Modeling Selection Pressure in XCS for Proportionate and Tournament Selection

Albert Orriols-Puig[1,2], Kumara Sastry[2], Pier Luca Lanzi[2,3], David E. Goldberg[2], Ester Bernado Mansilla[2]
[1]Group of Research in Intelligent Systems, Computer Engineering Department
Enginyeria i Arquitectura La Salle — Ramon Llull University, 08022, Barcelona, Catalonia, Spain.
[2]Illinois Genetic Algorithm Laboratory (IlliGAL)
University of Illinois at Urbana Champaign, Urbana, IL 61801, USA
[3]Dipartimento di Elettronica e Informazione, Politecnico di Milano, I-20133 Milano, Italy
aorriols@salle.url.edu,ksastry@uiuc.edu,lanzi@elet.polimi.it,deg@uiuc.edu,esterb@salle.url.edu

ABSTRACT

In this paper, we derive models of the selection pressure in XCS for proportionate (roulette wheel) selection and tournament selection. We show that these models can explain the empirical results that have been previously presented in the literature. We validate the models on simple problems showing that, (i) when the model assumptions hold, the theory perfectly matches the empirical evidence; (ii) when the model assumptions do not hold, the theory can still provide qualitative explanations of the experimental results.

Categories and Subject Descriptors

F.1.1 [**Models of Computation**]: Genetics Based Machine Learning, Learning Classifier Systems

General Terms

Algorithms, Theory.

Keywords

LCS, XCS, Proportionate Selection, Tournament Selection.

1. INTRODUCTION

Learning Classifier Systems [12, 9] combine genetic algorithms, reinforcement learning, and a general-purpose rule-based representation to solve problems online. Because they combine different paradigms, they are also daunting to study. The analysis of how a learning classifier system works requires knowledge about how each component works and, most important, about how such components interact. As a consequence, few theoretical models have been developed (e.g., [13, 1]) and learning classifier systems are more often studied empirically.

Recently, *facetwise modeling* [10] has been successfully applied to develop bits and pieces of a theory of XCS [16]. Butz et al. [6] modeled different generalization pressures in XCS so as to provide the theoretical foundation of Wilson's *generalization hypothesis* [17]; they also derived population bounds that ensure effective genetic search in XCS. Later, Butz et al. [2] also derived a bound for the learning time of XCS until maximally accurate classifiers are found. More recently, Butz et al. [4] presented a Markov chain analysis of niche support in XCS which resulted in another population size bound to ensure effective problem substainance.

In this paper, we present a model of selection pressure under proportionate selection [16] and tournament selection [7]. These selection schemes have been empirically compared by Butz et al. [3, 7] and later by Kharbat et al. [14] leading to different claims. In genetic algorithms, these selection schemes have been exhaustively studied through the analysis of takeover time (see [11, 10] and references therein). In this paper, we follow the same approach as [11, 10] and develop theoretical models of selection pressure in XCS for proportionate and tournament selection. Specifically, we perform a takeover time analysis to estimate the time from an initial proportion of best individuals until the population is substantially converged to the best. We start from the typical assumption made in takeover time analysis [11]: XCS has converged to an optimal solution, which in XCS typically consists of a set of non-overlapping niches. We write differential equations that describe the change in proportion of the best classifier in one niche for roulette wheel and tournament selection. Initially, we focus on classifier accuracy, later we extend the model taking into account also classifier generality. We solve the equations and derive a closed form solution of takeover time in XCS for the two selection schemes (Section 3). In Section 4, we use these equations to determine the conditions under which proportionate and tournament selection (i) produce the same initial growth of the best classifiers in the niche, or (ii) result in the same takeover time. Then, in Section 6, we validate the models using two artificial test problems showing that when the assumptions of non-overlapping niches hold, the models perfectly match the empirical evidence while they accurately approximate the empirical results when such an assumption is violated. Finally, in Section 7, we extend the models to include classifier generality and show that, again, the models fit empirical evidence when the model assumptions hold.

2. THE XCS CLASSIFIER SYSTEM

We briefly describe XCS giving all the details that are relevant to this work. We refer the reader to [16, 8] for more detailed descriptions.

Knowledge Representation. In XCS, classifiers consist of a condition, an action, and four main parameters: (i) the prediction p_k, which estimates the payoff that the system expects when the classifier is used; (ii) the prediction error ϵ_k, which estimates the error affecting the prediction p_k; (iii) the fitness F_k, which estimates the accuracy of the payoff prediction given by p_k; and (iv) the numerosity n_k, which indicates how many copies of classifiers with the same condition and the same action are present in the population.

Performance Component. At time t, XCS builds a *match set* [M] containing the classifiers in the population [P] whose condition matches the current sensory input s_t; if [M] contains less than θ_{nma} actions, *covering* takes place and creates a new classifier that matches s_t and has a random action. For each possible action a in [M], XCS computes the *system prediction* $P(s_t, a)$ which estimates the payoff that XCS expects if action a is performed in s_t. The system prediction $P(s_t, a)$ is computed as the fitness weighted average of the predictions of classifiers in [M] which advocate action a. Then, XCS selects an action to perform; the classifiers in [M] which advocate the selected action form the current *action set* [A]. The selected action a_t is performed, and a scalar reward R is returned to XCS.[1]

Parameter Updates. The incoming reward R is used to update the parameters of the classifiers in [A]. First, the classifier prediction p_k is updated as, $p_k \leftarrow p_k + \beta(R - p_k)$. Next, the error ϵ_k is updated as, $\epsilon_k \leftarrow \epsilon_k + \beta(|R - p| - \epsilon_k)$. To update the classifier fitness F_k, the classifier *accuracy* κ is first computed as,

$$\kappa = \begin{cases} 1 & \text{if } \epsilon < \epsilon_0 \\ \alpha(\epsilon/\epsilon_0)^{-\nu} & \text{otherwise} \end{cases} \quad (1)$$

the *accuracy* κ is used to compute the *relative accuracy* κ' as, $\kappa' = \kappa / \sum_{[A]} \kappa_i$. Finally, the classifier fitness F_k is updated towards the classifier's relative accuracy κ' as, $F_k \leftarrow F_k + \beta(\kappa' - F_k)$.

Genetic Algorithm. On a regular basis (dependent on the parameter θ_{ga}), the genetic algorithm is applied to classifiers in [A]. It selects two classifiers, copies them, and with probability χ performs crossover on the copies; then, with probability μ it mutates each allele. The resulting offspring classifiers are inserted into the population and two classifiers are deleted to keep the population size constant.

Two selection mechanisms have been introduced so far for XCS: proportionate, roulette wheel, selection [16] and tournament selection [7]. Roulette wheel selects a classifier with a probability proportional to its fitness. Tournament randomly chooses the τ percent of the classifiers in the action set and among these it selects the classifier with higher "microclassifier fitness" f_k; that is, as the numerosity is included in the fitness calculation, tournament selection obtains the numerosity of a single classifier as: $f_k = F_k/n_k$ [7, 5].

[1] In this paper we focus on XCS viewed as a pure classifier, i.e., applied to single-step problems. A complete description is given elsewhere [16, 8].

3. MODELING TAKEOVER TIME

The analysis of takeover time in XCS poses two main challenges. First, while in genetic algorithms the fitness of an individual is usually constant, in XCS classifier fitness changes over time based on the other classifiers that appear in the same evolutionary niche. Second, while in genetic algorithms the selection and the replacement of individuals are usually performed over the whole population, in XCS selection is niche based, while deletion is population based.

To model takeover time [11], we assume that XCS has evolved a set of non-overlapping *niches*, where a niche represents a region in the solution space that consists of classifiers that represent the same *schema* [12] and predict the same output. Accordingly, we can focus on one niche without taking into account possible interactions among overlapping niches. We consider a simplified scenario in which a niche contains two classifiers, cl_1 and cl_2; classifier cl_k has fitness F_k, prediction error ϵ_k, numerosity n_k, and *microclassifier fitness* $f_k = F_k/n_k$. In this initial phase, we focus on classifier accuracy and hypothesize that cl_1 be the "best" classifier in the niche, because is the most accurate, i.e., $\kappa_1 \geq \kappa_2$, while we assume that cl_1 and cl_2 are equally general and thus they have the same reproductive opportunities (this assumption will be relaxed later in Section 7). Finally, we assume that deletion selects classifiers randomly from the same niche. Although this assumption may appear rather strong, if the niches are activated uniformly this assumption will have little effect as the empirical validation of the model will show (Section 6).

3.1 Roulette Wheel Selection

Under Roulette Wheel Selection (RWS), the selection probability of a classifier depends on the ratio of its fitness F_i over the fitness of all classifiers in the action set. Without loss of generality, we assume that classifier fitness is a simple average of classifier's relative accuracies and compute the fitness of classifiers cl_1 and cl_2 as,

$$F_1 = \frac{\kappa_1 n_1}{\kappa_1 n_1 + \kappa_2 n_2} = \frac{1}{1 + \rho n_r}$$

$$F_2 = \frac{\kappa_2 n_2}{\kappa_1 n_1 + \kappa_2 n_2} = \frac{\rho n_r}{1 + \rho n_r}$$

where $n_r = n_2/n_1$ and ρ is the ratio between the accuracy of cl_2 and the accuracy of cl_1 ($\rho = \kappa_2/\kappa_1$). The probability P_s of selecting the best classifier cl_1 in the niche is computed as,

$$P_s = \frac{F_1}{F_1 + F_2} = \frac{1}{1 + \rho n_r}$$

Once selected, a new classifier is created and inserted in the population while one classifier is randomly deleted from the niche with probability $P_{del}(cl_j) = n_j/n$ with $n = n_1 + n_2$.

We can now model the evolution of the numerosity of the best classifier cl_1 at time t, $n_{1,t}$, which will (i) increase in the next generation if cl_1 is selected by the genetic algorithm and another classifier is selected for deletion; (ii) decrease if cl_1 is not selected by the genetic algorithm but cl_1 is selected for deletion; (iii) remain the same, in all the other cases.

More formally,

$$n_{1,t+1} = \begin{cases} n_{1,t}+1 & \text{with prob. } \frac{1}{1+\rho n_r}\left(1-\frac{n_1}{n}\right) \\ n_{1,t}-1 & \text{with prob. } \left(1-\frac{1}{1+\rho n_r}\right)\frac{n_1}{n} \\ n_{1,t} & \text{otherwise} \end{cases}$$

where n is the niche size. This model is relaxed by assuming that one classifier only can be deleted when sampling the niche it belongs. Grouping the equations above, we obtain,

$$n_{1,t+1} = n_{1,t} + \frac{1}{1+\rho n_r} - \frac{n_{1,t}}{n}$$

which can be rewritten in terms of the proportion P_t of classifiers cl_1 in the niche (i.e., $P_t = n_{1,t}/n$). Using the equality $n_r = (1-P_t)/P_t$ we derive,

$$P_{t+1} = P_t + \frac{1}{n}\cdot\frac{1}{1+\rho\frac{1-P_t}{P_t}} - \frac{1}{n}P_t$$

assuming $P_{t+1} - P_t \approx dp/dt$ we have,

$$\frac{dp}{dt} \approx P_{t+1} - P_t = \frac{1}{n}\left[\frac{P_t(1-\rho) - P_t^2(1-\rho)}{P_t(1-\rho)+\rho}\right] \qquad (2)$$

that is,

$$\frac{P_t(1-\rho)+\rho}{P_t(1-P_t)}dp \approx \frac{1-\rho}{n}dt \qquad (3)$$

which can be solved by integrating each side of the equation, between P_0, the initial proportion of cl_1, and the final proportion P of cl_1 up to which cl_1 has taken over,

$$\int_{P_0}^{P}\frac{P_t(1-\rho)+\rho}{P_t(1-P_t)}dp \approx \frac{1-\rho}{N}\int dt \approx \frac{t(1-\rho)}{N} \qquad (4)$$

the left integral can be solved independently as [15],

$$\int_{P_0}^{P}\frac{P_t(1-\rho)+\rho}{P_t(1-P_t)}dp \approx \frac{t(1-\rho)}{N} \qquad (5)$$

from which we derive the takeover time of cl_1 in roulette wheel selection,

$$t^*_{rws} \approx \frac{N}{1-\rho}\left[\rho\ln\left(\frac{P}{P_0}\right) + \ln\left(\frac{1-P_0}{1-P}\right)\right] \qquad (6)$$

The takeover time t^*_{rws} under the assumption that the two classifiers are equally general depends (i) on the ratio ρ between the accuracy of cl_2 and the accuracy of cl_1, as well as, (ii) on the initial proportion of cl_1 in the population. A higher ρ implies an increase in the takeover time. When $\rho = 1$ (i.e., cl_1 and cl_2 are equally accurate), t^*_{rws} tends to infinity, that is, both cl_1 and cl_2 will remain in the population. When ρ is smaller, the takeover time decreases. For a ρ close to zero, t^*_{rws} can be approximated by $t^*_{\rho\to 0} \approx n\ln\frac{1-P_0}{1-P}$. The takeover time *also* depends on the initial proportion P_0 of cl_1 in the population. A lower P_0 increases the term inside the brackets, and so it increases the takeover time. In contrast, an higher P_0 results in a lower takeover time.

3.2 Tournament Selection

To model takeover time for tournament selection in XCS we assume a fixed tournament size of s, instead of the typical variable tournament size [5]. The underline assumption of the takeover time analysis is that the system already converged to an optimal solution made of non overlapping niches. Thus, there is basically no difference between a fixed tournament size and a variable one. Tournament selection randomly chooses s classifiers in the action set and selects the one with higher fitness. As before, we assume that cl_1 is the best classifier in the niche, which in terms of tournament selection translates into requiring that $f_1 > f_2$, where f_i is the fitness of the microclassifiers associated to cl_i. In this case, the numerosity n_1 of cl_1 will (i) increase if cl_1 participates in the tournament and another classifier is selected to be deleted; (ii) decrease if cl_1 does not participate in the tournament but is selected by the deletion operator; (iii) remain the same, otherwise. More formally,

$$n_{1,t+1} = \begin{cases} n_{1,t}+1 & \text{with prob. } \left[1-\left(1-\frac{n_1}{n}\right)^s\right]\left(1-\frac{n_1}{n}\right) \\ n_{1,t}-1 & \text{with prob. } \left(1-\frac{n_1}{n}\right)^s\frac{n_1}{n} \\ n_{1,t} & \text{otherwise} \end{cases}$$

By grouping the above equations we derive the expected numerosity of cl_1,

$$n_{t+1} = n_t + \left[1-\left(1-\frac{n_1}{n}\right)^s\right]\left(1-\frac{n_1}{n}\right) - \left(1-\frac{n_1}{n}\right)^s\frac{n_1}{n}$$

which we rewrite in terms of the proportion P of cl_1 in the niche (i.e., $P_t = n_{1,t}/n$) as,

$$P_{t+1} = P_t + \frac{1}{n}(1-P_t)\left[1-(1-P_t)^{s-1}\right]$$

Assuming $\frac{dp}{dt} \approx P_{t+1} - P_t$, we derive,

$$\frac{dp}{dt} \approx P_{t+1} - P_t = \frac{1}{n}\left[(1-P_t)[1-(1-P_t)^{s-1}]\right], \qquad (7)$$

that is,

$$\frac{dt}{n} \approx \frac{1}{1-P_t}dp + \frac{(1-P_t)^{s-2}}{1-(1-P_t)^{s-1}}dp$$

Integrating each side of the equation we obtain,

$$\int\frac{dt}{n} \approx \int_{P_0}^{P}\frac{1}{1-P_t}dp + \int_{P_0}^{P}\frac{(1-P_t)^{s-2}}{1-(1-P_t)^{s-1}}dp$$

i.e.,

$$\frac{t}{n} \approx \ln\left(\frac{1-P_0}{1-P}\right) + \frac{1}{s-1}\ln\left[\frac{1-(1-P)^{s-1}}{1-(1-P_0)^{s-1}}\right]$$

so that the takeover time of cl_1 for tournament selection is,

$$t^*_{TS} \approx n\left[\ln\left(\frac{1-P_0}{1-P}\right) + \frac{1}{s-1}\ln\left[\frac{1-(1-P)^{s-1}}{1-(1-P_0)^{s-1}}\right]\right] \qquad (8)$$

Given our assumptions, takeover time for tournament selection depends on the initial proportion of the best classifier P_0 and the tournament size s. Both logarithms on the right hand side take positive values if $P > P_0$, indicating that the best classifier will always take over the population regardless of its initial proportion in the population. When $P < P_0$, both logarithms result in negative values, and so does the takeover time.

4. COMPARISON

We now compare the takeover time for roulette wheel selection and tournament selection and we study the salient differences between the models. For this purpose, we compute the values of s and ρ for which, (i) the two selection schemes have the same initial increase of the proportion of the best classifier, and for which (ii) the two selection schemes have the same takeover time. This analysis results in two expressions that permit to compare the behavior of roulette wheel selection and tournament selection in different scenarios.

First, we analyze the relation between the ratio of classifier accuracies ρ and the selection pressure s in tournament selection to obtain the same initial increase in the proportion of the best classifier with both selection schemes. Taking the equations 2 and 7, and replacing P_t by P_0 we get that the initial increase of the best classifier in each selection scheme is:

$$\Delta_{RWS} \approx \frac{1-\rho}{n}\frac{P_0(1-P_0)}{(1-\rho)P_0+\rho}$$

$$\Delta_{TS} \approx \frac{1}{n}(1-P_0)[1-(1-P_0)^{s-1}]$$

By requiring $\Delta_{RWS} = \Delta_{TS}$ and simplifying the equation further [15] we obtain,

$$\frac{P_t(1-\rho)}{(1-\rho)P_t+\rho} \approx 1 - \left[(1-P_t)^{s-1}\right],$$

that is,

$$s \approx 1 + \frac{\ln\rho - \ln\left[(1-\rho)P_0+\rho\right]}{\ln(1-P_0)} \quad (9)$$

This equation indicates that, in tournament selection, the tournament size s which regulates the selection pressure has to increase as the ratio of accuracies ρ decreases to have the same initial increase of the best classifier in the population as roulette wheel selection. For example, for $\rho = 0.01$ and $P_0 = 0.01$, tournament selection requires $s \approx 70$ to produce the same initial increase of the best classifier as roulette wheel selection. On the other hand, low values of s result in the same effect as roulette wheel selection with high values of ρ. Equation 9 indicates that, even with a small tournament size, tournament selection produces a stronger pressure towards the best classifier in scenarios in which slightly inaccurate but initially highly numerous classifiers are competing against highly accurate classifiers. On the other hand, when the competition involves highly inaccurate classifiers, a larger tournament size s is required to obtain the same selection pressure provided by roulette wheel selection.

We now analyze the conditions under which both selection schemes result in the same takeover time. For this purpose, we equate t^*_{RWS} in Equation 6 with t^*_{TS} in Equation 8,

$$\frac{\rho}{1-\rho}\ln\left(\frac{P^*(1-P_0)}{P_0(1-P^*)}\right) \approx \frac{1}{s-1}\ln\left(\frac{1-(1-P^*)^{s-1}}{1-(1-P_0)^{s-1}}\right).$$

By approximating $(1-x)^{s-1}$ by its first order Taylor series at the point 0, that is, $(1-x)^{s-1} \approx 1+(s-1)x$, and by futher simplifications, we derive,

$$s \approx 1 + \frac{1-\rho}{\rho}\frac{\ln(P^*)-\ln(P_0)}{\ln[P^*(1-P_0)]-\ln[P_0(1-P^*)]} \quad (10)$$

Given an initial proportion of the best classifier, the equation is guided by the term $\frac{1-\rho}{\rho}$. As the accuracy-ratio ρ decreases, s needs to increase polynomially. On the other hand, higher values of ρ require a low tournament size s. Thus, Equation 10 reaches the same conclusions of Equation 9: tournament selection is better than roulette wheel in scenarios where highly accurate classifiers compete with slightly inaccurate ones.

5. TEST PROBLEMS

To validate the models of takeover time for roulette wheel and proportionate selection, we designed two test problems. The **single-niche problem** fully agrees with the model assumption. It consists of a niche with a highly accurate classifier cl_1 and a less accurate classifier cl_2. The covering is off and the population is initialized with $N \cdot P_0$ copies of cl_1 and $N \cdot (1-P_0)$ copies of cl_2, where N is the population size. The prediction error ϵ_1 of the best classifier cl_1 is always set to zero while the prediction error ϵ_2 of cl_2 is set as,

$$\epsilon_2 = \epsilon_0\left(\frac{\rho}{\alpha}\right)^{\nu} \quad (11)$$

where ρ is the ratio between the accuracy of cl_2 and the accuracy of cl_1 i.e., $\rho = \kappa_2/\kappa_1$ (Section 3). In the experiments, we set $\alpha = 0.1$ and $\nu = 5$. Note that varying ρ we are changing the fitness scaling between cl_1 and cl_2. This could be equivalently done by maintaining ρ and varying ν, as in [14].

The **multiple-niche** problem consists of a set of m niches each one containing one maximally accurate classifier and one classifier that belongs to all niches. The population contains $P_0 \cdot N$ copies of maximally accurate classifiers (equally distributed in the m niches) and $(1 - m \cdot P_0) \cdot N$ copies of the classifier that appears in all the niches. The parameters of classifiers are updated as in the case of the single niche in the previous section, permitting to vary the ratio of accuracies ρ, and so, validating the model under different circumstances. This test problem violates two model assumptions. First, the size of the different niches differ from the population size since the problem consists of more than one niche. While in the model we assumed that the deletion would always select a classifier in the same niche, in this case deletion can select any classifier from the population. Second, the niches are overlapping since there is a classifier that belongs to all the niches. This also means that the sum of all niche sizes will be greater than the population size, i.e., $\sum_1^m(n_{i,1}+n_{i,2}) > N$, where $n_{i,1}$ is the numerosity of the maximally accurate classifier in niche i and $n_{i,1}$ is the numerosity of the less accurate classifier in the niche i.

6. EXPERIMENTAL VALIDATION

We validated the takeover time models using the *single-niche* problem and the *multiple-niche* problem. Our results show a full agreement between the model and the experiments in the *single-niche problem*, when all the model assumptions hold. They also show that the theory correctly predicts the practical results in problems where the initial assumptions do not hold, if either the accuracy-ratio is low enough in roulette wheel selection or the tournament size is high enough in tournament selection.

Single-Niche. At first, we applied XCS on the *single-niche* problem and analyzed the proportion of the best classifier cl_1 under roulette wheel selection and tournament selection. Figure 1 compares the proportion of the best classifier in the niche as predicted by our model (reported as lines) and the empirical data (reported as dots) for roulette wheel selection and tournament selection with $s = \{2, 3, 9\}$ when $\rho = 0.01$. The empirical data are averages over 50 runs. Figure 1 shows a perfect match between the theory and the empirical results. It also shows that, as predicted by the models and discussed in Section 4, roulette wheel produces a faster increase in the proportion of the best classifier for $\rho = 0.01$.

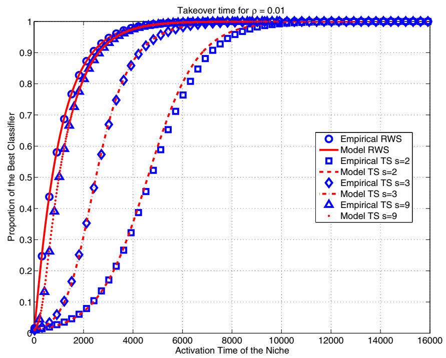


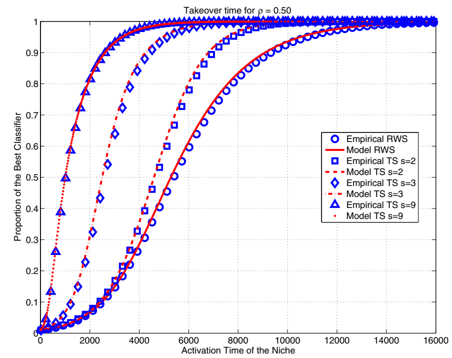
Figure 1: Empirical and the theoretical takeover time when $\rho = 0.01$ for roulette wheel selection and tournament selection with s=2, s=3 and s=9.

To obtain a similar increase with tournament selection we need an higher tournament size: in fact, the second best increase in the proportion of the best classifier is provided by tournament selection with $s = 9$. Finally, as indicated by the theory, the time to achieve a certain proportion of the best classifier in tournament selection increases as the selection pressure decreases.

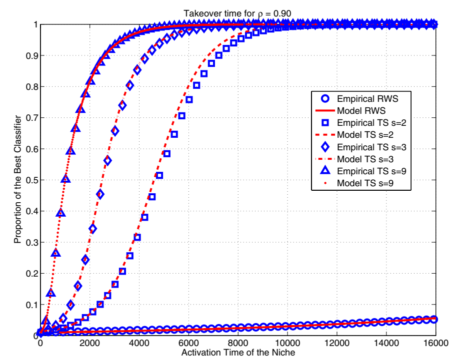
We performed another set of experiments and compared the proportion of the best classifier in the niche for an accuracy ratio ρ of 0.5 and 0.9. Figure 2 compares the proportion of cl_1 as predicted by our model (reported as lines) and the empirical data (reported as dots) for (a) $\rho = 0.5$ and (b) $\rho = 0.9$; the empirical data are averages over 50 runs. The results show a good match between the theory and the empirical results. As predicted by the model, tournament selection is not influenced by the increase of ρ: the increase in the proportion of cl_1 for $\rho = 0.5$ (Figure 2a) is basically the same as the increase obtained when $\rho = 0.9$ (Figure 2b). Thus, coherently to what was empirically shown in [7], tournament selection demonstrates its robustness in maintaining and increasing the proportion of the best classifier even when there is a small difference between the fitness of the most accurate classifier (cl_1) and the fitness of the less accurate one (cl_2). In contrast, roulette wheel selection is highly influenced by the accuracy ratio ρ: as the ratio approaches 1, i.e., the accuracy of the two classifiers become very similar, the increase in the proportion of the best classifier becomes smaller and smaller. In fact, when $\rho = 0.90$, after 16000 activations of the niche the best classifier cl_1 has taken over only the 5% of the niche.

Multiple-niche. In the next set of experiments, we validated our model of takeover time on the *multiple-niche* problem, where the assumptions about non-overlapping niches and about deletion being performed in the niche are violated. For this purpose, we ran XCS on the multiple-niche problem with two niches ($m = 2$); each niche contains one maximum accurate classifier and there is one, less accurate, overlapping classifier that participates in both niches.

Figure 3a compares the proportion of the best classifier in the niche for roulette wheel selection for an accuracy ratio ρ of 0.01 and 0.20. The plots indicate that, for small values of the accuracy ratio ρ, our model (reported as lines) slightly underestimates the empirical takeover time



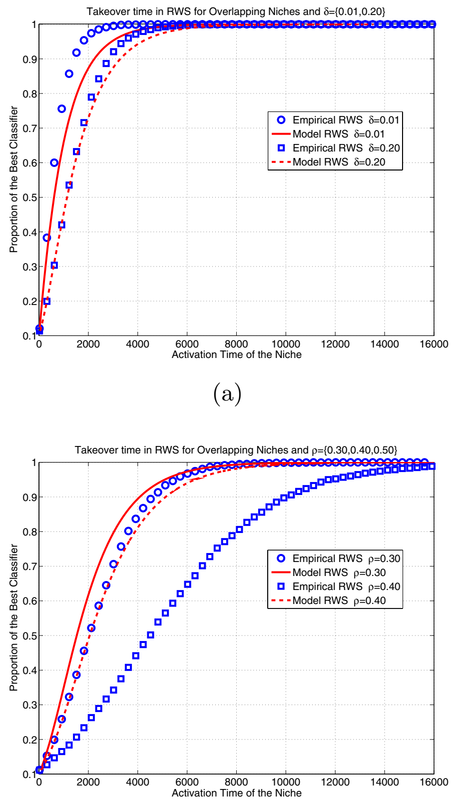
(a)



(b)

Figure 2: Empirical and the theoretical takeover time for roulette wheel selection and tournament selection with s=2, s=3 and s=9 when (a) $\rho = 0.50$ and (b) $\rho = 0.90$.

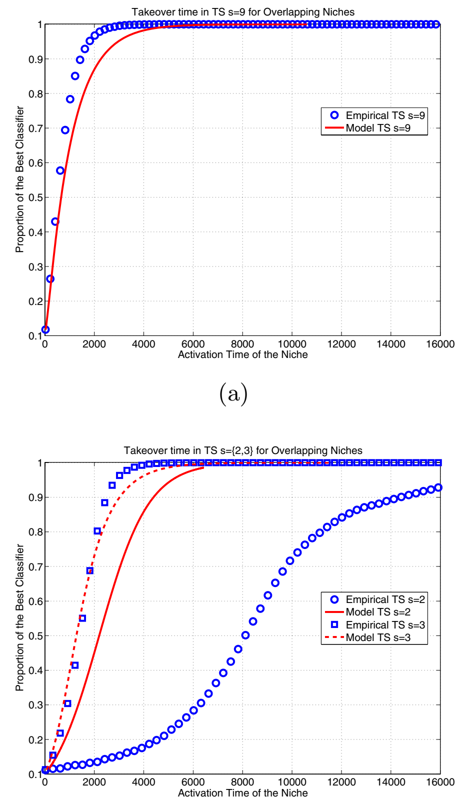
(reported as dots). As the accuracy ratio ρ increases, the model tends to overestimate the empirical data (Figure 3b). This behavior can be easily explained. The lower the accuracy ratio (Figure 3a), the higher the pressure toward the highly accurate classifiers, and consequently, the faster the takeover time. When ρ is 0.20 and 0.30, the difference between the model and the empirical results is visible only at the beginning, while it basically disappears as the number of activations increases. For higher values of ρ, the overgeneral, less accurate, cl_2 has more reproductive opportunities in both niches where it participates. These results indicate that (i) the model for roulette wheel selection is accurate in general scenarios if the ratio of accuracies is small (i.e., when there is a large proportion of accurate classifiers in the niche) and (ii) that in situations where there is a small proportion of the best classifier in one niche competing with other slightly inaccurate and overgeneral classifiers (above a certain threshold of ρ), the overgeneral classifier may take over the population removing all the copies of the best classifier. It is interesting to note that, as the number of niches increases from $m = 2$ to $m = 16$, the agreement between the theory and the experiments gently degrades (see [15] for more results). Figure 4a compares the proportion of the best classifier as predicted by our model and as empirically determined in tournament selection with $s = 9$. As in roulette wheel selection for a small ρ, the results for tournament



(a)

(b)

Figure 3: Takeover time for roulette wheel selection when (a) $\rho = \{0.01, 0.20\}$ and (b) $\rho \in \{0.30, 0.40\}$.



(a)

(b)

Figure 4: Takeover time in tournament selection for (a) $s = 9$, (b) $s \in \{2, 3\}$.

selection show that the empirical takeover time is slightly faster than the one predicted by the theory. Again, this behavior is due to the presence of the overgeneral classifier in both niches, causing a higher pressure toward its deletion. Increasing the tournament size s produces little variations in either the empirical results or the theoretical values, and so the conclusions extracted for $s = 9$ can be extended for higher s. On the other hand, decreasing s causes that the empirical values go closer to the theoretical values, since the pressure toward the deletion of the overgeneral classifier decreases. Figure 4b reports the proportion of one of the maximum accurate classifiers for $s \in \{2, 3\}$. The results show that the theory predicts accurately the empirical values for $s = 3$, but for $s = 2$ the difference between the model and the data is large. In this case, a small tournament size combined with the presence of the overgeneral classifier in both niches produces a strong selection pressure toward the overgeneral classifier that delays the takeover time.

The results in the multi-niche problem support what was empirically shown in [7]: tournament selection is more robust than roulette wheel selection. Under perfect conditions both schemes perform similarly, which is coherent to what was shown in [14]. However, the takeover time of the best classifier is delayed in roulette wheel selection for a higher accuracy ratio ρ. The empirical results indicate that, with roulette wheel selection, the best classifier was eventually removed from the population when $\rho \geq 0.5$.

On the other hand, tournament selection demonstrated theoretically and practically to be more robust, since it does not depend on the individual fitness of each classifier. In all experiments with tournament selection, the best classifier could take over the niche, and only for the extreme case (i.e., for $s = 2$) the experiments considerably disagreed with the theory.

7. MODELING GENERALITY

Finally, we add classifier generality to the picture. As before, we model the proportion of the best classifier cl_1 in one niche, however, in this case cl_1 is not only the maximally accurate classifier in the niche, but also the maximally general with respect to niche. We focus on one niche and assume that cl_1, being the maximally general and maximally accurate classifier, appears in the niche with probability 1 so that cl_2 appears in the niche with a relative probability ρ_m. We model the numerosity of the classifier cl_1 at time t, $n_{1,t}$ as follows. The numerosity $n_{1,t}$ of cl_1 in the next generation will (i) increase when both cl_1 and cl_2 appear in the niche and cl_1 is selected by the genetic algorithm while cl_2 is selected for deletion; (ii) increase when only cl_1 appears in the niche and another classifier is deleted; (iii) decrease only if both cl_1 and cl_2 are in the niche, cl_1 is not selected by the genetic algorithm but it is selected for deletion. Otherwise, the numerosity of cl_1 will remain the same. More formally,

$$n_{1,t+1} = \begin{cases} n_{1,t}+1 & \text{with prob.} \quad \rho_m \left(\frac{1}{1+\rho n_r}\right)\left(1-\frac{n_1}{n}\right) + \\ & \quad (1-\rho_m)\left(1-\frac{n_1}{n}\right) \\ n_{1,t}-1 & \text{with prob.} \quad \rho_m \left(1-\frac{1}{1+\rho n_r}\right)\frac{n_1}{n} \\ n_{1,t} & \text{otherwise} \end{cases}$$

As done before, we can group these equations to obtain,

$$n_{1,t+1} = n_{1,t} + \rho_m \left(\frac{1}{1+\rho n_r} - \frac{n_{1,t}}{n}\right) + (1-\rho_m)\left(1-\frac{n_{1,t}}{n}\right)$$

we rewrite $n_{1,t}$ in terms of the proportion P_t of cl_1 in the niche ($P_t = n_{1,t}/n$ with $n_r = (1-P_t)/P_t$) and obtain,

$$P_{t+1} = P_t + \frac{1}{n} \cdot \frac{1-P_t}{\rho + (1-\rho)P_t}\left[(1-\rho)P_t + (1-\rho_m)\rho\right]$$

Assuming $P_{t+1} - P_t \approx dp/dt$, we obtain [15],

$$\begin{aligned} t^*_{rws} \quad \approx & \quad \frac{n}{1-\rho\rho_m}\left[\ln\left(\frac{1-P_0}{1-P}\right) + \right. \\ + & \quad \left.\rho\rho_m \ln\left(\frac{(1-\rho)P + \rho(1-\rho_m)}{(1-\rho)P_0 + \rho(1-\rho_m)}\right)\right] \end{aligned} \qquad (12)$$

which depends on the ratio of accuracies ρ, the initial proportion of the best classifier P_0 and the generality of the inaccurate classifier ρ_m. In the previous model, cl_1 would not take over the niche when it was as accurate as cl_2 (Section 3). In this case, cl_1 will take over the population if it is either *more accurate* or *more general* than cl_2. Otherwise, if cl_1 and cl_2 are equally accurate and general, both will persist in the population ($t^*_{RWS} \rightarrow \infty$), coherently with the previous model. Low values of ρ_m suppose quicker takeover times than high values of ρ_m. For $\rho_m = 1$, i.e., the classifiers cl_1 and cl_2 are equally general, Equation 12 is equivalent to Equation 6.

Similarly, we extend the model of takeover time for tournament selection taking classifier generality into account. In this case, the numerosity $n_{1,t}$ of cl_1 at time t is,

$$n_{1,t+1} = \begin{cases} n_{1,t}+1 & \text{with prob.} \quad \rho_m \left[1-\left(1-\frac{n_1}{n}\right)^s\right] \cdot \left(1-\frac{n_1}{n}\right) \\ & \quad +(1-\rho_m)\left(1-\frac{n_1}{n}\right) \\ n_{1,t}-1 & \text{with prob.} \quad \rho_m \left(1-\frac{n_1}{n}\right)^s \frac{n_1}{n} \\ n_{1,t} & \text{otherwise.} \end{cases}$$

As we did before, we can group these equations and we can rewrite $n_{1,t+1}$ in terms of P_t as,

$$P_{t+1} = P_t + \frac{1}{n}(1-P_t)\left[1-\rho_m(1-P_t)^{s-1}\right]$$

which, by assuming $\frac{dp}{dt} \approx P_{t+1} - P_t$, and through integration (see [15] for details), results in the following equation for takeover time with tournament selection,

$$t^*_{TS} \approx n\left[\ln\left(\frac{1-P_0}{1-P}\right) + \frac{1}{s-1}\ln\left[\frac{1-\rho_m(1-P)^{s-1}}{1-\rho_m(1-P_0)^{s-1}}\right]\right] \qquad (13)$$

Takeover time for tournament selection now depends on the tournament size s, the initial proportion of the best classifier P_0, and *also* on the generality of the inaccurate classifier ρ_m. For low ρ_m or high s, the value of the second logarithm in the right term of the equation diminishes so that the takeover time mainly depends logarithmically on P_0. High values of ρ_m imply slower takeover times; for $\rho_m = 1$, this model equates Equation 8.

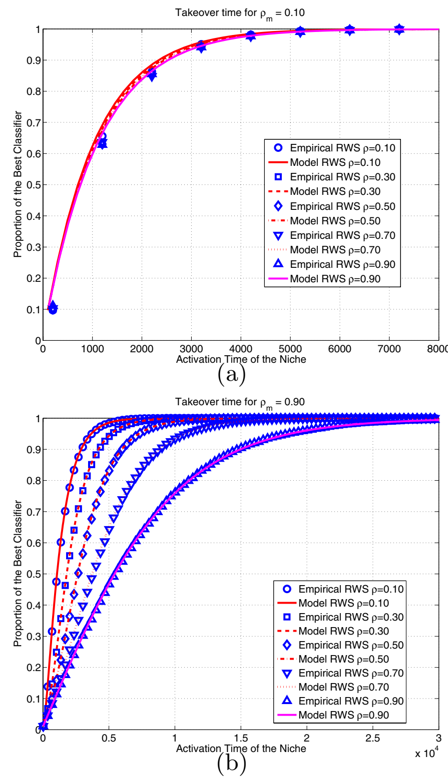
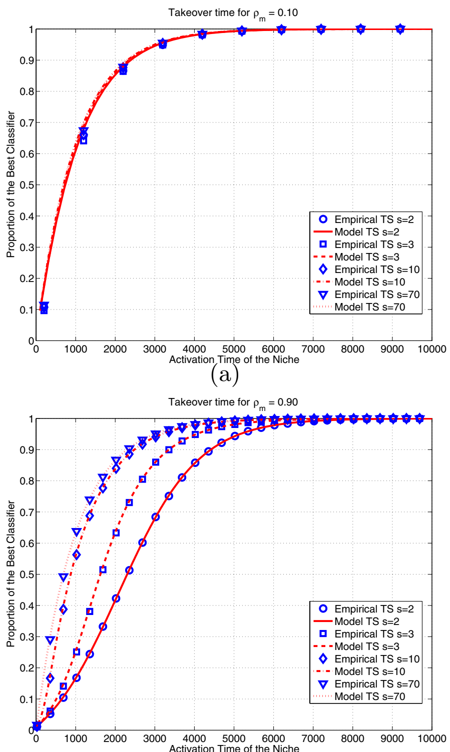


Figure 5: Takeover time for roulette wheel selection when (a) ρ_m is 0.1 and (b) ρ_m is 0.9.

We validated the new models of takeover time for roulette wheel selection and tournament selection on the single-niche problem for a matching ratio ρ_m of 0.1 and 0.9. The experimental design is essentially the same one used in the previous experiments (Section 6), but in this case the less accurate and less general classifier cl_2 appears in the niche with probability ρ_m. Figure 5 compares the proportion of the best classifier cl_1 in the niche as predicted by the theory and both the plots for roulette wheel selection (Figure 5) and tournament selection (Figure 6) show a perfect match between the theory (reported as lines) and the empirical data (reported as dots).

8. CONCLUSIONS

We have derived theoretical models for the selection pressure in XCS under roulette wheel and tournament selection. We have shown that our models are accurate in very simplified scenarios and they can qualitatively explain the behavior of the two selection mechanisms in more complex scenarios. Overall, our models support what empirically shown in [7]: tournament selection is more robust than roulette wheel selection. Under perfect conditions both schemes perform similarly, which is coherent to what shown in [14]. However, the selection pressure is weaker in roulette wheel selection when the classifiers in the niche have similar accuracies. On the other hand, tournament selection turns out to be more robust both theoretically and practically, since it does not depend on the individual fitness of each classifier.



(a)

(b)

Figure 6: Takeover time for tournament selection when (a) ρ_m is 0.1 and (b) ρ_m is 0.9.

Acknowledgments

The authors thank the support of *Enginyeria i Arquitectura La Salle*, Ramon Llull University, as well as the support of *Ministerio de Ciencia y Tecnología* under project TIN2005-08386-C05-04, and *Generalitat de Catalunya* under Grants 2005FI-00252 and 2005SGR-00302.

This work was also sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant FA9550-06-1-0096, the National Science Foundation under ITR grant DMR-03-25939 at Materials Computation Center, UIUC. The U.S. Government is authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright notation thereon.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research, the National Science Foundation, or the U.S. Government.

9. REFERENCES

[1] L. Bull. Simple markov models of the genetic algorithm in classifier systems: Accuracy-based fitness. In *Proceedings of the International Workshop on Learning Classifier Systems (IWLCS-2000), in the Joint Workshops of SAB 2000 and PPSN 2000*, 2000. Extended abstract.

[2] M. Butz, D. E. Goldberg, and P. L. Lanzi. Bounding Learning Time in XCS. In *Genetic and Evolutionary Computation – GECCO-2004*, LNCS, Seattle, WA, USA, 26-30 June 2004. Springer-Verlag.

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

[4] M. Butz, D. G. Goldberg, P. L. Lanzi, and K. Sastry. Bounding the Population Size to Ensure Niche Support in XCS. Technical Report 2004033, Illinois Genetic Algorithms Laboratory – University of Illinois at Urbana-Champaign, Nov. 2004.

[5] M. V. Butz. Documentation of XCS+TS C-Code 1.2. Technical Report 2003023, Illinois Genetic Algorithm Laboratory (IlliGAL), University of Illinois at Urbana-Champaign, 2003.

[6] M. V. Butz, T. Kovacs, P. L. Lanzi, and S. W. Wilson. Toward a Theory of Generalization and Learning in XCS. *IEEE Transaction on Evolutionary Computation*, 8(1):28–46, Feb. 2004.

[7] M. V. Butz, K. Sastry, and D. E. Goldberg. Strong, Stable, and Reliable Fitness Pressure in XCS due to Tournament Selection. *Genetic Programming and Evolvable Machines*, 6:53–77, 2005.

[8] M. V. Butz and S. W. Wilson. An Algorithmic Description of XCS. *Journal of Soft Computing*, 6(3–4):144–153, 2002.

[9] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, Mass., 1989.

[10] D. E. Goldberg. *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*. Kluwer Academic Publishers, 2002.

[11] D. E. Goldberg and K. Deb. A Comparative Analysis of Selection Schemes Used in Genetic Algorithms. In G. J. E. Rawlins, editor, *FOGA*, pages 69–93. Morgan Kaufmann, 1990.

[12] J. H. Holland. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, 1975.

[13] J. Horn, D. E. Goldberg, and K. Deb. Implicit Niching in a Learning Classifier System: Nature's Way. *Evolutionary Computation*, 2(1):37–66, 1994.

[14] F. Kharbat, L. Bull, and M. Odeh. Revisiting Genetic Selection in the XCS Learning Classifier System. In *Congress on Evolutionary Computation*, volume 3, pages 2061–2068, Edinburgh, UK, 2-5 September 2005. IEEE.

[15] A. Orriols-Puig, K. Sastry, P. L. Lanzi, D. E. Goldberg, and E. Bernadó-Mansilla. Modeling Selection Pressure in XCS for Proportionate and Tournament Selection. Technical report, IlliGAL Report No. 2007004, Urbana-Champaign IL 61801, USA, 2007.

[16] S. W. Wilson. Classifier Fitness Based on Accuracy. *Evolutionary Computation*, 3(2):149–175, 1995. http://prediction-dynamics.com/.

[17] S. W. Wilson. Generalization in the XCS classifier system. In *Genetic Programming 1998: Proceedings of the Third Annual Conference*, pages 665–674. Morgan Kaufmann, 1998.