore badly needed and should be catched up on.

9.3 Multi-step tasks

The third major topic that has to be dealt with formally more in-depth are multi-step tasks since most of the formal research on XCS assumes a single-step setting. While there are some exceptions [e. g. 5, 6, 11, 12, 25, 26, 28], XCS's behaviour on the complete

RL setting has not yet been formally examined in full and there is still no conclusive answer (beyond empirical results) as to whether the original XCS is a stable multi-step RL learner.

9.4 Hyperparameter selection and elimination

In order to really compete with (and perhaps outdo) other well-known RL or ML methods such as Deep Q-learning, there also has to be more work on hyperparameter selection: In practice, there is still some arcane feeling to getting the hyperparameters right for a certain learning task since only a small number of them were analysed formally. Moreover, with all that we already know, it might be possible to eliminate some of the hyperparameters; for example, some parts of the work Nakata et al. recently provided can be understood that way [48].

10 SUMMARY AND FUTURE WORK

With this paper, we aimed at mitigating the misconception that XCS is not backed by formal theory. We found many XCS-related publications that conduct formal analysis but we also came upon several vital open questions; nevertheless our goal of attenuating that argument was hopefully still achieved. In the process, we provided researchers with a comprehensive survey on the most notable works regarding formal theory on XCS; to make the result usable as a reference text, we used—in our opinion—sensible sectioning. Our research also enabled us to identify four major research directions that deserve more attention in the future.

Our plans include to take another step towards a more thorough and more detailed review of the formal advances in the field of LCS research. We want to find a unifying notation that frames all (or most) of the formally backed artifacts, that is, definitions, proved propositions and formulated conjectures, into a tangible form.

REFERENCES

[1] Jaume Bacardit, Ester Bernadó-Mansilla, and Martin V. Butz. 2008. Learning Classifier Systems: Looking Back and Glimpsing Ahead. In *Learning Classifier Systems*, Jaume Bacardit, Ester Bernadó-Mansilla, Martin V. Butz, Tim Kovacs, Xavier Llorà, and Keiki Takadama (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 1–21.

[2] Ester Bernadó-Mansilla and Tin Kam Ho. 2005. Domain of Competence of XCS Classifier System in Complexity Measurement Space. *IEEE Transactions on Evolutionary Computation* 9, 1 (Feb 2005), 82–104.

[3] Ester Bernadó-Mansilla and Josep-Maria Garrell-Guiu. 2003. Accuracy-Based Learning Classifier Systems: Models, Analysis and Applications to Classification Tasks. *Evolutionary Computation* 11, 3 (2003), 209–238.

[4] Larry Bull. 2001. Simple Markov Models of the Genetic Algorithm in Classifier Systems: Accuracy-Based Fitness. In *Advances in Learning Classifier Systems*, Pier Luca Lanzi, Wolfgang Stolzmann, and Stewart W. Wilson (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 21–28.

[5] Larry Bull. 2001. Simple Markov Models of the Genetic Algorithm in Classifier Systems: Multi-step Tasks. In *Advances in Learning Classifier Systems*, Pier Luca Lanzi, Wolfgang Stolzmann, and Stewart W. Wilson (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 29–36.

[6] Larry Bull. 2002. On accuracy-based fitness. *Soft Computing* 6, 3 (2002), 154–161.

[7] Larry Bull. 2003. *A Simple Accuracy-based Learning Classifier System.* Technical Report UWELCSG03-005. Learning Classifier Systems Group, University of the West of England, Bristol.

[8] Larry Bull. 2005. *Two Simple Learning Classifier Systems.* Springer Berlin Heidelberg, Berlin, Heidelberg, 63–89.

[9] Larry Bull. 2015. A brief history of learning classifier systems: from CS-1 to XCS and its variants. *Evolutionary Intelligence* 8, 2 (Sep 2015), 55–70.

[10] Larry Bull and Tim Kovacs. 2005. *Foundations of Learning Classifier Systems: An Introduction.* Springer Berlin Heidelberg, Berlin, Heidelberg, 1–17.

[11] Martin V. Butz. 2001. *Rule-based Evolutionary Online Learning Systems: Learning Bounds, Classification, and Prediction.* Technical Report UIUCDCS-R-2004-2474. Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign.

[12] Martin V. Butz. 2006. *Rule-Based Evolutionary Online Learning Systems - A Principled Approach to LCS Analysis and Design.* Studies in Fuzziness and Soft Computing, Vol. 191. Springer Berlin Heidelberg, Berlin, Heidelberg.

[13] Martin V. Butz and David E. Goldberg. 2003. Bounding the Population Size in XCS to Ensure Reproductive Opportunities. In *Genetic and Evolutionary Computation — GECCO 2003*, Erick Cantú-Paz, James A. Foster, Kalyanmoy Deb, Lawrence David Davis, Rajkumar Roy, Una-May O'Reilly, Hans-Georg Beyer, Russell Standish, Graham Kendall, Stewart Wilson, Mark Harman, Joachim Wegener, Dipankar Dasgupta, Mitch A. Potter, Alan C. Schultz, Kathryn A. Dowsland, Natasha Jonoska, and Julian Miller (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 1844–1856.

[14] Martin V. Butz, David E. Goldberg, and Pier Luca Lanzi. 2004. Bounding Learning Time in XCS. In *Genetic and Evolutionary Computation – GECCO 2004*, Kalyanmoy Deb (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 739–750.

[15] Martin V. Butz, David E. Goldberg, Pier Luca Lanzi, and Kumara Sastry. 2007. Problem Solution Sustenance in XCS: Markov Chain Analysis of Niche Support Distributions and the Impact on Computational Complexity. *Genetic Programming and Evolvable Machines* 8, 1 (2007), 5–37.

[16] Martin V. Butz, David E. Goldberg, and Pier Luca Lanzi. 2005. *Computational Complexity of the XCS Classifier System.* Springer Berlin Heidelberg, Berlin, Heidelberg, 91–125.

[17] Martin V. Butz, David E. Goldberg, and Kurian Tharakunnel. 2003. Analysis and Improvement of Fitness Exploitation in XCS: Bounding Models, Tournament Selection, and Bilateral Accuracy. *Evolutionary Computation* 11, 3 (2003), 239–277.

[18] Martin V. Butz, Tim Kovacs, Pier Luca Lanzi, and Stewart W. Wilson. 2001. *How XCS Evolves Accurate Classifiers.* Technical Report 2001008. Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign.

[19] Martin V. Butz, Tim Kovacs, Pier Luca Lanzi, and Stewart W. Wilson. 2004. Toward a Theory of Generalization and Learning in XCS. *IEEE Transactions on Evolutionary Computation* 8, 1 (Feb 2004), 28–46.

[20] Martin V. Butz and Martin Pelikan. 2001. Analyzing the Evolutionary Pressures in XCS. In *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation (GECCO '01)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 935–942.

[21] Martin V. Butz and Martin Pelikan. 2004. *Analyzing the Evolutionary Pressures in XCS.* Technical Report 2001009. Department of Computer Science, University of Illinois at Urbana-Champaign.

[22] Martin V. Butz and Martin Pelikan. 2004. *Bounding the Population Size to Ensure Niche Support in XCS.* Technical Report 2004033. Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign.

[23] Martin V. Butz and Stewart W. Wilson. 2002. An algorithmic description of XCS. *Soft Computing* 6, 3 (Jun 2002), 144–153.

[24] Essam Debie and Kamran Shafi. 2017. Implications of the curse of dimensionality for supervised learning classifier systems: theoretical and empirical analyses. *Pattern Analysis and Applications* (Aug 2017), 1–18.

[25] Jan Drugowitsch. 2007. *Learning classifier systems from first principles: a probabilistic reformulation of learning classifier systems from the perspective of machine learning.* Ph.D. Dissertation. University of Bath (United Kingdom).

[26] Jan Drugowitsch. 2008. *Design and Analysis of Learning Classifier Systems - A Probabilistic Approach.* Studies in Computational Intelligence, Vol. 139. Springer Berlin Heidelberg, Berlin, Heidelberg.

[27] Jan Drugowitsch and Alwyn Barry. 2006. *A Formal Framework and Extensions for Function Approximation in Learning Classifier Systems.* Technical Report CSBU-2006-01. Department of Computer Science, University of Bath.

[28] Jan Drugowitsch and Alwyn Barry. 2006. *A Formal Framework for Reinforcement Learning with Function Approximation in Learning Classifier Systems.* Technical Report CSBU-2006-02. Department of Computer Science, University of Bath.

[29] Jan Drugowitsch and Alwyn Barry. 2006. *Towards convergence of learning classifier systems value iteration.* Technical Report CSBU-2006-03. University of Bath, Department of Computer Science.

[30] Jan Drugowitsch and Alwyn M. Barry. 2007. Mixing Independent Classifiers. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, Hod Lipson (Ed.). ACM, New York, NY, USA, 1596–1603.

[31] Jan Drugowitsch and Alwyn M. Barry. 2008. A formal framework and extensions for function approximation in learning classifier systems. *Machine Learning* 70, 1 (2008), 45–88.

[32] Narayanan Unny Edakunni, Gavin Brown, and Tim Kovacs. 2011. Online, GA based mixture of experts: a probabilistic model of UCS. In *13th Annual Genetic and Evolutionary Computation Conference, GECCO 2011, Proceedings, Dublin, Ireland, July 12-16, 2011*, Natalio Krasnogor and Pier Luca Lanzi (Eds.). ACM, New York, NY, USA, 1267–1274.

[33] Narayanan Unny Edakunni, Tim Kovacs, Gavin Brown, and James A. R. Marshall. 2009. Modeling UCS As a Mixture of Experts. In *Proceedings of the 11th Annual*

Conference on Genetic and Evolutionary Computation (GECCO '09). ACM, New York, NY, USA, 1187–1194.

[34] David E. Goldberg. 1985. Genetic algorithms and rule learning in dynamic system control. In *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*. Lawrence Erlbaum Associates, 8–15.

[35] John H. Holland. 1976. Adaptation. In *Progress in Theoretical Biology*, Robert Rosen and F.M. Snell (Eds.). Vol. 4. Academic Press, New York, 263–293.

[36] John H. Holmes and Jennifer A. Sager. 2005. Rule Discovery in Epidemiologic Surveillance Data Using EpiXCS: An Evolutionary Computation Approach. In *Artificial Intelligence in Medicine*, Silvia Miksch, Jim Hunter, and Elpida T. Keravnou (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 444–452.

[37] Tim Kovacs. 1996. *Evolving Optimal Populations with XCS Classifier Systems*. Master's thesis. University of Birmingham, Birmingham, UK.

[38] Tim Kovacs. 1998. XCS Classifier System Reliably Evolves Accurate, Complete, and Minimal Representations for Boolean Functions. In *Soft Computing in Engineering Design and Manufacturing*, P. K. Chawdhry, R. Roy, and R. K. Pant (Eds.). Springer London, London, 59–68.

[39] Tim Kovacs. 2000. Strength or Accuracy? Fitness Calculation in Learning Classifier Systems. In *Learning Classifier Systems*, Pier Luca Lanzi, Wolfgang Stolzmann, and Stewart W. Wilson (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 143–160.

[40] Tim Kovacs. 2001. Towards a Theory of Strong Overgeneral Classifiers. In *Foundations of Genetic Algorithms 6*, Worthy N. Martin and William M. Spears (Eds.). Morgan Kaufmann, San Francisco, 165–184.

[41] Tim Kovacs and Manfred Kerber. 2001. What Makes a Problem Hard for XCS?. In *Advances in Learning Classifier Systems*, Pier Luca Lanzi, Wolfgang Stolzmann, and Stewart W. Wilson (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 80–99.

[42] Tim Kovacs and Robin Tindale. 2013. Analysis of the Niche Genetic Algorithm in Learning Classifier Systems. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation (GECCO '13)*. ACM, New York, NY, USA, 1069–1076.

[43] Pier Luca Lanzi. 2002. Learning classifier systems from a reinforcement learning perspective. *Soft Computing* 6, 3 (Jun 2002), 162–170.

[44] Pier Luca Lanzi. 2008. Learning classifier systems: then and now. *Evolutionary Intelligence* 1, 1 (Mar 2008), 63–82.

[45] Pier Luca Lanzi and Rick L. Riolo. 2000. A Roadmap to the Last Decade of Learning Classifier System Research. In *Learning Classifier Systems, From Foundations to Applications*. Springer Berlin Heidelberg, Berlin, Heidelberg, 33–62.

[46] Daniele Loiacono, Jan Drugowitsch, Alwyn Barry, and Pier Luca Lanzi. 2008. Analysis and Improvements of the Classifier Error Estimate in XCSF. In *Learning Classifier Systems*, Jaume Bacardit, Ester Bernadó-Mansilla, Martin V. Butz, Tim Kovacs, Xavier Llorà, and Keiki Takadama (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 117–135.

[47] Masaya Nakata, Will Browne, and Tomoki Hamagami. 2018. Theoretical Adaptation of Multiple Rule-generation in XCS. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '18)*. ACM, New York, NY, USA, 482–489.

[48] Masaya Nakata, Will N. Browne, Tomoki Hamagami, and Keiki Takadama. 2017. Theoretical XCS Parameter Settings of Learning Accurate Classifiers. In *Proceedings of the Genetic and Evolutionary Computation Conference 2017 (GECCO '17)*, Peter A. N. Bosman (Ed.). ACM, New York, NY, USA, 473–480.

[49] Albert Orriols-Puig and Ester Bernadó-Mansilla. 2006. Bounding XCS's Parameters for Unbalanced Datasets. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation (GECCO '06)*. ACM, New York, NY, USA, 1561–1568.

[50] Albert Orriols-Puig and Ester Bernadó-Mansilla. 2008. *Mining Imbalanced Data with Learning Classifier Systems*. Springer Berlin Heidelberg, Berlin, Heidelberg, 123–145.

[51] Albert Orriols-Puig, Ester Bernado-Mansilla, David E. Goldberg, Kumara Sastry, and Pier Luca Lanzi. 2009. Facetwise Analysis of XCS for Problems With Class Imbalances. *IEEE Transactions on Evolutionary Computation* 13, 5 (Oct 2009), 1093–1119.

[52] Albert Orriols-Puig, Jorge Casillas, and Ester Bernadó-Mansilla. 2008. Genetic-based machine learning systems are competitive for pattern recognition. *Evolutionary Intelligence* 1, 3 (Oct 2008), 209–232.

[53] Albert Orriols-Puig, Xavier Llorà, and David E. Goldberg. 2010. How XCS Deals with Rarities in Domains with Continuous Attributes. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation (GECCO '10)*. ACM, New York, NY, USA, 1023–1030.

[54] Albert Orriols-Puig, Kumara Sastry, Pier Luca Lanzi, David E. Goldberg, and Ester Bernadó-Mansilla. 2007. Modeling Selection Pressure in XCS for Proportionate and Tournament Selection. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation (GECCO '07)*. ACM, New York, NY, USA, 1846–1853.

[55] Albert Orriols-Puig, Kumara Sastry, Pier Luca Lanzi, David E. Goldberg, and Ester Bernadó-Mansilla. 2007. Modeling Selection Pressure in XCS for Proportionate and Tournament Selection. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation (GECCO '07)*. ACM, New York, NY, USA, 1846–1853.

[56] David Pätzel and Jörg Hähner. 2018. An Algebraic Description of XCS. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO '18)*. ACM, New York, NY, USA, 1434–1441.

[57] Olivier Sigaud and Stewart W. Wilson. 2007. Learning Classifier Systems: A Survey. *Soft Computing* 11, 11 (2007), 1065–1078.

[58] Patrick O. Stalph and Martin V. Butz. 2010. Current XCSF Capabilities and Challenges. In *Learning Classifier Systems*, Jaume Bacardit, Will Browne, Jan Drugowitsch, Ester Bernadó-Mansilla, and Martin V. Butz (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 57–69.

[59] Patrick O. Stalph, Martin V. Butz, David E. Goldberg, and Xavier Llorà. 2009. On the Scalability of XCS(F). In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation (GECCO '09)*. ACM, New York, NY, USA, 1315–1322.

[60] Patrick O. Stalph, Xavier Llorà, David E. Goldberg, and Martin V. Butz. 2012. Resource management and scalability of the XCSF learning classifier system. *Theoretical Computer Science* 425 (2012), 126–141.

[61] Christopher Stone and Larry Bull. 2003. For Real! XCS with Continuous-Valued Inputs. *Evolutionary Computation* 11, 3 (2003), 299–336.

[62] Christopher Stone and Larry Bull. 2005. *An Analysis of Continuous-Valued Representations for Learning Classifier Systems*. Springer Berlin Heidelberg, Berlin, Heidelberg, 127–175.

[63] Kurian K. Tharakunnel, Martin V. Butz, and David E. Goldberg. 2003. Towards Building Block Propagation in XCS: A Negative Result and Its Implications. In *Genetic and Evolutionary Computation — GECCO 2003*, Erick Cantú-Paz, James A. Foster, Kalyanmoy Deb, Lawrence David Davis, Rajkumar Roy, Una-May O'Reilly, Hans-Georg Beyer, Russell Standish, Graham Kendall, Stewart Wilson, Mark Harman, Joachim Wegener, Dipankar Dasgupta, Mitch A. Potter, Alan C. Schultz, Kathryn A. Dowsland, Natasha Jonoska, and Julian Miller (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 1906–1917.

[64] Ryan J. Urbanowicz and Will N. Browne. 2017. *Introduction to Learning Classifier Systems* (1 ed.). Springer Berlin Heidelberg, Berlin, Heidelberg.

[65] Ryan J. Urbanowicz and Jason H. Moore. 2009. Learning Classifier Systems: A Complete Introduction, Review, and Roadmap. *Journal of Artificial Evolution and Applications* 2009 (2009), 1.

[66] Ryan J. Urbanowicz and Jason H. Moore. 2010. The Application of Michigan-style Learning Classifiersystems to Address Genetic Heterogeneity and Epistasisin Association Studies. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation (GECCO '10)*. ACM, New York, NY, USA, 195–202.

[67] Atsushi Wada, Keiki Takadama, Katsunori Shimohara, and Osamu Katai. 2005. *Learning Classifier System with Convergence and Generalization*. Springer Berlin Heidelberg, Berlin, Heidelberg, 285–304.

[68] Stewart W. Wilson. 1995. Classifier Fitness Based on Accuracy. *Evolutionary Computation* 3, 2 (1995), 149–175.

[69] Stewart W. Wilson. 1996. Generalization in XCS. In *Machine Learning, Proceedings of the 13th International Conference (ICML '96)*, Lorenza Saitta (Ed.). Morgan Kaufmann, 3–11.

[70] Stewart W. Wilson. 2000. Get Real! XCS with Continuous-Valued Inputs. In *Learning Classifier Systems*, Pier Luca Lanzi, Wolfgang Stolzmann, and Stewart W. Wilson (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 209–219.

[71] Stewart W. Wilson. 2000. State of XCS Classifier System Research. In *Learning Classifier Systems*, Pier Luca Lanzi, Wolfgang Stolzmann, and Stewart W. Wilson (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 63–81.

[72] Stewart W. Wilson and David E. Goldberg. 1989. A Critical Review of Classifier Systems. In *Proceedings of the 3rd International Conference on Genetic Algorithms*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 244–255.