Coevolving different knowledge representations with fine-grained parallel Learning Classifier Systems

Xavier Llorà and Josep M. Garrell
Enginyeria i Arquitectura La Salle, Universitat Ramon Llull
Psg. Bonanova 8, 08022-Barcelona, Catalonia, Spain
{xevil,josepmg}@salleURL.edu

Abstract

This paper deals with the coevolution of different knowledge representations using fine-grained parallel learning classifier systems for data mining tasks. The objective is to demonstrate that a fine-grained parallel classifier systems can evolve individuals codifying different knowledge representations at the same time. This goal is achieved exploiting spatial relations of fine-grained parallel algorithms to favor the coevolution of knowledge representations, as well as extinction patterns. Experiments were performed with GALE2, a fine-grained parallel learning classifier system. Experiments focused on the diversity of the coevolved individuals, their classification accuracy, and the usefulness of the method proposed.

1 INTRODUCTION

The goal of a data mining process for classification tasks is the extraction of a certain knowledge from a given data set. The knowledge obtained from a data set ($\mathcal{P}$) to be mined is usually expressed in a certain formal language or representation. The knowledge representation used by data mining algorithms may differ between approaches. For instance, common knowledge representations for data mining are rules or decision trees, among others. On the other hand, some data mining algorithms are specially tailored for a given knowledge representation, constraining the scope of their application. Fine-grained parallel learning classifier systems can overcome this situation. Furthermore, they provide a knowledge-independent model for data mining [Llorà and Garrell, 2001c]. This paper explores how this kind of classifier systems can coevolve different knowledge representations simultaneously.

Learning classifier systems, like XCS [Wilson, 1995], have been applied to data mining problems, often looking for rule sets induction. Some examples of the application of XCS to data mining problems can be found in [Wilson, 2000, Saxon and Barry, 2000, Bernadó et al., 2001]. But, there has also been some attempts to introduce changes into the knowledge representation used by XCS, using, for instance, Lisp-like s-expressions as the condition part of the rules [Lanzi and Perrucci, 1999, Lanzi, 2001]. On the other hand, fine-grained parallel learning classifier systems, like GALE [Llorà and Garrell, 2001c], differ from this type of learning classifier systems using evolutionary models that exploits knowledge independence.

This paper explores how different knowledge representations can be coevolved in a fine-grained learning classifier scheme. This characteristic is useful when dealing with data mining problems. Presenting the knowledge using different representations may help further understanding and usage of the knowledge mined. Thus, in order to achieve this goal, GALE is modified to deal with heterogeneous runs, where individuals of the population codify different knowledge representations in their genotypes. However, the coevolution of different knowledge representations at the same time has some problems that must be taken into account. Among others, knowledge representations usually require different amounts of time to find a solution. Therefore, the knowledge representations that can achieve a solution (or a local optima) rapidly may over-take the space in the board. But, these solutions may not be the best ones in the long term run.

Therefore, this approach leads to a fine-grained parallel learning classifier system (GALE2) that exploits: (1) spatial relations to favor the coevolution of individuals using a board, and (2) extinction patterns to avoid local optima [Kirley and Green, 2000] and the take over of the board for a given knowledge representation. This paper focuses on extinction patterns and explores how they can help GALE2 to coevolve efficiently different knowledge representations at the same time. Using these two new contributions, GALE2 becomes a knowledge-independent data mining model capable of expressing the induced knowledge using several knowledge representations.

The rest of the paper is structured as follows. Section 2 describes the modification introduced in GALE to enable the coevolution of different knowledge representations that led to GALE2. The description presents the evolutionary model used, as well as the coevolutionary mechanisms introduced (selective neighborhood and extinction patterns). Next, section 3 presents the results obtained using GALE2 solving well-known data mining problems for classification tasks. Finally, section 4 presents some conclusions for the work presented.

2 GALE2 VERSUS GALE

Genetic and Artificial Life Environment (GALE) is a classifier scheme based on fine-grained parallel genetic algorithms. GALE was firstly introduced in [Llorà and Garrell, 2001c] as a data mining algorithm, being designed for solving classification tasks [Llorà and Garrell, 2001b, Llorà, 2002]. This section begins describing GALE2, focusing on its parallel evolutionary model and the main differences when compared to GALE. Then, the section pays attention to the knowledge representations evolved by GALE2 in this paper. Finally, section concludes discussing one of the main issues of GALE2: extinction patterns.

2.1 MODELING GALE2

GALE uses a 2D grid (board $\mathcal{T}$) form by $m \times n$ cells for spreading spatially the evolving population. Each cell ($\mathcal{T}_{ij}$) of the grid contains either one ($\zeta(\mathcal{T}_{ij}) = 1$) or zero individuals ($\zeta(\mathcal{T}_{ij}) = 0$); thus, for instance a 32×32 grid can contain up to 1024 individuals, each one placed on a different cell. Each individual ($\mathcal{T}_{ij}^I$) is a complete solution to the classification problem, in fact, each individual codifies the knowledge that describes the mined data set. GALE2 differs from GALE in the fact that the individuals are not homogeneous. Thus, an individual in GALE2 codifies one of the different knowledge representations available, as later explained (see section 2.2). Genetic operators are restricted to the immediate neighborhood ($\mathcal{T}_{ij}^\nu$) of the cell in the grid. The size of the neighborhood is r. Given a cell $\mathcal{T}_{ij}$ and $r = 1$, the neighborhood $\mathcal{T}_{ij}^\nu$ of $\mathcal{T}_{ij}$ is defined by the 8 adjacent cells to $\mathcal{T}_{ij}$ (being $\zeta\left(\mathcal{T}_{ij}^\nu\right)$ the number of occupied cells in $\mathcal{T}_{ij}^\nu$). Thus, r is the number of hops that defines the neighborhood.

To introduce the coevolution of different knowledge representations, GALE2 uses a modified version of the neighborhood definition proposed in GALE. Selective neighborhood ($\mathcal{T}_{ij}^\vartheta$) is defined in terms of the whole neighborhood $\mathcal{T}_{ij}^\nu$, but the selective neighborhood is restricted to the cells in $\mathcal{T}_{ij}^\nu$ that contain individuals codifying the same knowledge representation of $\mathcal{T}_{ij}^I$ ($\zeta\left(\mathcal{T}_{ij}^\vartheta\right) \subseteq \zeta\left(\mathcal{T}_{ij}^\nu\right)$). Changing the neighborhood definition implies a change in the way that genetic operators are used in GALE2 in comparison to GALE, as we introduce later. Every cell in GALE runs the same algorithm in parallel which summarizes as:

```
GALE2(T,P)
   FOR-EACH T_ij ∈ T
   DO IN PARALLEL
       t ← 0
       initialize T_ij
       evaluate T_ij^I using P
       REPEAT
             t ← t+1
             merge T_ij^I among T_ij^ϑ
             split T_ij^I among T_ij^ν
             evaluate T_ij^I using P
             survival of T_ij among T_ij^ϑ
             extinction of T_ij among T_ij^ϑ
                if O(T) reaches a 100%
       UNTIL Ω(T_ij, t)
   DONE
RETURN T
```

During the initialization of each cell, GALE2 builds a random individual, as it is done in GALE. Not all the cells contain individuals (probability of occupation p_ζ), thus they can be full (with one individual) or empty. Each knowledge representation used has the same likelihood to be used in this process. $O(\mathcal{T})$ is the percentage of occupied cells in the board $\left(O(\mathcal{T}) = \frac{\sum_{ij}\zeta(\mathcal{T}_{ij})}{n \times m}\right)$. The individual $\mathcal{T}_{ij}^I$ is evaluated using the data set $\mathcal{P}$ to be mined. The fitness function used in GALE2 is the same used in GALE, $fit(I) = \left(\frac{I^c}{l}\right)^2$ [De Jong and Spears, 1991], being I^c the number of correctly classified instances and l the number of instances of the $\mathcal{P}$ data set. Next, the evolutionary cycle starts. This process runs until $\Omega(\mathcal{T}_{ij}, t)$ is satisfied. $\Omega(\mathcal{T}_{ij}, t)$ is satisfied when all the instances are correctly classified ($fit(I) = 1$), or a certain amount of iterations (k_{max}) are completed.

The *merge* in GALE2 crosses the individual in the cell with one individual randomly chosen among its selective neighborhood $\mathcal{T}_{ij}^{\vartheta}$, with a given probability $p_{\mathcal{M}}$, instead of using the whole neighborhood $\mathcal{T}_{ij}^{\nu}$ proposed by GALE. This implementation used in GALE2 leads to restricted mating among individuals codifying the same type of knowledge representation. *Merge* generates only one individual that replaces the individual in the cell, as later explained (see section 2.2).

Then, *split* is applied with a given probability $p_s(\mathcal{T}_{ij}^{I}) = k_{sp} \cdot fit(\mathcal{T}_{ij}^{I})$, being $k_{sp} \in [0,1]$ the maximum splitting rate. In GALE2, *split* works in the same way as proposed in GALE, but combining the whole and the selective neighborhood information. *Split* clones and mutates the individual in the cell. The new individual is placed in the empty cell $\mathcal{T}_{kl}$ of the whole neighborhood $\mathcal{T}_{kl} \in \mathcal{T}_{ij}^{\nu}$ with higher number of occupied cells in its whole neighborhood $(\max(\zeta\left(\mathcal{T}_{kl}^{\nu}\right))$. If all cells of the whole neighborhood are full $\left(\zeta\left(\mathcal{T}_{ij}^{\nu}\right) = 8\right)$, the new individual is placed in the cell of the selective neighborhood $\mathcal{T}_{ij}^{\vartheta}$ that contains the worst individual (lower fitness).

The last step in the evolutionary cycle, *survival*, decides if the individual is kept for the next cycle or not. This process uses the neighborhood information. If a cell has up to one neighbor $(\zeta(\mathcal{T}_{ij}^{\nu}) \leq 1)$, then the probability of survival of the individual is $p_{sr}^{\zeta(\mathcal{T}_{ij}^{\nu})\leq 1}(\mathcal{T}_{ij}) = fit(\mathcal{T}_{ij}^{I})$, as proposed in GALE. Else if a cell has seven or eight neighbors $\zeta(\mathcal{T}_{ij}^{\nu}) \geq 7$ then $p_{sr}^{\zeta(\mathcal{T}_{ij}^{\nu})\geq 7}(\mathcal{T}_{ij}) = 0$, where the individual is replaced by the best selective neighbor in $\mathcal{T}_{ij}^{\vartheta}$. This method, introduced in GALE2, proposes a restricted selective pressure among individuals codifying the same knowledge representation. On the other neighborhood configurations $(1 < \zeta(\mathcal{T}_{ij}^{\nu}) < 7)$, an individual survives if and only if $fit(\mathcal{T}_{ij}^{I}) \geq \overline{\mu}_{nei}^{\vartheta} + k_{sr} \times \sigma_{nei}^{\vartheta}$; $\overline{\mu}_{nei}^{\vartheta}$ is the average fitness value of the occupied selective neighbor cells $\mathcal{T}_{ij}^{\vartheta}$, and σ_{nei}^{ϑ} their standard deviation. k_{sr} is a parameter that controls the survival pressure over the current cell.

2.2 KNOWLEDGE REPRESENTATION

The evolutionary model of GALE2 coevolves different knowledge representations. In this paper, GALE2 coevolves three different knowledge representations in its heterogeneous runs: (1) sets of fully-defined instances [Llorà and Garrell, 2001b], (2) orthogonal decision trees [Quinlan, 1993], and (3) oblique decision trees [Breiman et al., 1984, Van de Merckt, 1993]. Rules can be extracted from orthogonal decision trees. Instance sets are evolved sets of instances that describe the set $\mathcal{P}$ mined, based on *nearest neighbor* algorithms. *Merge* uses two-point crossover [De Jong and Spears, 1991], and splitting is done using mutation based on generating some new values for genes randomly. The other two evolved knowledge representation are based on decision trees, codified as dynamic trees [Llorà and Garrell, 2001a]. The genetic operators used are one point crossover and random constants perturbation [Koza, 1992].

2.3 EXTINCTION PATTERNS

The last modification introduced by GALE2 is the usage of extinction patterns. The idea behind these patterns is the deletion of individuals from $\mathcal{T}$ leaving some room. The goal is to help the evolutionary algorithm to avoid local optima and the over-take of the space in $\mathcal{T}$ by a single type of knowledge representation. This idea is similar to the work proposed by [Kirley and Green, 2000], although they were solving optimization problems using *cellular genetic algorithms* [Whitley, 1993]. Nevertheless, extinction patterns have to favor the diversity across the board, and ensure the coevolution of all the knowledge representations used. This is a key point if we want to coevolve all the knowledge representations at the same time, without losing any of them along the evolutionary path.

There are several ways to approach to extinction patterns. This paper explores two different types of extinction patterns: (1) lower bound extinction patterns, and (2) upper bound extinction patterns. Lower bound extinction patterns bias board evolution toward selective neighborhoods with higher connection degrees $\left(\zeta\left(\mathcal{T}_{ij}^{\vartheta}\right)\right)$. On the other hand, upper bound extinction patterns favors selective neighborhoods with lower connection degrees. Section 3 discusses, among others, which one of these patterns is the most suited for the coevolution of different knowledge representations.

The extinction patterns used by GALE2 are applied when the occupation of the board $(O(\mathcal{T}))$ reaches 100%. We introduce two kinds of extinction patterns: (1) lower bound extinction patterns defined as $\zeta\left(\mathcal{T}_{ij}^{\vartheta}\right) \leq k$, and (2) upper bound extinction patterns $\zeta\left(\mathcal{T}_{ij}^{\vartheta}\right) \geq k$, being $k \in [0,8]$. Once the board $\mathcal{T}$ reaches full occupation, each cell $\mathcal{T}_{ij}$ test the extinction pattern used. If it is satisfied, the individual $\mathcal{T}_{ij}^{I}$ is deleted, leaving the cell empty, $\zeta(\mathcal{T}_{ij}) = 0$. For a given run, GALE2 uses only one extinction pattern. For instance, if the extinction pattern were $\zeta\left(\mathcal{T}_{ij}^{\vartheta}\right) \leq 4$, this test would delete all the individuals that were kept in cells that satisfy that they have less than five selective neighbors.

3 RESULTS

This section focuses on the coevolution of several knowledge representations using GALE2. The results presented in this section do not deal with the generalization capabilities of classification accuracy (in terms of cross-validation runs) of the algorithm. Some previous work in this direction using GALE can be found in [Llorà and Garrell, 2001c]. This previous work evaluate the competence of GALE when compared to well known classifiers like XCS [Wilson, 1995], C4.5 [Quinlan, 1993], or IBL [Aha and Kibler, 1991], among others. Instead, the experiments conducted in this paper look inside the evolutionary process focusing on the coevolution of individuals that encode different knowledge representations. Moreover, the experimental runs were also prepared to study the impact of the extinction patterns, presented in the previous section, in the behavior of the coevolution that takes place in the board $\mathcal{T}$ of GALE2.

In the experiments, GALE2 coevolved simultaneously the three different knowledge representations presented in the previous section. This means that an individual in a run encode in its genotype either an orthogonal decision tree, or an oblique decision tree, or a set of fully-defined instances. In order to illustrate the behavior of GALE2, it was used to solve two well-known data sets provided by the UCI repository [Merz and Murphy, 1998]: (1) the *Iris* data set (`irs`) , and (2) the *Wisconsin Breast Cancer* data set (`wbc`). A deeper analysis using other data sets is part the further work of this paper. Thus, the experiments were designed to show the usefulness of using extinction patterns in GALE2. If they are not used, GALE2 behavior is constrained by the spatial distribution of individuals at the initialization phase, being unable to guarantee the right coevolution of all the knowledge representations available.

Table 1 summarizes the results obtained when the extinction patterns presented in the previous section are used. For each extinction pattern, GALE2 was run 50 times using different random seeds, averaging the results obtained. This table presents, on the left hand side, the results obtained for `irs` data set, whereas on the right hand side, table shows the results for the `wbc` data set. Results for each data set are summarized in terms of the extinction pattern used. Lower bound extinction patterns, defined as $\zeta\left(\mathcal{T}_{ij}^{\vartheta}\right) \leq k$, are presented at the top, whereas the bottom of the table presents the results provided by the upper bound extinction patterns $\zeta\left(\mathcal{T}_{ij}^{\vartheta}\right) \geq k$. Each row in the tables shows the accuracy of the individuals in board $\mathcal{T}$, as well as the number of spatial demes and the board occupation $O\left(\mathcal{T}\right)$ for different k values. A spatial deme is defined as the set of individuals that are kept in connected cells that contain the same type of individuals. One cell c_0 is connected to another cell c_n if there is a set of cells $\{c_0, c_1, c_2 \ldots c_n\}$ that satisfies that $c_i \in c_{i+1}^{\vartheta}$ given $i = \{0, 1, \ldots n-1\}$.

The lower bound extinction patterns, $\zeta\left(\mathcal{T}_{ij}^{\vartheta}\right) \leq k$, present a clear behavior. When the extinction pressure increases (k gets close to 8), the diversity of demes in $\mathcal{T}$ falls. Patters where $k \geq 4$ produce less than three demes. Therefore, $\mathcal{T}$ does not contain individuals for all the knowledge representations available. These extinction patterns favor that the most rapidly suited spatial deme takes over the board. This fact holds when we take into count the mean accuracy of the population, as it can be seen in table 1. Nevertheless, these results show that the lost of diversity is a serious drawback for the coevolution of different knowledge representations.

On the other hand, upper bound extinction patterns, $\zeta\left(\mathcal{T}_{ij}^{\vartheta}\right) \geq k$, present a different behavior. When the extinction pressure increases (k gets close to 0), the diversity holds. This is the result of favoring demes with a small connection degrees. This fact can be observed on the amount of demes kept in $\mathcal{T}$, and in the accuracy of the board. Special mention must be done on the extinction pattern $\zeta\left(\mathcal{T}_{ij}^{\vartheta}\right) \geq 4$. This pattern produces the larger number of accurate spatial demes. This fact was observed in both problems, `irs` and `wbc`. Thus, the balance between diversity and uniformity, proposed by upper bound patterns, produce in GALE2 rich boards. The worth of these boards is that they are examples of how different demes can be efficiently coevolve at the same time for all the knowledge representations available, without destroying diversity. Therefore, as a result of the tests done, upper bound extinction patterns help the coevolution and diversity of different knowledge representations in GALE2. This issue is important for achieving the goal of coevolving different knowledge representation in data mining.

Some look inside GALE2 dynamics can be found in figures 1 and 2. These figures show how an extinction pattern can change the behavior of GALE2. The two figures are obtained using GALE2 solving the `irs` problem. The runs presented in the figures share the same parameters values (as shown in the appendix), as well as the random seed. This means that all the runs share the same behavior until the board collapses, presenting no empty cells. Then, when $O(\mathcal{T})$ reaches 100% (no cell remains empty in the board), GALE2 applies an extinction pattern (see section 2.3).

Lower bound extinction patterns tend to produce a big extinction at the first application of the pattern (figure 1). But evolution adapts the spatial demes location contained in $\mathcal{T}$, reducing the scope of extinction in following pattern applications. But this adaptation is obtained by losing the diversity of the board, as early explained. This fact can be observed in figure 2. Each cell $\mathcal{T}_{ij}$ is represented using the following color code: white (empty cell), black (oblique decision three), dark gray (orthogonal decision tree), light gray (set of fully-defined instances). On the other hand, upper bound extinction patterns tend to produce a greater diversity, but eventually (as the extinctive pressure increases) they turn unstable leading to the total extinction of the population in $\mathcal{T}$, as shown in figure 1. Figure 2 also presents some snapshots of the board evolution using upper bound extinction patterns.

The main characteristic of GALE2 is that it can evolve several knowledge representations in the same heterogeneous run. As a data mining algorithm, it can provide different explanations for the data set being mined, helping the user to understand the problem being solved. We want to conclude this section of results showing some examples of the solutions coevolved using GALE2. The individuals presented are solutions to the `irs` problem. This problem is defined using 4 attributes (sepal length (**S.L.**), sepal width (**S.W.**), petal length (**P.L.**), and petal width (**P.W.**)), as well as three different classes (iris setosa (`set`), iris versicolor (`ver`), iris virginica (`vir`)). At the right hand side of each individual we also present its accuracy using the whole `irs`, by showing its confusion matrix. Each row in the matrix represents the class of the instance to classify, whereas each column is the predicted class by the individual. The first individual is an orthogonal decision tree (∗ marks the leaf that misclassified one instance), whereas the second one is a set of fully-defined instances.

```
S.W. ≤ 3.023 : set
|   P.W. ≤ 1.537
|   |   P.L. ≤ 4.957
|   |   |   P.W. ≤ 0.702 : set
|   |   |   P.W. > 0.702 : ver
|   |   P.L. > 4.957 : vir
|   P.W. > 1.537
|   |   P.L. ≤ 4.957 : vir
|   |   P.L. > 4.957
|   |   |   P.W. ≤ 1.783 : ver*
|   |   |   P.W. > 1.783 : vir
S.W. > 3.023
|   P.W. ≤ 1.014 : set
|   P.W. > 1.014
|   |   P.W. ≤ 1.537 : ver
|   |   P.W. > 1.537
|   |   |   P.L. ≤ 4.957 : ver
|   |   |   P.L. > 4.957 : vir
```

	set	ver	vir
set	50	0	0
ver	0	50	0
vir	0	*1	49

S.L.	S.W.	P.L.	P.W.	Cls
4.665	3.608	3.573	0.414	`set`
5.574	2.844	3.444	1.952	`vir`
5.574	2.858	3.444	1.631	`ver`
4.882	2.281	5.600	1.836	`vir`
5.574	2.353	6.627	2.450	`vir`
5.574	2.844	6.627	1.302	`vir`

	set	ver	vir
set	50	0	0
ver	0	50	0
vir	0	0	50

4 CONCLUSIONS

This paper presented how different knowledge representations can be coevolved in a fine-grained learning classifier scheme. In order to achieve this goal, a previous learning classifier systems (GALE) was modified to deal with heterogeneous runs, where individuals of the population codify different knowledge representations in its genotype (GALE2). This approach leads to a classifier scheme that exploits: (1) spatial relations to favor the coevolution of individuals, and (2) extinction patterns to avoid local optima.

Results show that the coevolution of different knowledge representations is possible. Moreover, the results obtained also show that, when an adequate extinction pattern is used, accurate individuals belonging to different knowledge representations can be coevolved efficiently. Upper bound extinction patterns also help GALE2 to avoid that a particular type of knowledge representation over-take the space of the board. Experiments show that upper bound extinction patterns tend to favor diversity (e.g. $\zeta\left(\mathcal{T}_{ij}^{\vartheta}\right) \geq 4$).

GALE2 also shows that with few more efforts, it performs as GALE. Nevertheless, GALE2 can effectively coevolve different knowledge representations at the same time reducing the number of homogeneous runs previously needed by GALE (one for each knowledge representation). Therefore, this leads to an important reduction of the resources needed (using the same parameter configuration for GALE and GALE2). On the other hand, as a data mining tool, GALE2 has the advantage, when compared to other approaches, of showing the user different kinds of solutions, favoring a deeper look at the knowledge mined.

Acknowledgments

We would like to thank the support provided by *Generalitat de Catalunya* (*DURSI* under grant number 1999FI-00719), as well as the support of *Enginyeria i Arquitectura La Salle* to our research group. We are also grateful to the Illinois Genetic Algorithms Lab people and to his director, David E. Goldberg, for the the challenging time that Xavier spent there. Finally, we also want to thank the anonymous reviewers for their useful comments that let us greatly impro-

Table 1: Results obtained using the irs (left) and wbc (right) data sets

$\zeta(\mathcal{T}_{ij}) \leq$	**Accuracy**	**#Demes**	**Occupation**	$\zeta(\mathcal{T}_{ij}) \leq$	**Accuracy**	**#Demes**	**Occupation**
0	81.87±3.12	37.16±4.89	100.00±0.00	0	72.77±4.16	29.18±7.89	100.00±0.00
1	85.60±3.32	18.32±3.48	100.00±0.00	1	76.34±3.27	32.20±4.32	100.00±0.00
2	88.11±3.58	12.62±2.90	100.00±0.00	2	84.24±2.80	14.82±3.23	100.00±0.00
3	92.11±3.26	5.02±1.50	99.99±0.03	3	87.32±2.67	6.30±1.73	98.56±0.04
4	96.57±1.87	2.14±0.76	99.94±0.18	4	93.24±2.33	1.96±0.57	99.62±0.02
5	97.28±1.30	1.94±0.65	99.53±0.67	5	95.98±1.20	2.01±0.40	99.04±0.05
6	97.09±1.55	1.82±0.56	98.88±1.76	6	96.66±0.01	1.96±0.60	98.80±0.12
7	97.26±1.56	1.58±0.53	98.05±3.42	7	96.42±0.01	1.92±0.60	97.99±0.24
$\zeta(\mathcal{T}_{ij}) \geq$	**Accuracy**	**#Demes**	**Occupation**	$\zeta(\mathcal{T}_{ij}) \geq$	**Accuracy**	**#Demes**	**Occupation**
8	80.98±3.20	43.30±6.08	98.93±1.23	8	90.58±0.01	40.02±5.07	99.60±0.43
7	81.08±3.48	40.42±5.93	99.96±0.09	7	90.33±0.01	37.82±5.00	99.99±0.03
6	83.55±3.47	30.46±5.41	100.00±0.00	6	91.36±0.01	28.90±5.64	100.00±0.00
5	83.98±3.06	37.44±6.06	99.99±0.01	5	91.03±0.01	37.64±6.02	100.00±0.00
4	84.19±3.40	85.78±23.24	94.61±8.51	4	87.80±0.01	124.80±10.91	99.99±0.03
3	87.37±6.50	14.62±8.04	54.54±36.50	3	92.65±0.01	19.54±10.81	69.47±30.36
2	42.34±29.87	0.94±1.28	22.54±1.50	2	56.82±0.28	1.00±0.97	23.06±19.19
1	1.68±8.36	0.10±0.51	0.68±0.01	1	0.00±0.00	0.00±0.00	0.00±0.00

ve the quality and the clarity of this paper.

Appendix

In order to allow the replication, the parameters of GALE2 were set as follows: m×n=32×32, k_{max}=100, p_{ζ}=.4, $p_{\mathcal{M}}$=.4, k_{sp}=.5, p_{mu}=.003, k_{sr}=-.25. Discussion about parameter setting can be found in [Llorà, 2002].

References

[Aha and Kibler, 1991] Aha, D. and Kibler, D. (1991). Instance-based learning algorithms. *Machine Learning*, 6:37–66.

[Bernadó et al., 2001] Bernadó, E., Llorà, X., and Garrell, J. M. (2001). XCS and GALE: a Comparative Study of Two Learning Classifier Systems with Six other Learning Algorithms on Classification Tasks. *Proceedings of the 4th International Workshop on Learning Classifier Systems (IWLCS-2001)*, page to appear.

[Breiman et al., 1984] Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Classification and Regression Trees.* Wadsworth International Group.

[De Jong and Spears, 1991] De Jong, K. A. and Spears, W. M. (1991). Learning Concept Classification Rules Using Genetic Algorithms. *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, pages 651–656.

[Kirley and Green, 2000] Kirley, M. G. and Green, D. G. (2000). An Empirical Investigation of Optimization in Dynamic Environments Using Cellular Genetic Algorithms. *Genetic and Evolutionary Computation Conference (GECCO2000)*, pages 11–18.

[Koza, 1992] Koza, J. R. (1992). *Genetic Programing: On the Programing of Computers by Means of Natural Selection (Complex Adaptive Systems).* MIT Press.

[Lanzi, 2001] Lanzi, P. L. (2001). Mining Interesting Knowledge from Data with XCS Classifier System. In *Proceedings of the Genetic and Evolutinary Computation Conference (GECCO 2001)*, pages 958–965. Morgan Kauffmann.

[Lanzi and Perrucci, 1999] Lanzi, P. L. and Perrucci, A. (1999). Extending the Representation of Classifier Conditions Part II: From Messy Coding to S-Expressions. In *Proceedings of the Genetic and Evolutinary Computation Conference (GECCO'99)*, pages 345–352. Morgan Kauffmann.

[Llorà, 2002] Llorà, X. (February, 2002). *Genetic Based Machine Learning using Fine-grained Parallelism for Data Mining.* PhD thesis, Enginyeria i Arquitectura La Salle. Ramon Llull University, Barcelona, Catalonia, European Union.

[Llorà and Garrell, 2001a] Llorà, X. and Garrell, J. M. (2001a). Evolution of Decision Trees. In *4th Catalan Conference on Artificial Intelligence (CCIA'2001)*, pages 115–122. Morgan Kauffmann.

[Llorà and Garrell, 2001b] Llorà, X. and Garrell, J. M. (2001b). Evolving Partially-Defined instances with Evolutionary Algorithms. In *Proceedings of the 18th International Conference on Machine Learning (ICML'2001)*, pages 337–344. Morgan Kauffmann.

[Llorà and Garrell, 2001c] Llorà, X. and Garrell, J. M. (2001c). Knowledge-Independent Data Mining with Fine-Grain Parallel Evolutionary Algorithms. In *Proceedings of the Genetic and Evolutinary Computation Conference (GECCO 2001)*, pages 461–468. Morgan Kauffmann.

[Merz and Murphy, 1998] Merz, C. J. and Murphy, P. M. (1998). UCI Repository for Machine Learning Data-Bases [http://www.ics.uci.edu/~mlearn/MLRepository. html]. *Irvine, CA: University of California, Department of Information and Computer Science.*

[Quinlan, 1993] Quinlan, R. (1993). *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers.

[Saxon and Barry, 2000] Saxon, S. and Barry, A. (2000). XCS and the Monk's Problems. In Lanzi, S. and Wilson, editors, *Learning Classifier Systems: From Foundations to Applications*, pages 223–242.

[Van de Merckt, 1993] Van de Merckt, T. (1993). Decision trees in numerical attribute spaces. In *Proceedings of the 13th Intenational Joint Conference on Artificial Intelligece*, pages 1016–1021. Morgan Kaufmann.

[Whitley, 1993] Whitley, D. (1993). Cellular Genetic Algorithms. *Proceedings of the 5th International Conference on Genetic Algorithms*, page 658.

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

[Wilson, 2000] Wilson, S. W. (July, 2000). Mining Oblique Data with XCS. *IlliGAL Report No. 2000028.*

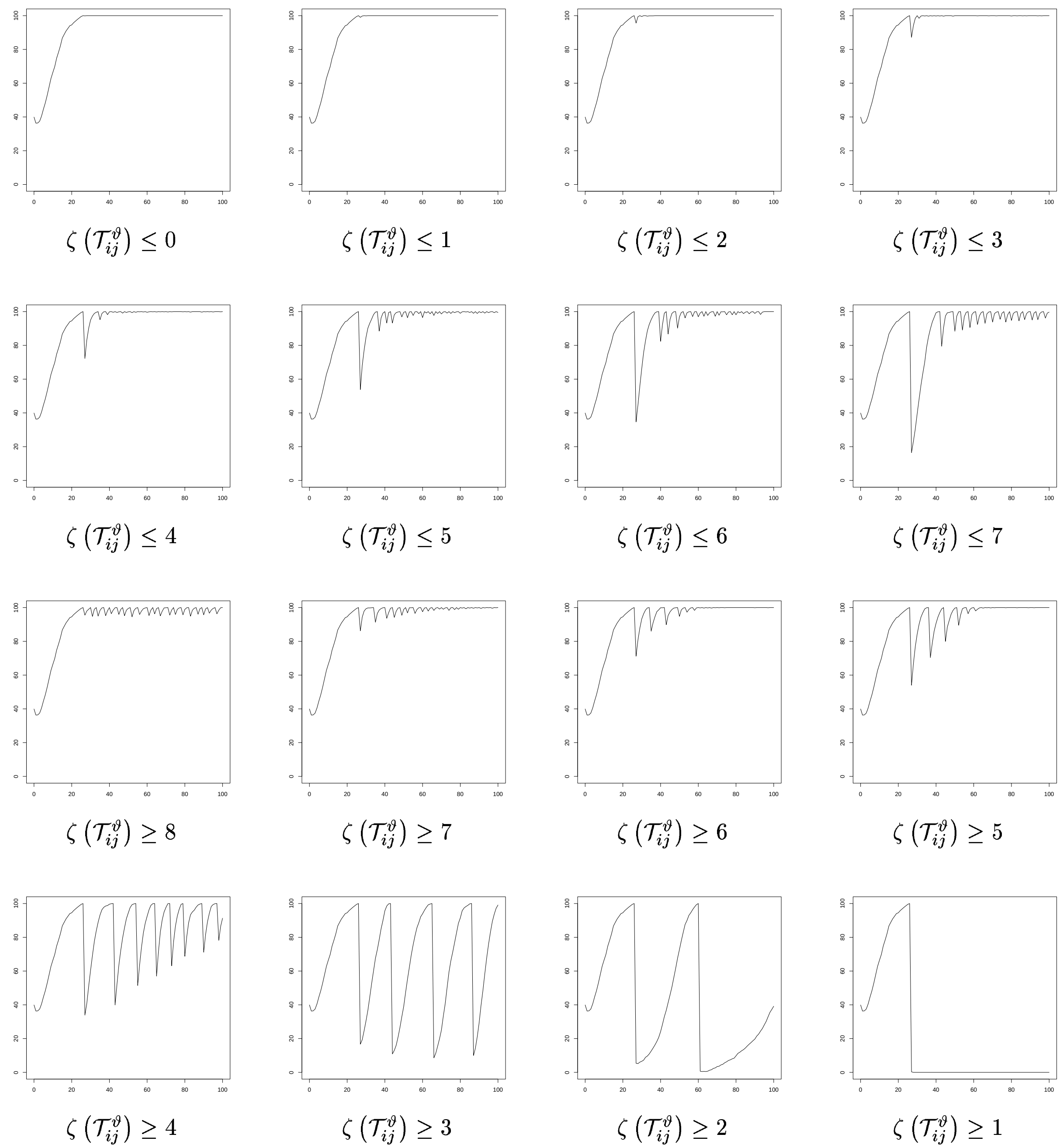

Figure 1: Board occupation for the `irs` problem using different extinction patterns. Each figure presents the board occupation percentage behavior, $O(\mathcal{T})$, through the run. The only difference between runs is the extinction pattern used. Therefore, all the runs share the same occupation behavior until iteration `t=26`. After the full board occupation, the extinction pattern is applied, leading to different evolutionary paths. The pattern used is shown below each figure. These curves are the ones obtained in the runs also presented in figure 2. As it can be seen, lower bound extinction patterns produce steady occupation of the board. On the other hand, upper bound extinction patterns produce oscillating occupation of the board favoring diversity, but eventually leading to a total extinction when extreme extinction pressure is applied.

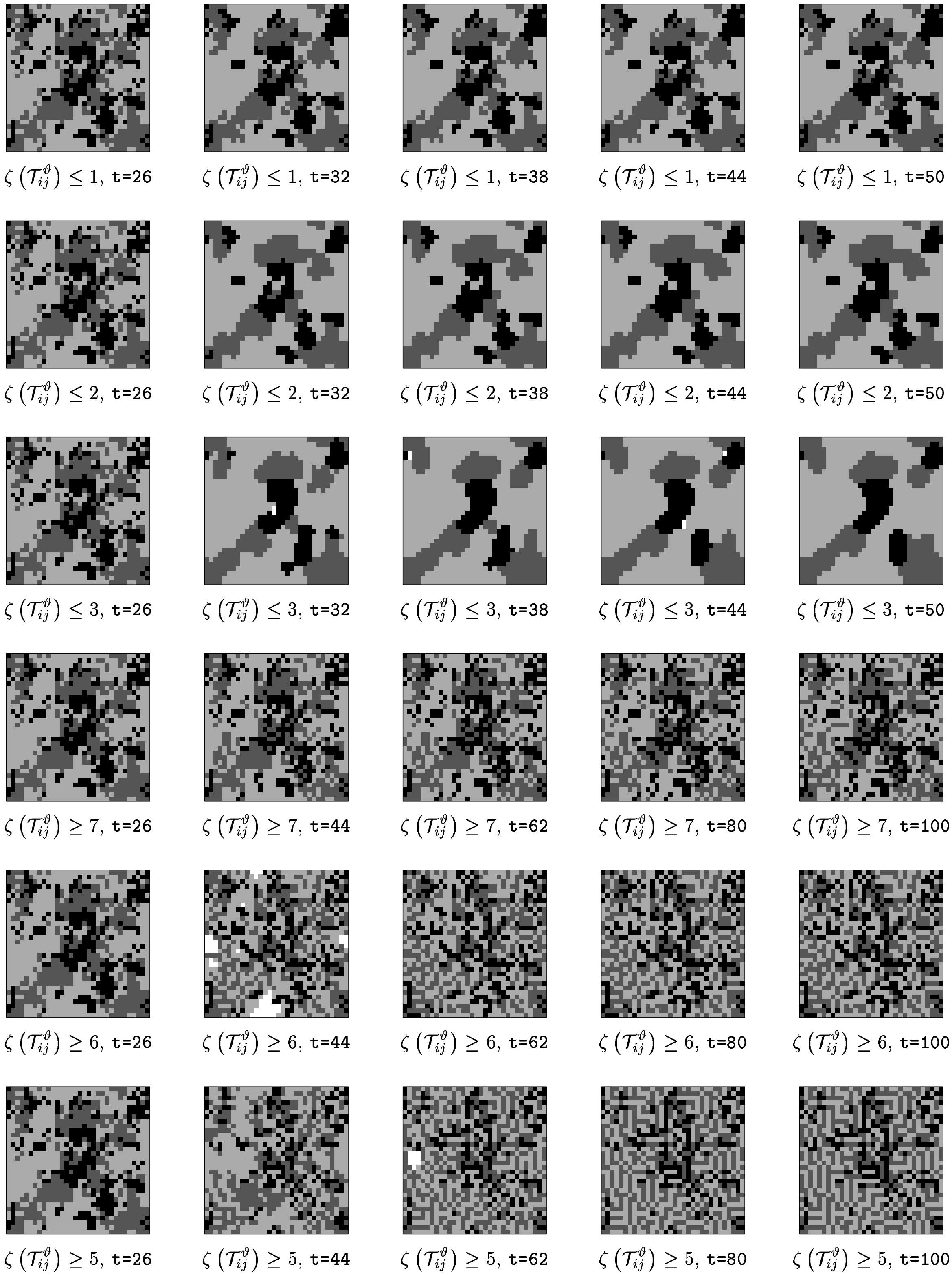

Figure 2: Board evolution for the `irs` problem using lower $\left(\zeta\left(\mathcal{T}_{ij}^{\vartheta}\right) \leq k\right)$ and upper $\left(\zeta\left(\mathcal{T}_{ij}^{\vartheta}\right) \geq k\right)$ bound extinction. Color code: white (empty cell), black (oblique decision three), dark gray (orthogonal decision tree), light gray (set of fully-defined instances).